Learning Analytics and Behavior of Distributed Self-assessment and Reflections in Programming Problem Solving

by

Mohammed Alzaid

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2022 by the
Graduate Supervisory Committee:

Ihan Hsiao, Co-Chair
Hasan Davulcu, Co-Chair
Kurt Vanlehn
Brian Nelson
Srividya Bansal

ARIZONA STATE UNIVERSITY

December 2022

ABSTRACT

Distributed self-assessments and reflections empower learners to take the lead on their knowledge gaining evaluation. Both provide essential elements for practice and self-regulation in learning settings. Nowadays, many sources for practice opportunities are made available to the learners, especially in the Computer Science (CS) and programming domain. They may choose to utilize these opportunities to self-assess their learning progress and practice their skill. My objective in this thesis is to understand to what extent self-assess process can impact novice programmers learning and what advanced learning technologies can I provide to enhance the learner's outcome and the progress. In this dissertation, I conducted a series of studies to investigate learning analytics and students' behaviors in working on self-assessments and reflection opportunities. To enable this objective, I designed a personalized learning platform named QuizIT that provides daily quizzes to support learners in the computer science domain. QuizIT adopts an Open Social Student Model (OSSM) that supports personalized learning and serves as a self-assessment system. It aims to ignite self-regulating behavior and engage students in the self-assessment and reflective procedure. I designed and integrated the personalized practice recommender to the platform to investigate the self-assessment process. I also evaluated the self-assessment behavioral trails as a predictor to the students' performance. The statistical indicators suggested that the distributed reflections were associated with the learner's performance. I proceeded to address whether distributed reflections enable self-regulating behavior and lead to better learning in CS introductory courses. From the student interactions with the system, I found distinct behavioral patterns that showed early signs of the learners' performance

trajectory. The utilization of the personalized recommender improved the student's engagement and performance in the self-assessment procedure. When I focused on enhancing reflections impact during self-assessment sessions through weekly opportunities, the learners in the CS domain showed better self-regulating learning behavior when utilizing those opportunities. The weekly reflections provided by the learners were able to capture more reflective features than the daily opportunities. Overall, this dissertation demonstrates the effectiveness of the learning technologies, including adaptive recommender and reflection, to support novice programming learners and their self-assessing processes.

ACKNOWLEDGMENTS

Alhamdulillah, I am grateful to my adviser Dr. Sharon Hsiao for her guidance, support and consistent availability that paved the way towards achieving my objectives and becoming an independent researcher. Her role model impact is exemplary beyond the academic realm. I would like to thank Professors Hasan Davulcu, Kurt Vanlehn, Brian Nelson, and Srividya Bansal for joining my dissertation committee, and providing their valuable time, comments, and feedback to enhance this work. I also thank my lab mates, Yancy Vance & Cheng-Yu Chung for their discussions and suggestions throughout my dissertation progress. I also would like to thank all my lab colleagues, advising unit, QuizIT partners, assistants, and studies participants. Without their effort and contributions to support this direction, I would not be able to fully achieve my objectives. A special thanks to my parents for their endless care, support, and encouragement throughout this journey. My wife for her support, endurance and assistance while shaping this dissertation. My family & friends for their support and prayers that kept me going during the challenging times. I would also like to thank King Saud University for providing me with the graduate scholarship towards my PhD work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

It is established that mastering programming languages and the ability to write programs has been a challenging task for novices (Robins et al., 2003; Bennedsen & Caspersen, 2007). Nowadays, introductory programming courses in universities tend to have hundreds of students, which increases the challenges for an instructor to conduct the classes with the same instructional methods. Not only that, but it also leaves the learners with limited or disrupted guidance for their self-assessment process, which is an essential process for effective learning. The manual distribution and grading process for programming practices is not scalable and may prevent learners from receiving enough learning opportunities and feedback. Additionally, if anything prevents the student from attending the class, subsequently they miss an opportunity to learn the given content, assuming it was presented during that class time. To surpass the challenge of learning programming everywhere, self-assessment should be an essential aspect of the learning process. Therefore, we now have the technology and capability to provide alternative opportunities to learners that are accessible, distributed, graded, and open for learners anywhere, at any time. However, students should be able to practice and self-assess their progress without sacrificing the guidance, support, and feedback of the formal instructor. Moreover, this process can provide them access to their knowledge representation which would help them become more aware of what they need to focus on while studying (Mitrovic & Martin, 2002).

1.1 Problem Statement & Hypotheses

We know that practicing is an essential component of learning and skill acquisition, yet student practice opportunities are limited. In introductory computer science lecture courses, students complete a handful of programming assessments throughout the semester. Each assignment may cover a range of topics rather than targeting individual concepts. Even though assignments are good practice opportunities for the students, their temporal sequence alignment could be inequivalent to the lectures. The programming practices during class may disturb the distributed retrieval practice (Benjamin, A. S., Tullis, J. 2010). Additionally, researchers considered the power of a small chunk of programming practices (Rohrer, D. 2015) and the strength of immediate reflections (Butler & Winne. 1995) to enhance the learners' opportunity in mastering the targeted objective. Thus, the only other opportunity for students to assess their performance would be the formal assessment during exams.

Typically, blended-instruction programming courses are supported by an online platform dedicated to educational technologies that facilitate multiple fronts, such as classroom response systems, Intelligent Tutoring Systems (ITS), automated assessments, etc. Research suggests that these systems can be used creatively and effectively (Blasco-Arcas, L. et al., 2013). However, there are drawbacks. For instance, students do not have enough time to analyze a complex problem and generate a meaningful response or reflection on the correct answer while using the reasoning process needed to obtain answers. Problem-solving activities may consume classroom time that might be otherwise devoted to teaching and focusing on important concepts. Moreover, there are options when choosing the content covered in the class, and tradeoffs between managing

complex technology and classroom disruptions (Sharples, M., 2013). The content usually is specialized to a specific domain and may not be comprehensive enough to address all the required topics in an introductory course.

Thus, I hypothesized that we could improve superficial fundamental knowledge through a sequence of distributed self-assessment practices and reflections. I was motivated by the positive effect of those learning principles and worked to provide a framework combining selected learning principles in the domain of programming to enhance the programming learning process.

1.2 Motivation & Objectives

In my research, I addressed the previously mentioned challenges to provide an abundant learning framework (including assessment, reflection space, and feedback) while balancing the content coverage and consolidating fundamental knowledge components for the introductory level of programming learning. The aim is to provide the learners with continuous opportunities for self-assessment and reflection and then understand the learning analytics when working on those distributed programming practices. Specifically, I focus on examining to what extent the behavior that the students show is effective in the context of distributed practices in the programming problem-solving domain.

The design rationales of the framework were based on the following learning science principles: distributed practice (Rohrer, 2015), retrieval practice and testing effects (Roediger & Butler, 2011), reflection and metacognition (Hacker, Dunlosky, & Graesser, 1998; Zimmerman & Schunk, 2012), feedback (Butler & Winne, 1995; Shute, 2008), and peer interaction (Roscoe & Chi, 2007; Topping, 2005). From That, I released QuizIT, a homegrown educational technology designed to follow programming learners' progress

through quizzes. Both the learner and the instructor are capable of consistently monitoring the progress of the course. The learners can compare their performance among the class for each learning opportunity provided, while the instructor subsequently views the aggregated progress for the whole class for each subject.

The daily learning opportunities followed the course organization while varying in topics and complexity. Students were encouraged to use the system as a supplemental non-mandatory tool for the course. To evaluate the system, I conducted an early classroom study on how students use the tool in which I collected a semester's worth of data. There, I logged the answer attempts, reviews, and reflections, among other activities, and made a comprehensive analysis of the data.

By analyzing the attempt sequences, I found that actively using the tool may lead to increased learner performance in the course. The preliminary results showed an indication of a positive correlation exists between students' reflections and the overall performance they made throughout the course, similarly, between the number of learning opportunities encountered and the overall success ratio. The temporal distribution of attempts showed a consistent activity with a higher turnout in the week before the exam. Moderate complexity opportunities were interestingly higher to capture learner reflections but that requires investigating the reflections triggers to explain the results. The small chunk of programming practices and the ability for immediate reflection had a positive effect on overall course performance.

1.3 Research Questions

Based on those findings, to enhance the programming learning process, I sought to investigate learning analytics when working on distributed programming self-assessment

and reflection. I also aim to examine to what extent the behavior that the students show in the context of distributed practices is effective in the programming problem-solving domain. Thus, the following research questions need to be addressed:

1. Can self-assessment behavioral trials be used to predict students' performance?
2. How does the inclusion of personalized practice recommender to the OSSM platform affect the self-assessment process?
3. What are the embedded reflective questions' impacts?
    a) On student's behavior?
    b) On student's performance?
    c) Differences between embedded reflective questions (weekly) vs daily questions? Are weekly reflections more effective?

The answer to these questions came from a series of classroom studies that I conducted. The studies followed a similar format and were designed based on the prior study results, which I will discuss in later chapters.

The next chapters of this dissertation is organized as follows, chapter 2 reviews the related work to programming self-assessment and reflections; the methodology follows in chapter 3 with the system design and study format; chapter 4 presents the results and evaluation of the non-adaptive release; Chapter 5 follows with evaluating the comparing the adaptive and non-adaptive releases. Chapter 6 focuses on the reflective release and reflection evaluation. Chapter 7 discusses the different system releases, address the answers for the main research questions and the lesson learned, and then conclude in chapter 8 with contributions and the limitations.

CHAPTER 2

RELATED WORK

2.1 Effect of Distributed Practices

The effect of distributed practices was studied by (Rohrer, 2015). He showed that learners perform better when the learning opportunity is distributed over longer periods. In the conducted study, the results showed that having longer and more distribution of learning material was reflected in high test scores. (Roediger, & Butler. 2011) reviewed the effect of testing on learners and concluded the importance of feedback to gain from learners from testing. They also argued that having more retrieval practices increases the retention and transfer of knowledge. (Roscoe, & Chi, 2007; Topping, 2005) evaluated the effectiveness of peer interaction in the learning process. Although (Roscoe, & Chi, 2007) argued the benefit is not substantial because of focusing on delivering knowledge, both peers will gain from that interaction. However, when this interaction is not formal, which can take place by other means, the negative effect of focusing on delivering knowledge was minimal. (Butler, & Winne, 1995) discuss self-regulators in which they identify the significant value of feedback. They report that Self-regulators use the feedback to evaluate their learning objectives and estimate the learning outcome.

Researchers have consistently found that distributed practice that is incremental and spaced over time- is superior to practice that is conceptually or temporally compressed (Benjamin, A. S., & Tullis, J. 2010), (Rohrer, D. 2015). The classic example of the ineffective massed practice is “cramming,” when students conduct most of their studying and exam preparation a few days (or even hours) before an assessment (Hartwig, M. K., & Dunlosky, J. 2012). Mass practice leads to fragile and isolated knowledge that may be

quickly lost or forgotten. In contrast, distributed practice helps to periodically re-instantiate and reinforce key memories, concepts, and skills, which in turn facilitates the acquisition, recall, and transfer of knowledge.

Another fundamental aspect of effective practice is that it includes opportunities for retrieval and self-assessment (Roediger, H.L. and Butler, A.C., 2011), (Roediger III, H.L., and Karpicke, J.D., 2006). Students benefit from being evaluated, which provides cues and structures to retrieve and reflect on their current knowledge and performance (Karpicke, J.D. and Aue, W.R., 2015), (Van Gog, T., and Sweller, J., 2015). Importantly, retrieval practice and distributed practice are highly complementary, as frequent practice activities are combined with assessments to enable well-spaced opportunities for self-testing.

2.2 Feedback and Metacognition in Learning

Feedback is the information provided by an agent regarding aspects of a person's performance or understanding (Hattie & Timperley, 2007). In the learning domain, feedback is expanded to include the process in which learners understand the information about their performance and use it to enhance their work or learning strategies. The feedback should focus on the external delivery of information based on observable performance on task while avoiding any internal or self-evaluative function in the feedback context (Butler & Winne, 1995). Moreover, the source of feedback can be internal or external. External feedback usually comes from instructors, ITS, or peers after performing the learning task. However, effective learners can develop distinctive cognitive routines to generate internal feedback as they learn. Educational literature agrees that learners are more effective when they deal with externally provided feedback (Bangert-Drowns et al., 1991).

What goes in the feedback matters, too; well-provided external feedback, which includes more than just the task outcome results, can trigger learners' ability to activate and access internal feedback on their strategy, goals, and knowledge.

Metacognition is a broader concept that evolves around the awareness and understanding of one's thought processes, in short, thinking about own thinking. The most relevant definition of the learning domain was "capturing two essential features of metacognition, self-appraisal, and self-management of cognition" (Paris & Winograd, 1990). Self-appraisal is a reflection about own knowledge states and abilities and the affective states concerning their knowledge, abilities, motivation, and characteristics as a learner. Reflections for self-appraisal usually address questions like (what you know, how you think, and when and why to apply knowledge or strategies) (Paris & Winograd, 1990). Self-management refers to "metacognitions in action," which is a mental process that helps to construct problem-solving (Paris & Winograd, 1990). Focusing on self-appraisal and self-management can help the learners to grow as independent learners. From this definition, we understand that metacognition does not depend on external factors; instead, its source is tied to the learner's internal mental structure, which can include what one knows about that structure, how it works, and how one feels about it. (Hacker,1998). Metacognition can be triggered consciously or subconsciously, which makes it even harder to detect. When it comes to supporting the learners, metacognition and feedback are intertwined. Metacognition is essential to process feedback, and proper feedback on tasks is important to developing metacognition. Feedback is an essential component of learning to trigger thinking about one's thinking. When efficiently provided, it can raise the learner's

awareness of their strengths and weaknesses and their thought process while performing the task.

2.3 Programming Self-assessments and Problem Solving

Studies show that novices in introductory programming classrooms often have limited fundamental knowledge or superficially organized knowledge and fail to apply relevant knowledge (Robins, A., Rountree, J., & Rountree, N.,2003). Because of the lack of solid fundamental knowledge, novices may develop a trial-and-error approach to fix code errors (Blikstein, P., 2011), (Lister, R. et al., 2004) in which students tend to use that approach throughout their learning and into the advanced courses. That may even create more errors than it solves (Law, L. C. 1998).

Gaining knowledge in the programming domain requires exposure to learning opportunities and practices. For novice programmers, the learning process usually begins with learning variables and iterations to prepare for the introduction of object concepts and procedures. With proper practice and consistent exposure to the concepts, learners may feel safe and proceed with missing fundamental concepts that may cause future coding errors (Law, L-C. 1998).

From capturing the traces of learning data while performing programming problem-solving, educators can use it as an indicator of the learner's performance. (Spacco et al., 2015), To aid with that, QuizJET (Hsiao, I.H, 2010) is one example of a program that facilitates automatic programming evaluation by using parameterized exercises to create programming questions. (Papancea et al., 2013) and (Baker & Inventado, 2014) aimed to capture the programming problem-solving process to get a better insight into the factors that could be utilized to enhance learning or identify at-risk students who might be failing

the course. By analyzing the student work patterns from programming exercises data, (Ihantola et al., 2015) noticed that students improved their programming skills as they did more exercises. They also confirmed the correlation between students' effort and their final grades. Another group of researchers (Rivers et al., 2016) examined the dataset of exercises' code produced by students from Cloud Coder (Papancea, Spacco & Hovemeyer, 2013). It analyzed the programming knowledge component in the programming problems to get a hint of which topics the students might be struggling with. From the data, they reported that students struggle the most with math operations and strings. To keep the learners on the right track, I can utilize analytics on the learning systems' data. An example of successfully implementing has been reported by (Azcona & Smeaton, 2017), from which they were able to see significant improvement in the student's performance by utilizing the model predictions built from the historical data to give direct feedback and support to those at risk of failing learners.

On the other hand, students need to practice at their own pace without sacrificing the guidance and support of a formal instructor. Programming learners should continually reinforce practice in hypothesizing about the behavior of their programs and then experimentally verify (or invalidate) their hypotheses. That is why they do need frequent and immediate feedback about their performance (Shute, Valerie, 2008), both in forming hypotheses and in experimentally evaluating them (Edwards, 2004). WEB-CAT (Edwards & Perez-Quinones, 2008) and ASSYST (Jackson, David & Michelle, 1997) are assessment tools built using pattern-matching techniques to verify students' answers by comparing them with the correct answers.

2.4 Behavioral Analytics in Programming Learning

Modeling students' programming learning is not a new topic. Student models usually reside in ITS or other adaptive educational systems. A student's learning is typically estimated based on the behavior logs, such as the interactions with tutors, resulting in updates on the knowledge components. In programming language learning modeling, there are several parameters used to estimate students' coding knowledge. For instance, learning can be gauged based on the sequence of programming problem-solving success (Guerra, J. et al. 2014), programming assignments progression (Piech, C. et al. 2012), dialogic strategies (Boyer, K. E. et al. 2011), programming information seeking strategies (Lu, Y., & Hsiao, I. H. (2016), assignment submission compilation behavior (Altadmri, A. et al. 2015; Jadud, M. C., & Dorn, B. 2015), troubleshooting & testing behaviors (Buffardi, K., & Edwards, S. H. (2013), code snapshot process state and generic Error Quotient measures (Carter, A. et al. 2015), etc.

Educational data mining techniques have helped educational researchers analyze snapshots of learning processes. It involves a combination of automated and semi-automated real-time coding to identify meaningful meta-cognitive planning processes in an online virtual lab environment (Montalvo, O. et al. 2010). It can also be supervised and unsupervised classification on log files and eye-tracking data to find meaningful events in an exploratory learning environment (Bernardini, A. & Conati; C., 2010).

2.5 Opening the Learner Models

In the area of ITS, the Open Learner Model OLM enables all stakeholders to evaluate the learning progress with a higher level of confidence. For the learner, especially when dealing with a new learning experience, having access to a representation of their

knowledge increases the metacognitive activity and promotes self-regulation behavior. Enabling the students to assess their knowledge helps them become more aware of what they must focus on while studying (Mitrovic & Martin, 2002). After analyzing the effect of an open student model on students' learning skills, results confirmed a positive effect on students' acquisition of knowledge and thus encouraged the use of OLM. Bull, S. (2004) presented several diverse kinds of OLM and the framework to apply it in an adaptive learning environment. Among the interesting works to open the student model was the StyLE-OLM (Dimitrova et al., 2001), a system for teaching technical terminology in a foreign language. It provided an open learner modeling component for the targeted language. SQL-Tutor and KERMIT (Mitrovic et al., 2007) applied the OLM in database learning, and the results showed a significant increase in the performance of a specific group of students. The students also reported that the OLM supported them in a deeper understanding of the domain. Mastery Grids (Brusilovsky, Peter, et al., 2015) and Open Social Student Modeling OSSM showed an ability to engage and retain students compared to the regular OLM. It also motivated students to consume significantly higher volumes of non-required content. QuizMap (Brusilovsky et al., 2011) and Progressor (Hsiao et al., 2013) utilized the OSSM in the programming self-assessment domain. Furthermore, in both systems, the students achieved significant growth in knowledge. In QuizIT, the aim is to maintain the effect of the distributed practices from the daily learning opportunities in an OSSM environment. It has a chance to combine the positive impact of both strategies on the learner's knowledge gain.

2.6 Reflective Learning

Reflective learning involves a metacognitive process that enables understanding of both the self and the learned content so that future learning can be influenced by this recurring process. Some educational domains and learning settings utilize reflection more than others, such as teaching education (Yost, 2006), medical education (Lew, 2006), and self-regulated learning (Deci, E. L. et al., 1981). Engineering and computer science education domains recognize the importance of reflection and continue to address the challenges in the way of benefiting from it (Turns et al., 2014; Fekete et al., 2000). Moreover, the impact of reflection varied based on the scope of interest from each domain. Some domains aim to utilize it to enhance the learner's performance directly and retention of the learned content (Lew, & Schmid, 2011; Yost, 2006), while others focus more on developing skills and improving the learning environment. If we examine self-regulated learning, we find reflection an essential aspect of the learning process (Winne, 1997). According to Pintrich (2004), self-reflection in self-regulated learning enables learners to use cognitive, metacognitive, and motivational processes, which assists them in achieving their learning goals. The skill of reflection is needed in education, but it may take time to develop, and there is no definite agreement on how to incorporate it in the learning setting (Boud et al., 2013). That is why there are variations of the methods that support reflective learning.

Analyzing the content of the reflections usually applies some form of dictionary-based, rule-based text analytics and machine learning (ML) approaches (Ullmann, 2015). In a series of research on reflective learning content (Ullmann, 2015), he provided classification ways to identify reflective and non-reflective sentences using rule-based and

ML. (Gibson et al. 2017) proposed a concept-matching analysis framework to detect reflective sentences and provide feedback to the writer. (Shashkov, et al., 2021) used Sentiment Analysis, topic modeling, and text clustering to identify student sentiment within the reflection and identify the topic objective of the reflection. (Chen et al. 2016) adapted topic modeling using Latent Dirichlet Allocation (LDA) to analyze students' reflection journal content. They found that the existence of weekly topics in the reflection significantly correlates with journal grades. As for a prediction, (Dorodchi et al., 2018) provided evidence that integrating reflections in the learning setting matter. They utilized it as the main feature to identify at-risk students early in a semester. By analyzing students' written reflections to extract sentiment feature vectors, they could predict at-risk students with high accuracy.

Because of its known impact on the learning process, many researchers attempted to encourage students to reflect while learning programming courses. Such attempts took different forms and settings, from group activities to self-assessments. In (Fekete, Alan, et al. 1999) encouraged reflection by supporting the students to submit a weekly plan. The plan shall include their thoughts and evaluation of the knowledge they have gained so far. Those efforts also include encouraging first-year computer science students to adopt a better learning strategy by providing their reflections on personal blogs (McDermott et al., 2010, Stone, Jeffrey, 2011). Moreover, the benefits of the reflections can extend the impact on the learner and provide valuable insight to the instructor. This textual content can be used to become an early indicator that identifies at-risk students as well as a predictor of the student's performance. In (Dorodchi, Mohsen, et al. 2018), researchers performed sequence analysis to predict the course outcome using the

reflections from self-assessments and group activities. They reported that their prediction model could achieve 95% accuracy in predicting student success in the programming course using the reflections' content. They also observed that the more the student reflects, the better performance they will get. If students put in the effort to keep reflecting for a month, they will be more hard-working students who will pass the course. Unlike their work, I am trying to motivate the students to include reflection in their self-assessment process.

2.7 Reflections Analysis and Evaluation

Regarding reflection evaluation, in literature, it is common to use a qualitative questionnaires approach as a means of evaluation, i.e. (Tang, 2002; Pee, 2002; Leung, 2010). However, questionnaires usually attempt to measure participants' qualities of reflective thinking. As such, they only help a little in evaluating the reflection content (Ullmann, 2015). The coding method is usually used for reflection assessment (Kember, 2008; Ullmann, 2015); however, there are ways to tackle that challenge. Thorough assessment of the student's reflections can reveal traces of their knowledge, behavior, and motivation. In (Kovanović et al., 2018), the authors aimed to identify linguistic indicators in self-reflection to assess the reflections. For that objective, they utilized learning analytics to classify the reflections based on the objective in writing using classifier. The classifier was constructed with features extracted from the student reflections to determine observation, motive, and goal existence in the reflection. In my work, I am considering these approaches to analyse and evaluate the short reflections.

CHAPTER 3

METHODOLOGY

3.1 The Principles Behind QuizIT Development

The research platform at the heart of this work is a homegrown system, named QuizIT. QuizIT is designed based on a set of learning science principles, including distributed practice (Rohrer, 2015), retrieval practice and testing effects (Roediger & Butler, 2011), reflection, and metacognition (Hacker et al., 1998; Zimmerman & Schunk, 2012), feedback (Butler & Winne, 1995; Shute, 2008), and peer interaction (Roscoe & Chi, 2007; Topping, 2005). The platform is introduced and used as a supplemental self-assessment tool for introductory programming courses. The objective is to provide the students with daily bite-size, distributed practices to master their programming knowledge and debugging skills. Each day, the system publishes assessment quizzes to measure the learning of the specified programming knowledge component and provides students with extended learning and reflective opportunities. The design rationale of QuizIT is further elaborated by the following learning sciences principles:

- Distributed practices: Rather than having the content presented at once, it is broken into chunks. This strategy is even more effective with constant increments of small practices over time.
- Retrieval practice and testing effects ensure that the learner keeps what they have learned and enhance long-term retention.
- Reflection and metacognition encourage students to take a moment to think about the learned content. This process helps the learner develop higher-level thinking and benefit from problem-solving.

- Feedback: When the student receives feedback, it facilitates their development as independent learners. Immediate feedback helps the students to evaluate and regulate their learning at their paste.
- Peer interaction: The benefit of peer interaction in learning is significant. The designated discussion board for the questions shows the discussions and reflections accompanied with other social features such as vote and like, which can enhance the social benefit of the reflections.
- Persistent and regularity: Providing one multiple-choice question daily keeps the student interested to check for newly posted questions and encourages them to practice regularly.

3.2 Platform, Functionality, and Design

Developing the system from scratch enabled control over the features and study objectives. This choice involved the design and development phases and a continuous evaluation of the features and users' feedback. It also includes maintaining the data (profile information, questions, reflections, interactions, and attempts), which requires additional resources to administer the system.

From the learners' perspective, the centerpiece of QuizIT is the daily questions, and all the other features evolve around it. Therefore, the system had three main user interfaces (UI) to support a seamless and continuous flow of daily quizzes: the Student UI, the Instructor UI, and the Admin UI. The focus in this section will be the student UI as it involves all critical features from a researcher's perspective. The instructor UI manages the course, while the admin UI provides support and administers the system.

3.2.1 The Student UI

Student UI contains the core components of the system and where everything exciting happens. Here, the students interact with the available features, record the answer attempts, and utilize the system to self-assess and regulate their learning in the course. The main components of the student UI are (the dashboard, the daily quiz, questions review, retry, discussion board, question list, calendar, Student profile, recommender, and announcements). Next, I will briefly discuss each of these features. The dashboard opens the student model and contains interactive analytical visualization showing student effort, progress, and performance of both the learner and the class. When clicking on a specific concept, the system will provide a list of the available questions on that concept and a detailed score of each question. The daily quiz is where all traffic will be directed when accessing the system. It contains a daily quiz tailored to follow the course syllabus on that day. In question review, the students can access the posted questions with recorded prior attempts, while in retry, the student retake the answered question with shuffled options. The discussion board is where the students post their questions, comments, and reflections. They can also engage with their peers using networking features such as upvotes and likes. The question list shows all the available questions in a sortable arrangement for the subject, date, question, number of attempts, and the associated color label of the student's question performance. The system adopted the following coloring scheme throughout the design: Green for high performance, gold for average performance, red color represents below average, and gray color means the lack of attempts recorded for the available question. This coloring scheme visually guides the students on their performance and progress. It also enables individual performance in comparison to the class. The calendar works as

another access point to the available quizzes. It also utilizes the unique color scheme mentioned earlier for question performance by decorating the months with the daily student performance. The student profile considers the student's preference, while the recommender engine provides personalized questions on demand. The announcement section provides important remarks about the course or the system updates. A snapshot shows the main aspects of the student UI can be seen in figure 1.

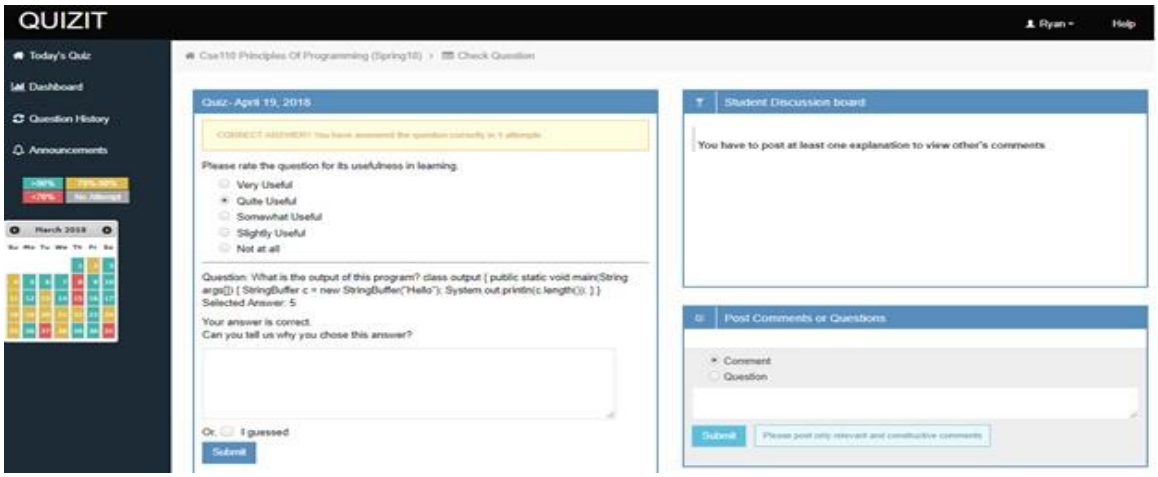


Figure 1. A Snapshot of An Answer Attempt on The QuizIT Adaptive Release.

3.2.2 The Instructor UI

With access to this UI, the instructor can manage the running course by scheduling the questions, following course progress, and reviewing students' performance. Here, the instructor can add multiple courses, edit the questions, and generate the class roster with students attempts and performance. The main course section shows a summary of the course data, interactive graphical visualization for the overall success ratio in the course, and a breakdown of each subject's performance. QuizIT provides a flexible interface for the instructor, including creating multiple courses, self-explanatory dashboard visualizations, setting and editing questions, viewing question status, and flexible calendar navigation for quick access. The QuizIT instructor dashboard visualization provides a

simplified view of the course performance using minimal color scheme and graphs. In each graph, the orange color describes the negative aspect, whereas blue represents the successful aspect of the described information. The dashboard gives the analysis of the course in three main sections. The main graphs in this section are the overall analysis and topic progress status. The overall analysis graph visualizes the course progress with detailed student information enrolled in that course, the number of questions, the number of topics, and the total number of attempts. This section also provides graphical visualization in the form of a pie and bar chart. The pie chart describes the overall success in attempts, whereas the bar chart compares the success of the class performance based on question complexity.

3.2.3 The Admin UI

This UI takes care of the administrative part of the system, such as managing the active courses, providing support to the users, and maintaining the system logs and data extraction. When the course is completed, using this UI, the admins can archive the course and generate the data for analysis.

3.3 Development Progress and Releases

The main motivation to design the system releases and the follow-up studies is enable learners to self-assess and benefit from problem-solving consequent effects. The development followed a steady and consistent introduction of system components to support three versions: Non-adaptive, Adaptive and reflective releases. To this end, QuizIT nowadays mainly emphasizes the following three components:

1. Self-Assessment and self-regulation: The main functionality of the system is to enable students to evaluate their progress with immediate feedback constantly. It also

raises their awareness of the effort they put into this task by with different features such as viewing the calendar's progress and color. I anticipated the students to utilize the system to self-assess their performance before major formal assessments. The hypothesis from the concept of Quiz a Day is to distribute that process and make them regularly evaluate promptly. This objective was the focus in the non-adaptive release and the subsequent studies.

2. Opening the student model: The generated system data and meta data from the self-assessment and the interactions enabled the integration of key system features and develop the adaptive release. Knowing that students who engages in a learning metacognitive process achieved significantly higher results than those who do not (White et al., 1999), I opened the student model. The learners are able to visualize the interactive performance and progress charts that show their performance on each course subject. The system also provides the class model for social comparisons as an OSSM component. Figure 2 shows a snapshot from the student dashboard showing the student and the class performance.

3. Personalized experience: The daily posted questions are tailored to follow the in-class progress; however, students' levels and progress may vary. Some may struggle with the daily quiz and require assistant to allocate their current level of knowledge accurately, while others could be ahead of the course and demand advanced practice opportunities. For that, I built the recommender engine at the core of the adaptive release. This engine utilized other components such as adaptive user profile, preference, and ratings. The students used the recommender to identify the questions

Figure 2. A Snapshot of the Student Dashboard Showing Both Student and Class Performance.

and subjects closer to their current knowledge level and take the lead to control and explore new subjects and question levels.

Initially, the reflection procedure was simply an open space for the learners to post their thoughts after the attempted question. later in the adaptive and reflective releases, I considered formalizing the reflection statement and time of reflection. I was able to enhance the reflection procedure by providing more incentives for reflections (Alzaid & Hsiao, 2018). The reflection procedure included the evaluation of the questions based on the usefulness of the question.

In all three releases, social interaction is maintained through the discussion board and the ability to reflect after each attempt. There are three opportunities for reflection on a given question. The preferred one by the learners was immediately after the attempt. The other two were during re-tries and the open discussion board. Each reflection or comment is associated with the question subject, complexity, and the correctness of the answer, when applicable. In the reflective release, I included another form of reflection in the reflective

release as we will see in the reflective study setup in section 7.1. To raise awareness of the reflection feature, students were encouraged to reflect during the system introduction in the classroom studies.

3.4 The Recommender Engine

The recommender engine was integrated in QuizIT adaptive release is tightly connected with the instructor and the flow of the ongoing course. The running course contains hand crafted questions by the instructor with the awareness of the recommender feature. The questions are scheduled to consistently provide daily distributed self-assessment opportunities to the learners. Therefore, the dataset of questions is limited to that daily set of questions. Each time a scheduled item is provided to the user, the recommended question option is shown during the attempt and revealed when the learner explicitly requests it.

As part of the recommender requirement, the student profile maintains the preference component for the list of concepts, complexity level and content type. Each question is tagged with the content type and concept that is mostly assessed on. The content types for programming course are defined as code, knowledge-based question, or both. To understand the underlying recommender algorithm, the recommender formula and fragments are discussed individually as follows.

$$\alpha = \max (\{f(q1),\ldots,f(qn)\}) \quad (1)$$

Assume that we have n questions in the dataset, in Eq.1 we want to calculate the score for each question and return the recommend question with the maximum score α were q= [q1...,qn]. The score for each question is maintained within the recommender. In case of tie scores α, I break ties using a random function between the tied questions.

$$f(\text{q}) = \text{Attempt q} + \Delta\text{Concept q} + \text{Performance qc} + \text{Preferences} + \text{Rating q} - \text{R} \quad (2)$$

The logic behind QuizIT adaptive recommender takes into consideration the progress of the ongoing course and the self-assessment performance. Next, let us review Eq.2 components, in which q represents the question and qc represents the question concept.

- Attempt: I consider the status of the question from the perspective of correctness on its first attempt. The algorithm set adjustable weights for questions that are new or incorrectly attempted more than the remaining set of yet to be revealed questions.
- Concept: The recommender considers the progression of a running course and the concept relation. This attribute considers the current concept in the course and the similarity between the concepts as a factor in the function. This is done using a sliding window mechanism for assigning the concept weights based on the course syllabus. Formally, it is based on the design of the course structure and syllabus in the instructor UI. The related concepts are introduced adjacently in the syllabus. Thus, the window size is set such that the nearest topics in the current course syllabus will be weighted more than other topics. The recommender is set to give the current and prior two concepts more visibility to enable retries and enhance the retention of knowledge. If the concept has already been covered, it puts more weight on recently introduced concepts since the daily quiz will continuously reveal new materials. It is important to mention here that the dataset is limited, and the recommended items may be exhaust from active learners. The Δ in Eq.2 is set at

1initially, then decreases as the course progress by a small factor for every unique question on that concept, to steadily shift towards newer concepts.

- Performance: The system maintains the learner's concept scores based on the first encounter of the daily quizzes. The increment is based on the question complexity for easy, moderate, and difficult questions with low to high values, respectively. It also decrements for any incorrect attempt. The weights are provided with an adjustable concept score threshold based on user complicity preference to reflect the learner's proficiency in the concept.
- Preferences: in the user profile, the learners are encouraged to set and update their preferences for the targeted concepts, complexity, and type of questions. If the question matches with a chosen preferred concept, complexity or content type, an additional weight is given based on the recommender specifics.
- Rating: Each question is rated by the learners for its usefulness on a scale of five after the initial encounter. I utilize the users' contribution as a collaborative filtering element to enhance the learning experience from other closely related peers.
- R: Lastly, when the learner accesses the recommended question, it will be pushed back in the list with the fixed recommended negative weight R. This ensures enhanced coverage for the available data set.

3.5 Classroom Study Designs and Objectives

The usage of this system was introduced as an additional resource to an introduction to programming courses for novices. The students were informed that the usage data would be collected for this study, and it is not counted towards their course grades.

I conducted a series of classroom studies to capture novices' learning data while working with bite-size quizzes. Initially, QuizIT was introduced as a non-adaptive release to an introductory Java programming course at Arizona State University. This course was the subject of studies in four different semesters discussed in chapter 4 & 5, having the same instructor, syllabus, and question dataset. Students were encouraged to use the system as a non-mandatory tool. These questions covered the following eighteen topics: (Java, Primitive Data Type, Method, Datatype, Expression, Variables, Strings, Arithmetic, Operator, Objects, Control, Decisions, Loops, Classes, Constructor, Arrays, 2D Arrays, Input Output). Three levels of complexity, varying from easy and moderate to more challenging questions, covered conceptual knowledge and programming skill, including code and non-code questions. The questions followed the course organization, in which quizzes on new topics are introduced once it has been initially covered in the class. I collected data until the final exam date. I used the final class grades to compare the students' performance using QuizIT and the class performance. I obtained the class grades after the final exams. The class performance was evaluated using homework, labs, quizzes, and exams. There were three study setups, with two studies in each setup. The non-adaptive study setup was used in the first and second studies and captures the initial usage data which was included in the analysis in chapter 4 and 5.

The adaptive study setup utilized the QuizIT adaptive release in third and fourth studies, with OSSM and personalized self-assessment opportunities. The evaluation is presented in chapter 5.

To encourage students to be more reflective and evaluate their learning, I released the reflective QuizIT design and conducted the reflective study setup in two

consecutive semesters. This study integrated the reflection procedure in QuizIT with weekly reflections, as we will see in chapter 7. Table 1 shows the three system releases, Non-Adaptive, Adaptive and Reflective, and the list of features included in each release. The order of the features reflects the timeline in which they were included in the systems.

Table 1. QuizIT Releases and the System Features

Features	**Non-Adaptive** ***(Chapter 4)***	**Adaptive** ***(Chapter 5)***	**Reflective** ***(Chapter 6)***
Daily Quiz	✓	✓	✓
Retries	✓	✓	✓
Calendar, List View	✓	✓	✓
General Reflection	✓		
Consecutive attempts	✓		
Discussion Board	✓	✓	✓
Peer Interaction	✓	✓	✓
Question & Comment	✓	✓	✓
Activity Tracker		✓	✓
Performance & Progress		✓	✓
Question Evaluation		✓	
Learner Profile		✓	✓
Personalized Question		✓	
Reflection Procedure		✓	✓
Learning Effort			✓
Weekly Reflection			✓

CHAPTER 4

ANALYTICS OF LEARNING BEHAVIOR

QuizIT non-adaptive release enabled consecutive attempts in which the learners can submit consecutive answer to find the correct choice. The learners showed distinct behaviors when tackling the available questions. In this chapter, I aim to investigate the learning analytics of the learner's self-assessment sessions. Here I wanted to study the impact of the trial-and-error approach that students showed during the problem-solving process. I wanted to know the impact of the trial-and-error approach in these distributed practices. I also wanted to see if the students' trails can be used to predict their performance. The main research question to be answered in this chapter is:

RQ1. Can self-assessment behavioral trails be used to predict student's performance?

For this objective, I conducted the non-adaptive classroom study setup as described in the next section.

4.1 Non-Adaptive Study Set Up and Evaluation

This study setup utilizes the initial design in QuizIT system over two semesters (Fall 16 – Spring 17) namely the first and second study consecutively. Once the learners access the system, they will be prompted with the quiz of the day. They can attempt the question at that moment or access it later from the question history. The system allows consecutive attempts when attempting a question until the correct answer is selected. Each attempt to answer is marked with the appropriate flag indicating the review source (quiz of the day, review, attempt & retry) and the correctness of the answer. Students are encouraged to reflect during the introduction of QuizIT in the classroom. The reflections

are promoted throughout the answer attempt. Peer interaction in discussion board is limited to reads and edits for participants. The peer's credentials are anonymized in the discussion board to preserve privacy and enable unbiased discussions and interactions. The student can access the already posted questions at any time using the calendar or the question history list.

In this chapter I worked with the first study data which was non-adaptive setup to evaluate the platform's effectiveness for programming novices. The second study course performance was not available and was not included in this chapter analysis. In the first study, there were 327 registered students. Among them, 130 students actively used the platform, generating a total of 9795 attempts for 110 questions during the semester. As in all the subsequent studies, the questions' design follows the course syllabus with three complexity levels to assess conceptual knowledge and programming skills. Using their course interface, the instructors can freely add questions or modify the schedule as they see fit during the course progress in the semester.

At the end of the study, the system reported an average active usage time of 51 minutes and 16 seconds. Table 2 shows the usage data captured from the active users. This table gives a snapshot of how the data distribution. Here, I only considered the data from active users, where I set the minimum threshold for the number of questions and active sessions, as seen in Table 2 in column Min. The values are calculated for active users showing the minimum and maximum values along with corresponding mean, median, and standard deviation to understand the data distribution. For clarification purposes, I will briefly describe each attribute. Questions and sessions are the number of unique quizzes attempted and the number of system sessions by a user, respectively. Attempts represent

the total number of attempts that have been performed by the learner for all quizzes attempted. It includes initial attempts and all consecutive reattempts, along with retries.

Table 2. The Collected System Data Description from the Semester Long Study Based on Users' Activities

System data		**Observed values**			
	Max	***Min***	***Mean***	***Median***	***SD***
Questions	98	**11**	34.3	35	20.6
Sessions	70	**2**	9.2	5.5	7.1
Attempts	220	16	80.4	61.5	19.8
Success	100	10	57.9	56.4	16.1
Reflections	47	0	6.8	1	11.2
Session actions	225	23	86.5	66	43.9

Retry is a system feature that provides the same question but clears the answer history and shuffles the options. Success attempts mark the sum of correct attempts containing all the correctly answered questions, including retries.

For reflections, I count the number of reflections made by active users before, during, or after the quiz attempt. Those reflections were monitored and evaluated to discard non-constructive reflections such as (true, false, correct, choice a, etc.). Session actions represent all the activities during a given session, including answering, reviewing, and retries.

To evaluate distributed programming problem-solving patterns' impacts on learning, I conducted the following analyses: grades and pattern distribution observation, distributed practices and learning correlational performance analysis, and predictive behavioral pattern modeling analysis.

4.2 Sequences Identification

To examine students' problem-solving patterns, I consider the following: The correctness of question-and-answer pairs; the pairs' repetitions and sequences. I label the correctness of every distinct question attempted by the student as 1 or 2. 1 indicates the attempt was correct while 2 indicates the incorrectness of the answer. I assume a sequence of responses within a brief time is equivalent to providing the same single solution.

Therefore, if a student correctly attempted a question repeatedly within the same session (same login period), the correctness sequences will be collapsed, (1, 1, 1,...) -> (1) or (2, 1, 1, 1 ...) -> (2, 1); however, if one answered a question incorrectly, depending on the incorrect solution choices, only the repeated incorrect choices will be collapsed and labeled as the same pattern, i.e. if question present option a and b, then (2a, 2a, 1) -> (2, 1) or (2a, 2a, 2b, 1) -> (2, 2, 1). The overall patterns that occurred fewer than 1% among all events were omitted to simplify the representation. For instance, there were cases that students went back and forth attempted correctly and incorrectly alternately (i.e., 2, 1, 2, 2, 1…), however, such patterns were typically rare and happened fewer than 1%, thus, removed from the data set.

Table 3 summarizes the patterns found in the dataset. I found that 57.92% of the problems were correctly solved straight away. Since one of the primary features of the system is to distribute a relevant practice daily, to keep the students on track and to prepare them for conquering future complex problems, such result was not a surprise to see that most of the problems were successfully solved at their first attempts. In addition, there were a range of correct and incorrect attempt sequence combinations. For instance, the student tried a question twice and got an incorrect attempt followed by a correct attempt (pattern

B, 18.54%); another session tried a question three times and got twice incorrect attempts and ended with a correct attempt (pattern C, 9.91%).

Table 3. Pattern Distribution and Labels of the Problem-Solving Sequences, 1 for Correct and 2 for Incorrect Attempts.

Sequence	Label	Pattern %
(1)	A	57.92 %
(2, 1)	B	18.54 %
(2, 2, 1)	C	9.91 %
(2, 2, 2, 1)	D	5.44 %
(2, 2)	E	2.33 %
(2, 2, 2, 2, 1)	F	2.31 %
(2)	G	2.14 %
(2, 2, 2)	H	1.41 %

Students tend to find the correct answer during the session. However, there were also patterns found that students failed the quiz a single or multiple time, but never strove a success at the same session (pattern E: 2.33%, G: 2.14%, H: 1.41%). Fig.2 shows the behavioral distribution of the patterns along with their labeling which will be discussed next. There are several possibilities that the students temporarily gave up. They could come back and try the same quiz in a later time, or they could simply stop testing themselves. From assessing these patterns, I identified three distinct behaviors. The first one is Affirmative Behavior AF, in which the student shows the desire to keep on going with a successful streak. The second identified behavior is Experimental behavior EX, where it

shows the process of trial and error until the learner figures out the correct answer. The third behavior is the Surrendering Act SUR, where the learner stopped trying to answer the question. Once I identified the three behavioral patterns, I wanted to answer the following question, can I consider these behaviors as useful or harmful?

Next, I will examine the pattern impacts on learning to determine the usefulness of the pattern and the behavior.

4.3 Effects of Problem-Solving Pattern Distribution and Learning Performance

Figure 3 illustrates an overview of students' performances and their problem-solving patterns distribution. At the macro level, better-performing students tend to display more affirmative behavior and fewer surrendering acts.

On the other hand, there is not a conclusive general trend for poor-performing students. Their behaviors tended to vary by grades. Thus, to examine the micro level patterns, I analyzed the pattern composition by grades. Here I am using letter grades received and behavioral patterns, not to be mixed with the pattern labels used in Figure.3 A for AF, BCDDF for EX and EGH for SUR.

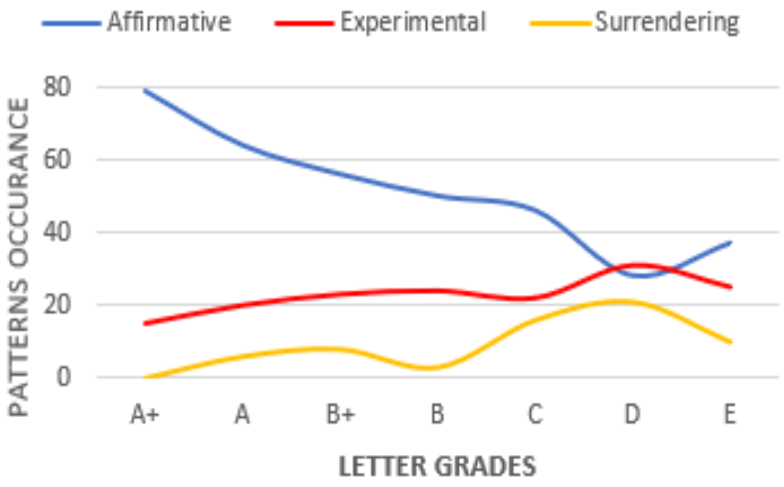


Figure 3. Pattern Distribution Labels Counts Over Course Performance in Letter Grades Received

I observed that top performers never give up. ‘A+’ students were observed as they had never given up. Not only they achieved the highest percentage of correctness rate in the questions, but also had zero surrendering patterns. Additionally, good students are affirmative and persistent. Above average students shared similar patterns. Better grades students’ persistence yielded better successes at their first tries and tended not to give up often. What separates ‘A’ to ‘A+’ or ‘B+’ to ‘A’ grades is the distribution of attempting patterns. From ‘A+’ to ‘B+’, the affirmative patterns dropped and the experimental and surrendering patterns increased. The average students persist, but lack of first-attempt success. What is interesting for average student’s ‘B’, is that they tried not to give up and committed a decent amount of trial and errors. However, they achieved low first-try-success compared to the better-performing students. This suggested that the students could be careless in answering a new quiz, they could be unprepared to take the quiz yet, or simply the pre-knowledge was insufficient. No matter which reason it might be, it is an indication of students may require more personalized help prior to problem-solving. Lastly, below average students were unprepared and lacked persistence.

C students were almost comparable to B students, except the amount of the surrendering patterns. They had shown a good amount of first-attempt correctness but failed to continue practicing by giving up too easily. Such persistent strategy may be fatal to segregate them from C to B. Meanwhile, D students appeared to be a clueless cohort in problem-solving. They showed the lowest first-attempt success rate, more experimental patterns than affirmative ones, and they surrendered the most. They assembled three negative strategies and resulted in the nearly failing grades. The outcome suggested that D students could be overly relying on the distributed practices and accompanied with insufficient trial-and-

error strategy. As a result, students were stopped from progressing. Finally, the least performing group, E students, which exhibited a mixed pattern like C & D students did. These students were unprepared for problem-solving, they ineffectively made trial and errors and did not persist. Such inconsistent behaviors produced inevitable failure outcome.

4.3.1 Distributed Practices' Effects in Learning

To understand the spacing effects on how students worked on programming quizzes, I examined the students' quiz coverage (percent of total quiz covered), time spent, and pattern frequency and the relations of these analytics with the learning performances.

Spending time to work on diverse quizzes is important, spreading the work in sessions is crucial. Correlational analysis results showed that there is a positive correlation between the number of overall distinct quizzes solved and the course performance ($r= 0.15$, $p<0.01$). However, the more attempts the students had per session, the worse performance they achieved ($r=-0.104$, $p<0.01$). In another word, students indeed benefited from practicing more quizzes and more diverse problems per session, however, overdoing the problems at the same time was harmful. In fact, the more time the students spent on a session was also crucial ($r=0.16$, $p<0.01$).

The importance of first attempt success, and the disturbance of aimless trial-and-error or neglect of practices. The correlational analysis indicated that affirmative patterns (the more the student gets it correctly at the first time) positively correlate to good performance as seen in Table 4. Such result is consistent with the literature, which highlights the importance of first attempt success (Chi, M. et al., 2010). Additionally, there were two alarming behavioral sequences in experimental and surrendering patterns: when one tried all the wrong solutions on a question and finally got it correct at the last attempt (pattern F

in Table 4), and when one only gave it a shot, but never continued (pattern G in Table 4). Both scenarios depicted the failure of the ineffective trial-and-error strategy or neglect the power of practices.

4.4 Predictive Behavioral Pattern Modeling

Based on the pattern distribution by grade classifications, I further built models to examine the pattern predictability in learning and to validate the observation generalizability. From the accuracy perspective, I attempted to classify the students' performance using features that include all problem-solving patterns and total attempts from Table 2 into the standard grade classification. These features were used to build a classifier with the following algorithms: Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM) with Sequential Minimal Optimization, and ZeroR as our baseline model. I used Weka implementation (Eibe Frank et al., 2016) to train these models and generate predictions using 10-fold cross-validation. The choice of these algorithms is based on the relevance to the domain and the current dataset.

Table 4. Identified Sequences, Patterns Label Counts & Performances Correlation

	Identified Sequence			
Label	**A**	**B**	**C**	**D**
R value	**0.274**	0.075	- 0.061	- 0.065
Label	**E**	**F**	**G**	**H**
R value	- 0.076	**- 0.110**	**- 0.176**	- 0.024

From Table 5. we can see how the models perform -accuracy-wise- compared to the baseline model (ZeroR), which is set to predict the majority class in the dataset. Here I am reporting the accuracy of Cohen's kappa K and mean absolute error MAE.

Table 5. Classifiers Results Shows the Accuracy of Classifiers with RF Performing the Best.

Classifier	**Accuracy**	**Kappa**	**MAE**
ZeroR (BL)	44.2%	0	0.167
Logistic Regression	49.2%	0.306	0.112
Random Forest	**70.7%**	**0.579**	**0.106**
SVM	56.4%	0.337	0.178

For the best-performing RF model, Cohen's Kappa measure of agreement between the predicted and actual course performance was a moderate 0.576. When reviewing Table 6 to analyze the results, I am only showing the results of the students that passed the course. Here, we can see the perfect recall score for 'A+' students (N=7), 'A' group (N=60) followed as the second-best classified grade with 0.885 f score. 'B' group (N=30) had decent accuracy as well, but the noticeable result here was from 'B+' considering it had the third largest set of students (N=10) with all the misclassification in the adjacent groups, as can be seen in Fig.4. Not only 'B+', 'C+' class (N=5) perform the worst among all the classes with zero score. The 'C' class (N=10) followed the same pattern with higher accuracy.

To clarify why B+ and C+ classes had weak accuracy, we can further examine the model's confusion matrix and analyze the accuracy heatmap. Table 6 lists the letter grade and the model's prediction results, showing the prediction and misclassifications. When analyzing the generated matrix, I observed that all misclassifications were in the neighboring grade classes. This shows that the model successfully predicted the overall learners' performance trajectory but needed more letter-grade accuracy. This work only included classifying users that completed and passed the course, and the results predicted how well they performed.

The model was performing the best when predicting high performance. All 'A+' instances were labeled correctly (recall 1.00), as seen in Table 6. Furthermore, in most instances, the classification was within the range of the nearest neighbors. Therefore, there may be better models to utilize when predicting the final score, but it is good enough to predict the overall performance. I hypothesize that adding other features and data derived from the system may enhance the accuracy of the results. However, the aim was to investigate the predictability power of the identified behavioral patterns. Such prediction could be used to inform the instructor of the predicted performance before formal assessments and enhance the learner's awareness of their behaviors in the system and the associated performance predictions if they continued to adopt such behavior.

Table 6. Best Performing Model (RF) Accuracy Results.

Grade	**RF Classifier Results**		
	Precision	**Recall**	**F-Score**
A+	**0.875**	**1.000**	**0.933**
A	0.841	0935	0.885
B+	0.333	0.071	0.118
B	0.578	0.813	0.675
C+	0.000	0.000	0.000
C	0.429	0.300	0.353
D	0.571	0.571	0.571

CHAPTER 5

PERSONALIZING SELF-ASSESSMENT EXPERIENCE

The initial release and evaluation of daily quizzes in the introductory programming courses conducted based on the non-adaptive setup in the first and second studies during the fall and spring semester of 2016/2017. In that non-adaptive setup, the daily quizzes were tailored to follow the course syllabus with consecutive attempts. The instructor can adjust the sequence of question as they progress into the semester. The analysis of the outcome from this non-adaptive release and follow-up evaluation (Alzaid, Trivedi & Hsiao, 2017) was keystone in choosing features to be introduced in the adaptive release and the study setup for the following Adaptive third and fourth study. In this chapter, I evaluated the impact of the personalized self-assessment questions on the learners and the self-assessment process. Additionally, I examined how the integration of the new components impacted the activities on the system. To see how the personalization of the recommender impacted the student effort and engagement, the main research question to be answered in this chapter is:

RQ2. How does the inclusion of personalized practice recommender to OSSM platform affect the self-assessment process?

5.1 QuizIT Adaptive Study Set Up and Evaluation

This study setup utilizes the adaptive design in QuizIT system over two semesters (Spring 18 - Fall 18) as third and fourth studies consecutively. Before the learners can access the system, they must initiate their profile for the personalized experience. The learners then proceed with the quiz of the day as before. They can choose to attempt the question at that moment or leave it for later and move to the other features, such as the

dashboard, question history, or calendar. When attempting a question, the reflection procedure is invoked, which interrupts the consecutive attempts. The procedure includes the evaluation of the question, a justification question to promote reflection, and the option to identify whether the answer was simply a guess. If they continue with the reflection procedure, an opportunity to take the recommended question is provided as an incentive. The discussion board and the access to questions interactions from the students are now captures by the system. The system also provides students with an interactive analytical visualization showing their performance and the class performance and progress for each course subject.

Here I evaluated the studies from the non-adaptive and adaptive study setups. In all four studies (first, second, third & fourth), the opportunity to sign up and explore the system was optional throughout the course. There was no association between the formal assessments in the course and the content or self-assessment on QuizIT. Therefore, using the system was always voluntary, with no pressure on the student side. However, the number of formal assessments may have affected how the students utilized the daily learning opportunities, as we will see later in the analysis. The number of available questions was about one hundred questions, of which no questions were posted beyond the final day of the course. With a turnout of more than 750 students participating in the studies, the interest in the course tends to be higher during the fall semester compared to the spring offering. In this section, I will analyze how the different usage of the QuizIT system related to the Quiz performance and the overall course performance from the formal assessment. QuizIT performance is defined as the success ratio of the user's first attempts, while the course performance is based on the final grade from the formal assessments. The course in

all studies had four formal assessments and the grade is based on the best three out of four by dropping the lowest exam grade.

In this chapter, I define the learners as ACTIVE in the system if they have answered ten questions or more, which represent 10% of the available quiz set in the course. If its lower than that, they are considered LOW ACTIVE, while HIGH ACTIVE attempted more than fifty quizzes, which represents 50% of the available question set. I apply the 10% threshold throughout the analysis of the data for other features as well. As with any system, the initial sign-up might be high and gradually drop to a certain percentage of users. In our experience from the studies, this percentage varied between 54% to 22%, which I consider to be the ACTIVE users. Table 7 shows the usage data and the number of participants in each semester.

Table 7. The Breakdown of Four Studies and Participants in Each Study.

USER/STUDY	**First - Fall 16/17**	**Second - Spring 16/17**	**Third - Spring 17/18**	**Fourth - Fall 18/19**
LOW ACTIVE	195	77	98	135
ACTIVE	131	17	53	32
HIGH ACTIVE	38	8	19	8
Attempts	11484	1863	5164 + 983 rcmnd.	2211 + 1001 rcmnd.

5.2 Impact of the Release of the Adaptive QuizIT

The initial noticeable observations were how the students preferred to access the available quizzes and how that preference was impacted by the updated flow in the system design. In figure 4, we can see how the students landed on the questions as it shows the four studies starting from the first one inside and moving outwards to the fourth one. The students in the first study preferred to utilize the question list as the main point of access, while in the second study, due to the low continuous participation, the daily quiz was the

highest access option. To understand this figure, we should examine how the three options were provided in the system design. In all studies, the students will land on the daily quiz as the main point of entry if they have yet to attempt the quiz of the day. In the first two studies, the calendar was shown when the student entered the system or attempted or answered the daily quiz. This design limited the calendar's visibility and made the question list the preferred option to access the question to review and re-attempts. After releasing QuizIT adaptive system, the calendar is in the side panel, which was the case in the latter classroom studies. After using the system's initial design, considering this change from the users' feedback. Here we can see the students' change of preferred access option in the adaptive release. The third and fourth classroom study data show comparable results, in which the students utilized the calendar as the main access option, followed by the list and the daily quiz. Such a result was explainable since the calendar is now always visible. Based on the design principles, the calendar's visibility enables the students to glance at and evaluate their progress and performance. The consistent appearance of the calendar can have a role by guiding the students into self-regulating themselves.

Another aspect was evaluated, how the change in attempt flow affected the accuracy of the first attempts and the willingness to find the correct answer. In the initial release, students always found the correct answer for the questions they attempted, with over 99.6% in the first two studies. However, that number decreased to 91.8% and 85.32% consecutively in the third and fourth studies. This is a result of the change in the adaptive release. The students could no longer immediately re-submit an answer; instead, they were directed to the reflection procedure. However, such a decrease came with a major increase in the accuracy of the attempts. The accuracy, defined as the number of attempts to find

the correct answer, dropped from 2.1 and 1.9 in the first two studies to 1.6 and 1.5 in the latter two studies. This indicates that the students now take the self-assessments more seriously instead of submitting random answers to find the correct one. I wanted to enable the student to capitalize on the positive impact of the reported reflections (Alzaid, Trivedi & Hsiao, 2017). Therefore, in the adaptive release, I increased the visibility of the reflective procedure. The recommender access is also used as an incentive to reflect after each question. This increased the quality and volume of the reflections by folds (Alzaid & Hsiao, 2018). However, it no longer presented the effect it had when the students reflected independently without any pressure to deliver. Figure 5 presents the overall course performance of students who occasionally provide short, thoughtful reflections compared to QuizIT's overall course performance. We can see the case of the first study, where the students could reflect and comment with no incentive provided, and how that choice was significantly associated with higher course performance. While in the third and fourth studies, although the performance of reflectors is higher than the course average, it is less significant than what the first study reported. This leads us to re-evaluate the reflection

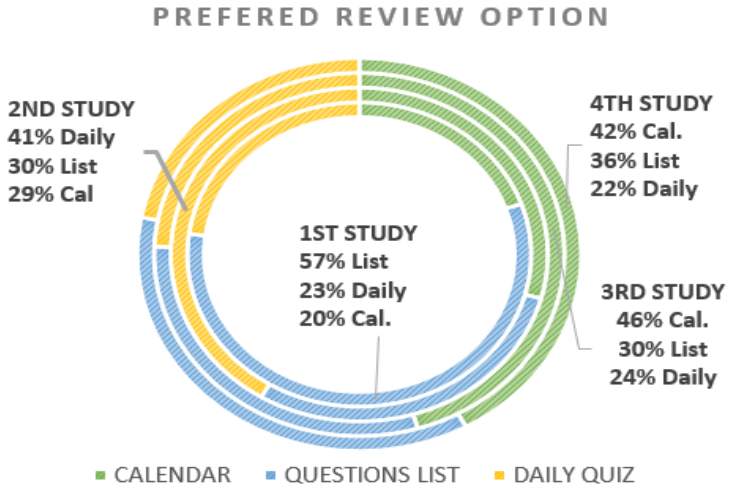


Figure 4: How Students Accessed the Questions to Take the Quizzes, the Studies are Arranged from Inside Out, First to Fourth.

procedure to enhance it by balancing the visibility and the pressure to deliver, which could maximize the learner's gain from such a feature.

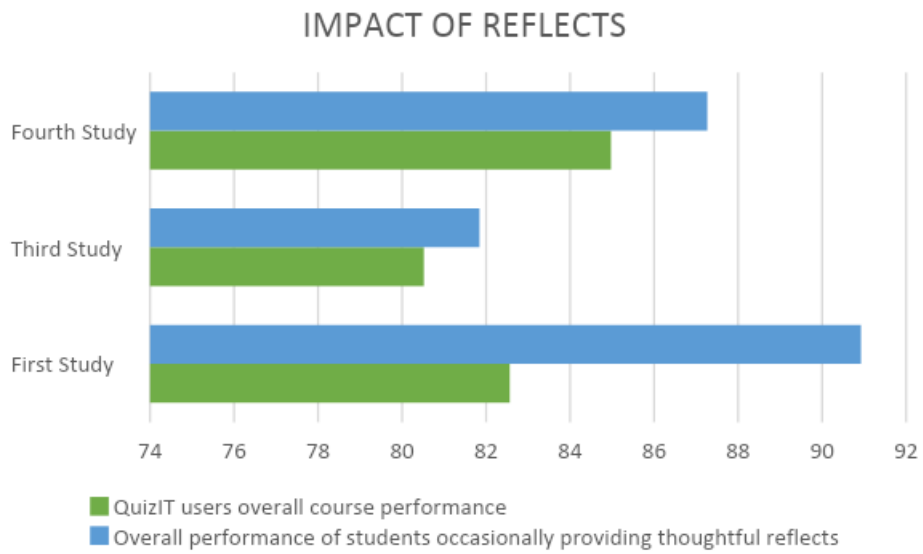
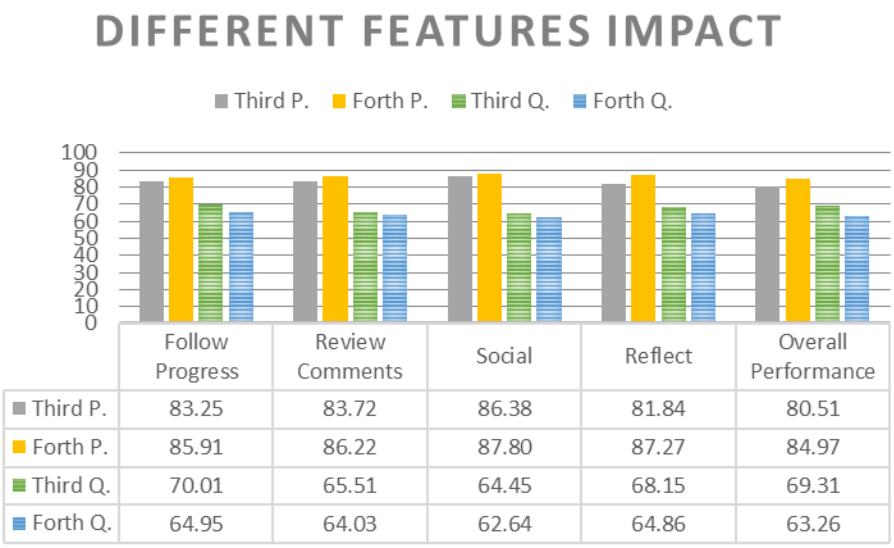


Figure 5: The Comparison Between the Course Performance from Reflective Students in the Studies.

Figure 6 shows our effort to pinpoint the effect of the different enhanced features in the adaptive release. Here, I am using the 10% active threshold for all the features to consider the learner as an active member of that feature group. The results are from the third and fourth studies, where the Third P represents course performance, the Third Q represents QuizIT performance, similarly for the fourth study. The figure shows five separate groups, which I will briefly explain. Follow progress is the number of times the students reviewed their performance chart in the dashboard on different days. Review comment is the number of times the students interacted with the peer's comments section by reading the comments and scrolling through that section. This shows the learner's interest in benefiting from their peers' thoughtful comments and reflections. The social aspect involves few actions, the interactions with the class performance chart, and the interaction with reflections are the number of thoughtful reflections posted by the students

and the number of utilizations of the retry functionality in the system. The overall score represents the class performance average for the active group.

The following progress group had the highest success ratio on the system in both



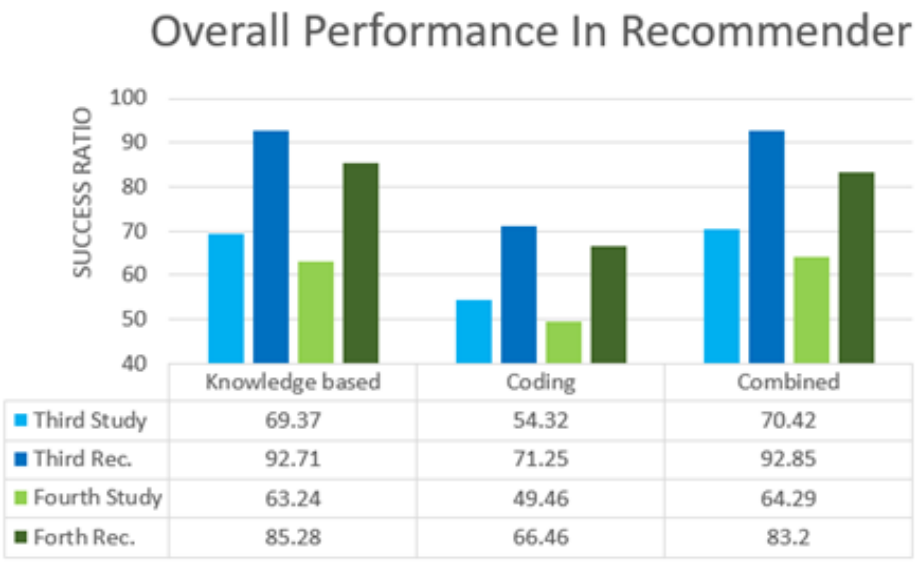
	Follow Progress	Review Comments	Social	Reflect	Overall Performance
Third P.	83.25	83.72	86.38	81.84	80.51
Forth P.	85.91	86.22	87.80	87.27	84.97
Third Q.	70.01	65.51	64.45	68.15	69.31
Forth Q.	64.95	64.03	62.64	64.86	63.26

Figure 6. The Impact of the Different Features in QuizIT Adaptive Release.

courses. They tend to care about how they perform in the quizzes to enhance their charts in the dashboard. The social group had the highest course performance in both studies as well. The social features show the tendency of the students to compare themselves and consistently evaluate the whole class. This is clearly shown by the number of times they interact with the class performance chart with a lack of interest among other groups in such a feature. The student who reviews other peers' comments may not be the highest to performer on the quizzes, as they seek an explanation from other peers. However, as I observed, such action and behavior may have resulted in higher performance in the course compared to most users. The reflection feature gives mixed messages here, which might be the cause of using the recommender as an incentive. When examining how the students interacted with peer comments by answering their questions, upvoting or downvoting their

comments, only a small set of students participated in upvoting in both studies. Most of the votes were positive and seemed to be in the exploration stage of the system. The data was not large enough to report, and it was combined with the review comment group.

About one-fifth of the attempted questions in the third study and one-third in the fourth study came from the recommender engine. This data was separated from the daily quizzes' dataset, as shown in table 7. One way to evaluate the recommender is to examine the QuizIT adaptive release performance on the recommended questions. Figure 7 shows how the students performed on the questions provided by the recommender in comparison to the daily quizzes. The breakdown is based on the question content type as being labeled by the instructor. Third Rec and fourth Rec represent the set of questions provided to the learners by the recommender in the third and fourth studies consecutively and how they performed on them on the first attempt. We can see how the recommended questions outperformed the daily quizzes on all three question categories. The recommender considers the learner's knowledge, past performance, and preference when providing a new question to attempt. It also tries to cover the previously posted questions that the student has yet to view, or attempt based on the student's score in the recommender. While the daily quizzes might be relevant to the current course flow, many students benefited from the personalized recommender ability to locate the questions they are supposed to cover. In this figure, we can also see how the novice students struggled the most with coding questions.



	Knowledge based	Coding	Combined
Third Study	69.37	54.32	70.42
Third Rec.	92.71	71.25	92.85
Fourth Study	63.24	49.46	64.29
Forth Rec.	85.28	66.46	83.2

Figure 7. Students' QuizIT Performance on the Questions in Comparison to the Recommender Performance.

Lastly, I examined how the effect of the student's effort in self-assessing their progress can impact their learning outcome. When comparing how the different activity groups interacted with the available quizzes in the system, we can see a clear difference in the course performance among them. I consider the recommender data as distinguished groups because the recommender effort data was not included before. Here, I label the recommender activity as I did with the daily quiz activity. Students attempting more than 10% questions are labeled as ACTIVE REC, while HIGH REC represents learners with more than 50% of recommended questions. In figure 8, we see how the effort plays a role in the increase of the course performance, and there were four letter grades different between QuizIT releases active users and the students who decided not to explore the system. The performance steadily increases as the learners consume more questions to self-assess their performance.

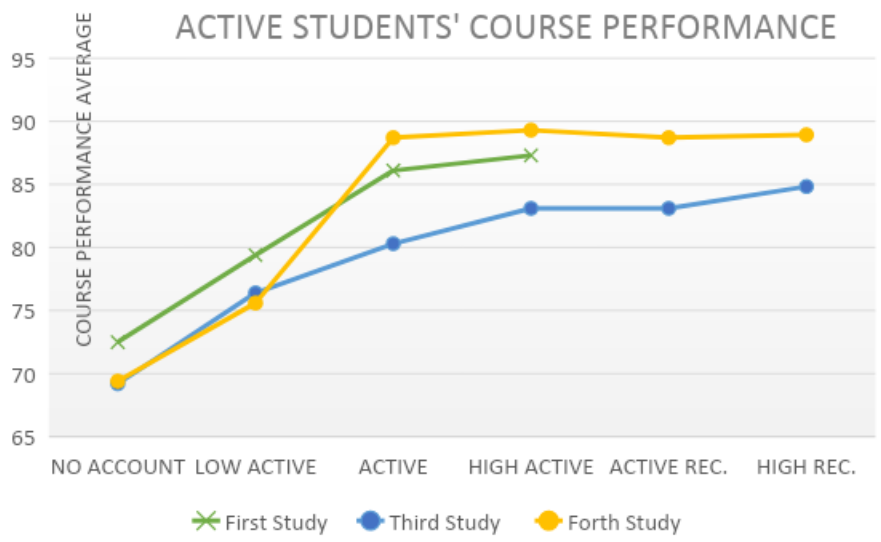


Figure 8. Students' Effort and Course Performance.

5.3 Overall Performance

The important result I need to evaluate is how the different student groups ended up with QuizIT performance. Figure 9 shows how the High-active students always performed lower than the other groups. This result is expected and understandable because the high-active group is exposed to more challenging questions and more concepts. It also shows those learners' effort into continually self-assess their progress. We also noticed that the second study had the lowest success ratio among all the studies and had the lowest interest in using the system. The change in features between the first two and last two studies may have eliminated the high success ratio for low-active students. The last two studies had comparable results when looking into the active and high-active groups.

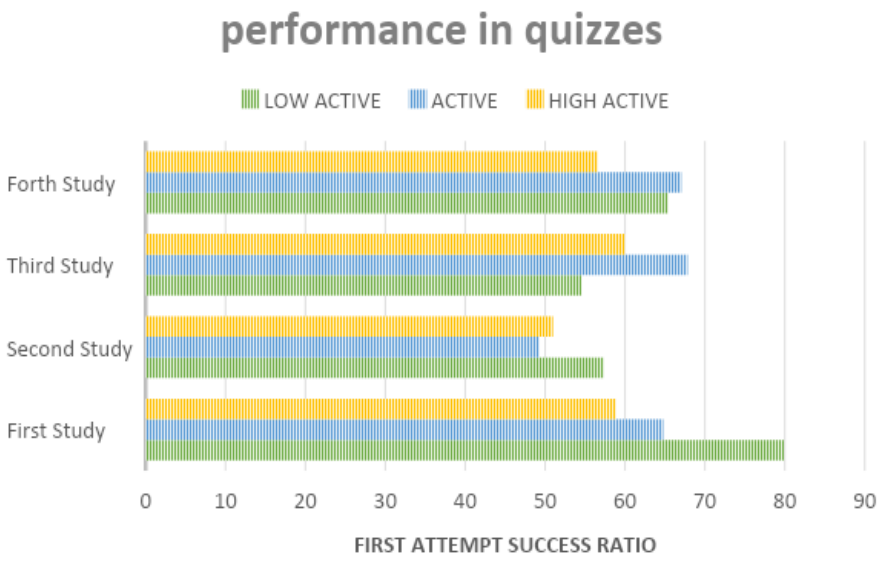


Figure 9: Active Student and Their Performance.

When analyzing how the students interacted which the available questions that were provided daily, we understand how they utilized the system. In figure 10 A&B, based on the number of activities, we can see how the students progressed daily in encountering new questions. These activities include reviews, retries, and consecutive attempts. The major spikes in the initial attempts and activities represent the formal assessments in all four studies as labeled accordingly. This indicates the students' behavior and tendency to self-assess their knowledge as they prepare for an upcoming formal assessment. These spikes usually span over two days, reflecting the time the learners preferred to evaluate themselves before the exam. We also notice that the last exam tends to have a low spike since most of the students have already taken the three earlier exams and chose to skip the final week.

Nothing speaks more clearly when evaluating persistence and self-regulation than the number of days the learner decided to self-assess and attempt some questions. When comparing the overall class performance among the registered users, I found that the group that stands out with the highest average score is the students who came to the system more than ten different days with a variety of gaps between each visit. Here I consider the impact of accessing the system for more than ten days as the effect of persistence.

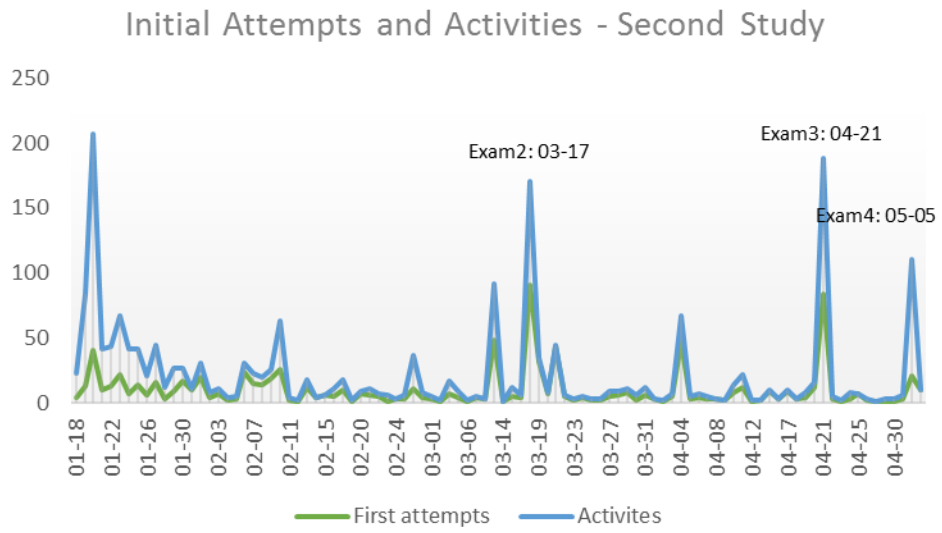


Figure 10A: The system Activities and Attempts Over the Days in Fall 2016.

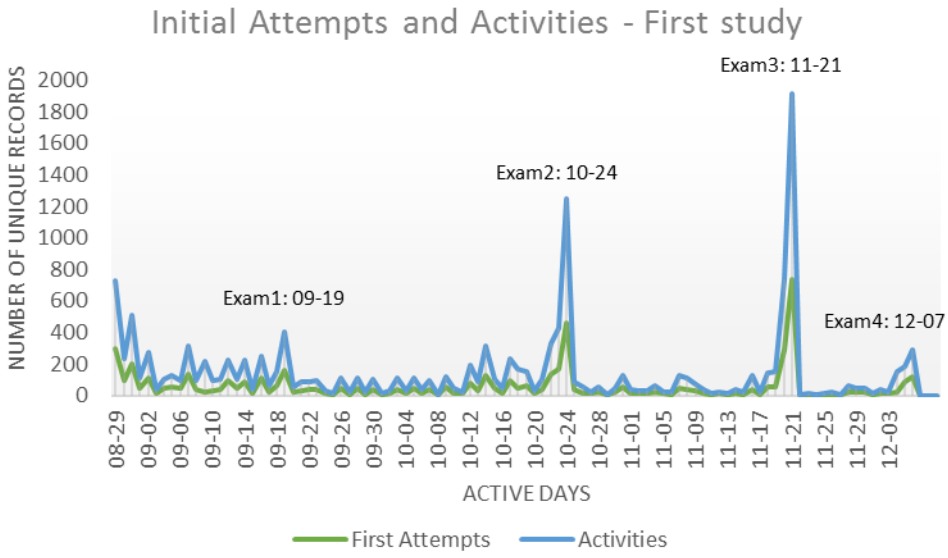


Figure 10B: The System Activities and Attempts Over the Days in Spring 2018.

In the first study, persistence users had an overall performance of 85.82% in comparison to the 82.56% QuizIT users' average. The second study performance was not reported. Third study had, 90.93% compared to 80.51%. Lastly, in the fourth study, 90.08% compared to 84.97% QuizIT users' average. Although the results were not statistically

significant in the studies, the persistence behavior made a difference in the persistence group having at least one letter grade in comparison to the whole class.

In figure 11, we observe the QuizIT performance of all four studies over the course subjects. The subjects here are listed as they were introduced in the class. Here we see that students struggled with Arrays the most, followed by decisions, while performing the highest for Methods, which was introduced later in the course. To understand why some subjects had lower performance in QuizIT, I examined the distribution and classification of the question content. Arrays had the highest code-based questions, which the students struggled with the most. In the decision case, the question content involved previous subjects such as Expression and Variables and had the highest number of challenging questions among all subjects.

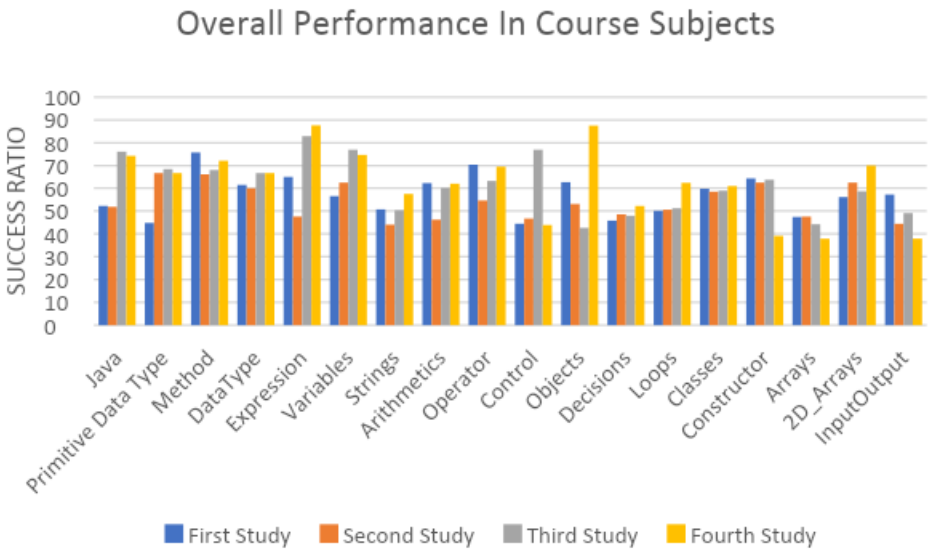


Figure 11. QuizIT Performance of All Four Studies Over Course Subjects.

To understand the learning effect of the available daily quizzes over the course subject, I considered comparing the first question and last question of each subject to its meaning. In figure 12, the last question tends to show more success ratio on the first attempt

than the subject mean in most cases. When I examine the cases where it did not follow that pattern, I found the number of questions may have had in impact. In the case of Primitive data types and constructors, both had the lowest number of available questions, two each in the semester. The limited number of available days to post the questions affected the observation of proficiency in some subjects as well. In the String subject case, the last question was of moderate difficulty and was introduced early, which may have impacted the overall success of that question.

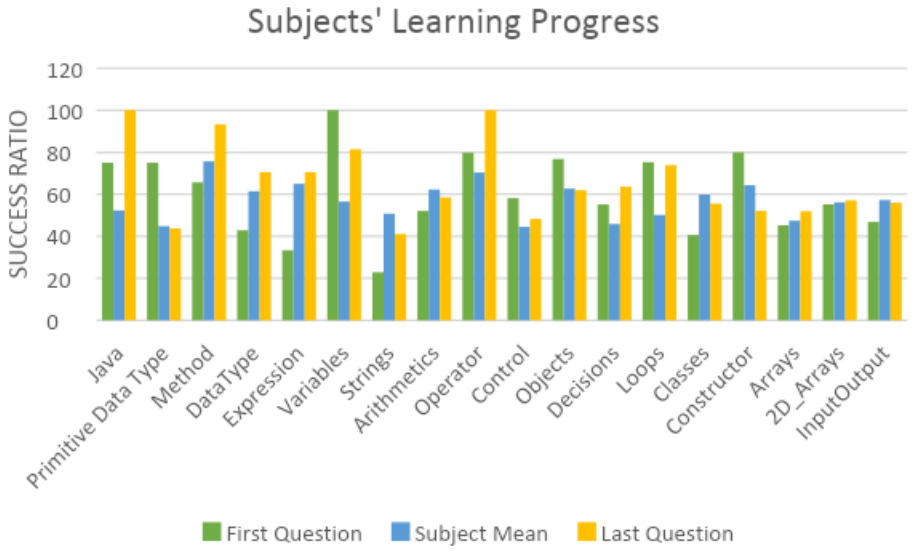


Figure 12. Comparison Between the First Question and Last Question in Each Subject.

CHAPTER 6

ANALYZING DISTRIBUTED REFLECTION OPPORTUNITIES

Utilizing students' reflections is a valuable method in educational settings. It enables the students to make the best of their learning process. It also allows the learners to analyze their thoughts, be self-motivated, increase their metacognition, and strengthen their content knowledge. Reflection may also enhance the learners' evaluation outcome when combined with self-assessment. While the impact of reflection has been addressed in different domains, it requires more attention from the engineering education research community. Initially, I aimed to answer two questions, what triggers the learner's engagement in providing reflections during self-assessments? Moreover, what effect does distributed reflection has on the overall learner performance in programming courses?

My early work indicated that reflections during distributed self-assessments positively correlated with course performance. This focused my interest in exploring further the impact of reflecting quality on the learner's performance. However, I was challenged by the low participation from the learners in the reflection procedure, along with majorly low-quality comments. I worked on enhancements to find a solution that would trigger more guided participation from the students and encourage SRL behavior.

6.1 Reflection Study Set Up

This study setup utilizes the enhanced reflective release in QuizIT over two semesters (Fall 20 – Spring 21) as the fifth and sixth studies. The study focuses on the effect of students' reflections as they progress into the course, with the addition of the weekly reflections set at the first lecture of the week. Like the non-adaptive study setup, all the students will have access to the QuizIT version without the adaptive recommender.

The objective is to study the learning effect of weekly reflections on student performance and engagement. Also, to validate the effect of the reflection procedure in QuizIT from the earlier studies and find out how students of different practice performances utilize the weekly reflection procedure.

The reflective release studies were conducted on two programming courses at Arizona State University using the QuizIT reflective release over two semesters. The fifth study was a pilot study, followed by the sixth study in the next semester. In this chapter, the sixth study data is being analyzed with a partial analysis of reflections from fifth study. The study format followed prior studies setups with a focus on the reflection feature. The students were informed of this study at the beginning of the semester and provided their consent for participation in the data collection process. The study design considered providing a focused weekly reflection opportunity on the first day of the lecture week. This reflection-prompted opportunity will be the first thing the students will encounter when they access the system on that day. To illustrate the sequence for this weekly reflection setup:

- The students will be asked to report their level of understanding on the weekly concept that was tailored to follow the course progress. The students are notified that there is no correct answer to this question. To report their level of understanding, they are asked the following options “This is an opportunity to summarize what you've learned and understood on Java concept, please choose one of the following options: “
    - I fully understood the concept.
    - I partially understood the concept.

    - I did not understand the concept.
    - I am not sure.
- Once they submit their answer, they will be prompted to reflect on what they understood on that concept.
- Finally, they will be able to review their peers' reflections on that concept. It is important to note that, without reflecting on that prompt, the learner will not be able to review others' comment.

A total of sixty-three students taking the course participated in this study. The participation was optional, with no additional incentive provided. In addition, the participants were given the option to withdraw at any point without impacting their grades. The study did not consider the participants' demographic data, gender, or background. To be consistent, the data is filtered for students with at least two days of activity, or more than 10% questions answered as applied in prior studies. This is done to focus on the data of the study participants who had enough time to utilize the system for their practice and self-assessment purposes. After the preprocessing of the data, I end up with (N=38) students.

6.2 Effect on Students' Performance

Based on the student responses to the reflection opportunities, I was able to classify them into four groups, weekly with response N=16, the students in this group answered the weekly reflections and provided their written reflection; weekly without response N=11, the students answered the reflection questions only without providing their own reflection content; there were students within that with reflections on daily questions only N=7, and no reflections N=11, where the students answer the daily questions only while avoiding the

weekly reflections. Based on QuizIT average performance of first attempts, I found that students who engaged in the reflection procedure outperformed their peers with no reflection. Figure 13 shows the students' performance in the reflective release based on how they differ in their performance. All reflective groups had higher performance than non-reflective students. The daily reflection group showed the highest success rate as seen in figure 13.

On each weekly reflection opportunity, I ask the students to report their level of understanding of the concept of the week. In Figure 14B, we can see students' responses. Almost 51% of the students reported that they fully understood the concept, 30% reported they partially understood the concept and only 5% reported they were not sure. I was more interested in how their performance will be impacted after that reported level of understanding. Figure 14A shows the success rate of the subsequent attempts after the reported level of understanding. It clearly shows that what the students have reported is reflected in their performance on the self-assessment.

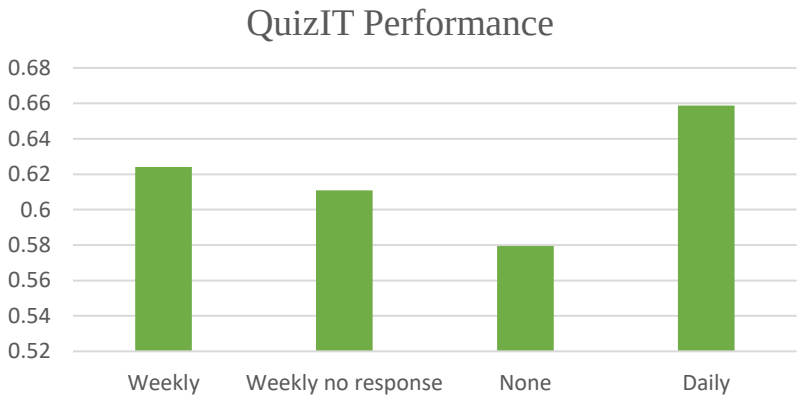


Figure 13. Learners' QuizIT Performance Based on Reflections

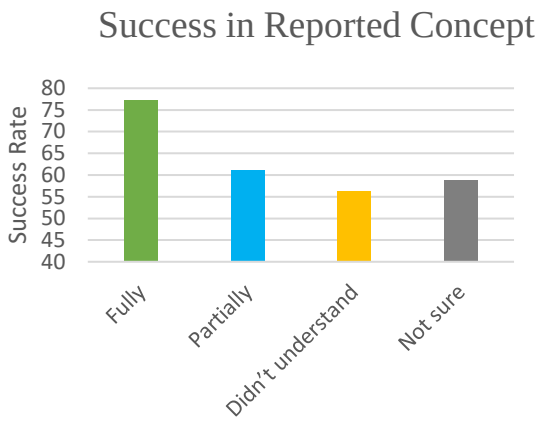


Figure 14A. Shows Student Performance After the Reported Understanding Level

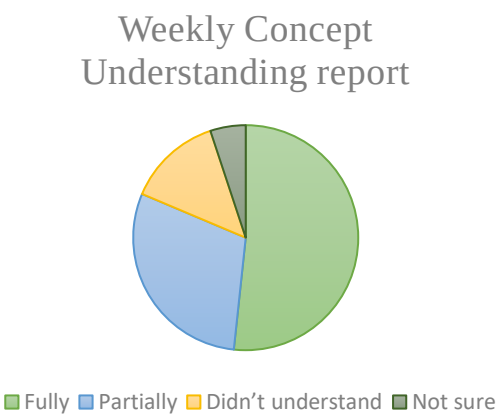


Figure 14B. Student Response to Weekly Reflections

6.3 Effect on Learners Behavior

To see how often each group took a practice session, I considered the average number of days each student in the group practiced. In figure 15, the reflection groups had roughly five to six practice sessions in the semester, while the no-reflection group only had two and a half sessions. Thus, reflective students tend to practice more and demonstrate better SRL abilities.

To further investigate the differences between the two major reflection groups, weekly response vs. daily response, an independent sample t-test was used. It did not show a significant difference between the groups' performance based on their successful attempts. I also examined whether reflection does have an immediate impact on the practice session. Using the reflected point and the result of the following answer attempt. Weekly reflection is more likely to be consecutively followed by correct attempts (76.4%) than daily reflection opportunities (60.1%). This observation could indicate that students who participate in those opportunities tend to pay more attention during the answer sessions.

When examining the weekly reflection response to the level of understanding, students were asked to report their level of understanding through a question. Those who chose "I fully understand the concept" were labeled as the "Mastering" group. On the other hand, students who responded with any other choice were in the "Developing" group. Figure 15 shows a closely split between the Mastering group (51.6%) and the Developing group (48.4%). The same trend followed with an even split for students who responded with their own words and those who did not (50%). However, when looking at the reflection trigger for other reflection opportunities, most students' reflections came after submitting the correct answer, as seen in figure 16.

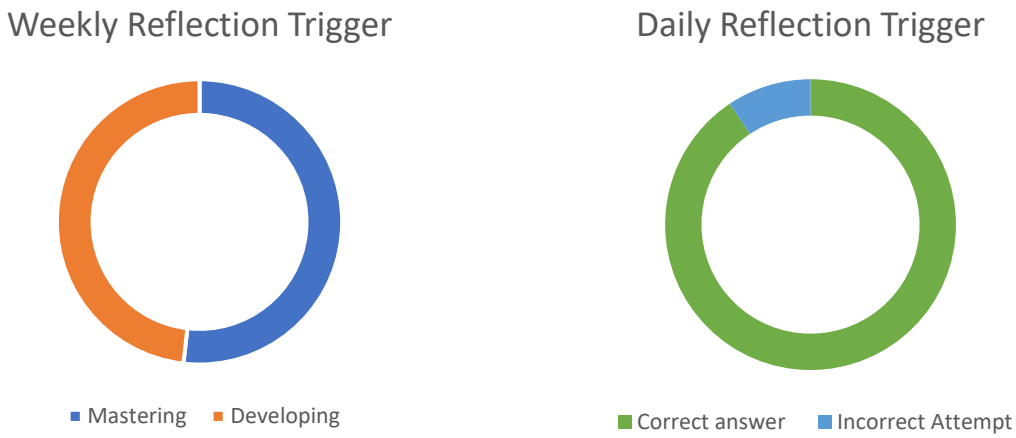


Figure 16: Reflection Triggers Based on the Result of the Answer Attempt, and Their Response to the Weekly Reflection Opportunities.

Figure 17 shows the distribution of the attempts and reflections throughout the study. I wanted to see any noticeable difference in the trajectory of each action. Here we see the attempts of the questions, daily reflections, and weekly reflections. Looking at the chart, we cannot see a clear distinction between them besides a slight increase in the weekly

reflection before the second midterm. We note that the exams were in week 5 and 13.

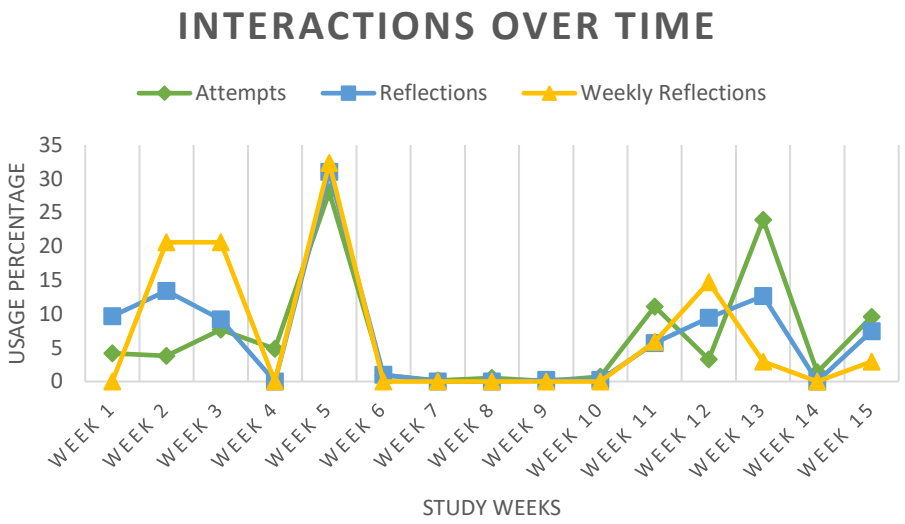


Figure 17. System Interactions and Answer Attempt Overtime.

Figure 18 shows the distribution of whom answered the distributed questions in the system. Here we see that the weekly reflection group attempted the most questions in the system. They were followed by the weekly no-response group. This shows that students who utilized the weekly reflection opportunities had better self-regulating skills than their peers.

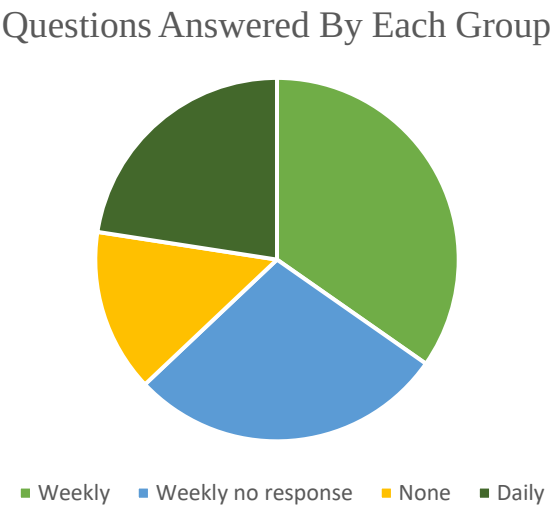


Figure 18. Distribution of Questions Answered Between Different Groups.

6.4 Reflection Classification

To classify the student's data while responding to the reflective prompts, I examined the generated dataset and attempted to classify it as reflective and non-reflective classes in two phases. To establish a baseline for the classification, I investigated the existence of sentiment in the student responses. For that purpose, I analyzed the dataset for the statement expression and labeled each comment accordingly. Most of the responses were short statements (8.4 words on average). After preparing and cleaning the dataset, based on the short review lexicon (Hu & Liu, 2004), I labeled the short responses as (reflect, comment) as it matches the lexicon. This approach resulted in 32% of the dataset to be labeled as reflections. I then evaluated the reliability of this process by training a classifier model and using RF Weka implementation (Witten & Frank, 2002), with 80% of the dataset and testing on the remaining 20%. The results at the end of this report show that RF model was only successful in labeling comments and could not detect reflective responses.

In the second phase of the classification process, I aimed to establish a rule-based classifier that considers the response's linguistic and knowledge elements. Based on the reflective writing indicators and reflective writing model proposed by Ryan (Ryan 2011), I adopted the following features to use in the classification objective:

- **(F01) First person voice:** Reflective writing usually use the first-person voice (Lindsay, et al 2010).
- **(F02) Thinking and sensing verbs:** The use of verbs that shows learning and thinking such as feel, believe, understand, consider, etc.

- **(F03) Domain knowledge**: The analysis of effective academic reflections shows its usually contain domain-specific language. (Ryan 2011)
- **(F04) Comparison language:** The use of comparison words is evidence of more effective reflections (Luk, 2008).
- **(F05) Reasoning and explanation:** Use of words (i.e., so, therefore, because) enhances the reflection thought process (Ryan 2011).
- **(F06) Temporal link and future tense:** For the reflective writing model, Ryan provided a list of temporal link and future tenses usually found in students reflections (Ryan 2011).

Based on the six features, I built lexicons with a list of words for each feature (non-exhaustive) to set as evidence of the use of the feature in the student response. I then labeled the student response as reflecting when at least it contains half of the identified linguistic features. This classification approach resulted in labeling 14% of the response as reflections.

To get a sense of how the student reflects, I will give an example of a labeled reflective and non-reflective comments. The following responses are examples of a reflective response using this approach:

"I partially understand loops. I am comfortable with loops regarding creating menus and storing data. I am also confident with nested loops. The only problem have is choosing the right loop to solve the problem I'm facing.".

"I am not sure how to explain what I know about loops, but I am familiar with while and for loops For loops are used more often when dealing with arrays and while loops are more flexible to use for other purposes".

Example of a labeled non reflective response:

• "knowing all the information that needed to be included and where certain items go".

• "first time learning this information".

Given the context of the reflection, the short format was overwhelmingly present in the generated content. The weekly reflections were able to provide an opportunity to enhance the reflection quality which can be seen in the classification results.

I then trained the RF classifier model to evaluate this approach using 80% of the data. Although there was a slight improvement using the rule-based approach in comparison to the sentiment approach, it still shows that being able to classify the short reflections in this dataset automatically is a challenging task.

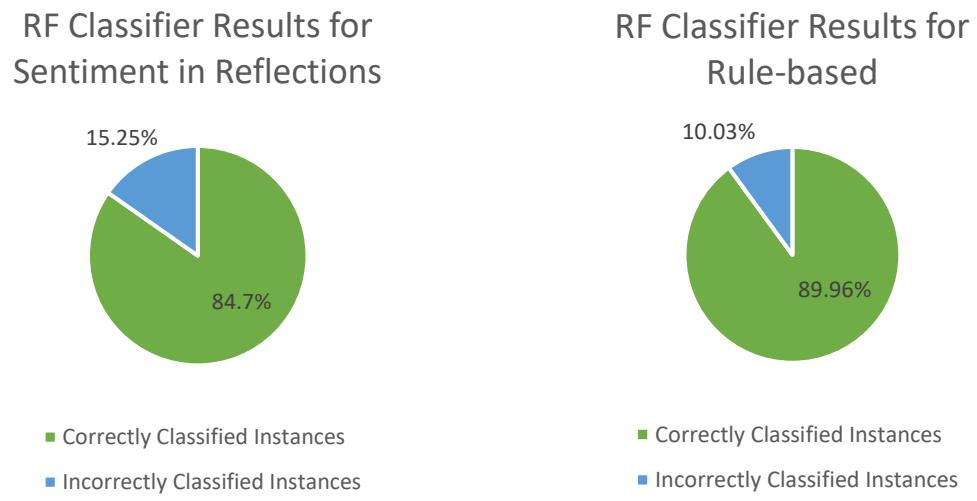


Figure 19. Result of Classifying Reflections Based on Sentiment and Rule-Based Method

From this analysis, I believe that the rule-based approach might be the best fit for this dataset which can be enhanced by considering features that address the short form of reflection as presented in this dataset, such as length and exhaustive list identifiers in each of the used features.

CHAPTER 7

DISCUSSIONS

7.1 Summery

In this dissertation, I designed a learning system to provide the learners with daily opportunities to self-assess their knowledge in programming courses. The development of the system took three main phases of releasing features and assessing the impact in classroom studies. In all the classroom studies, QuizIT was provided as an optional learning recourse to the participating classes. The first release of QuizIT was non-adaptive, focusing on providing bite-size MCQ questions and free form of reflection. Study results showed a positive impact on learners' effort which associated with higher performance on the learners who chose to utilize it. The results and users' feedback encouraged the second release of QuizIT design to be adaptive, which included OSSM and personalized learning experience. The adaptive QuizIT release included an enhanced reflection procedure. The personalized learning experience was based on the student knowledge, performance, and peers' evaluation of questions. The students were able to access that feature once they evaluated the daily questions or posted their reflection or review the peer reflection on the discussion board. After this release evaluation, which showed the positive impact of the personalization of self-assessments, I wanted to focus on studying the impact of reflection in the self-assessment process. The reflective QuizIT release provided weekly opportunities of reflection, which follows the course topic progress. This release was non-adaptive and included the daily justification reflection prompt. Each release was followed and evaluated by two consecutive classroom studies. In Table 8, I summaries the results, impact, and implications from each release and the

associated set up and studies. The non-adaptive release had consecutive attempts enabled which showed the learners tendency to adopt a trial-and-error approach. In the following releases, the consecutive attempts are disrupted by the reflection procedure. From the adaptive release, I saw the interest in personalized self-assessment opportunities which led to an increased engagement in the system. The reflective release impacted the reflection quality and demonstrated how reflective learners had better SRL.

Table 8. Summary of QuizIT releases and studies results, impacts, and implications.

Study Summary	**Non-Adaptive** ***Ch4: 1st & 2nd Study***	**Adaptive** ***Ch5: 3rd & 4th study***	**Reflective** ***Ch6: 5th & 6th study***
Results	• Identified distinct patterns of self-assess. • Predicted learners performance based on self-assess behavior.	• Significant increase in attempted practices from recommended questions. • Increased reflection participation. • Evaluated the impact of the OSSM.	• Reflective learners were more active. • Reflection quality differs between weekly & daily reflection opportunites. • Classified the reflections content.
Impacts	• Practice & reflection enhance the performance. • Self-assess in sessions enabled SRL. • Self-assessment utilized by self-motivated learners.	• Personalization enhanced the learner's engagement. • Enhanced features improved the performance. • Incentive reflection decreased its impact.	• Weekly reflections increased the reflection quality. • Reflective learners demonstrated SRL. • Reflective learners had better performance.
Implications	• Consective attempts associated with trial & error approach. • Ineffective trial & error can be harmful.	• Personlizing self-assess experience shows to be benefitial to learners. • Reflection utilization can be enhanced with limiting consective attempts.	• Promoting reflection requires guidance. • Rule based short reflection classification is recommended.

7.2 Finding and Takeaways

Going through the iteration of system releases the main objective was to address research questions which eventually formulized in the following three main research questions. The first research question asks about whether we can predict the learner's performance based on the behavior shown during the self-assessment session.

RQ1. Can self-assessment behavioral trails be used to predict student's performance?

To answer this question, I utilized the student data from the non-adaptive QuizIT release. When I looked at how the students spent time working on self-assessment, I found a positive correlation between the number of sessions and their overall performance. I also found distinct behavioral patterns from the self-assess attempt sequences. The affirmative behavior which always find the correct answer corelates with good performance. However, there were indicators of disengaging behavior that was correlated with low performance such as when the learner randomly seeks the answer or stop seeking the correct answer. From there I was able to utilize the attempts patterns to predict the overall performance in the course.

The second research question was addressed in chapter 5 and asked about how the personalized practice recommender in QuizIT platform affects the self-assessment experience and process for the learners.

RQ2. How does the inclusion of personalized practice recommender to OSSM platform affect the self-assessment process?

After I integrated the recommendation system to QuizIT, I conducted the subsequent studies on the QuizIT adaptive release. From that, I noticed a significant

increase in the system usage which was observed in the number of questions that were attempted by the students. The recommender was able to provide the students with questions that considered the learners knowledge, performance, and preferences. This resulted in enhancing students' self-assessment results and encouraged the students to continue evaluating themselves through the recommender feature.

The third research question was addressed in chapter 6 and asked about the reflective questions' impacts on student behavior and performance. This question focused on the weekly reflection study from the reflective release and compares the daily and weekly reflection opportunities.

RQ3. What are the embedded reflective questions' impacts?

a. On student's behavior?
b. On student's performance?
c. Differences between embedded reflective questions (weekly) vs daily questions? Are weekly reflections more effective?

From the non-adaptive release, I saw that students tend to reflect on the question once they get the correct answer. I wanted to further understand this behavior, from which I designed and released reflective QuizIT and conduct the subsequent classroom studies. Which led me to formalize and answer this research question. I found that most of the attempts were done by the reflective students which indicates that the reflective learners are more active and seek to continuously self-asses their knowledge. When I evaluated the performance, I found that learners who engaged in the reflection procedure outperformed their peers with no reflection. Additionally, when I investigated the distribution of weekly

reflections over the course of the study, I saw that it was consistent and followed other action trajectory such as, the attempts and daily reflections.

During the reflection procedure, I asked the learners to report their level of understanding. The students' responses clearly showed that the reported levels were reflected in their performance in the self-assessment and the students were able to identify their knowledge level in the subject. Thus, it is important that the instructor consider the learners feedback when providing practice to the learners.

Finally, when looking at the differences between embedded reflective questions (weekly) vs daily questions, weekly reflections enabled the student to reflect better. When considering the effectiveness of weekly reflections, weekly reflections are more capable of capturing reflective features than the daily reflections. This enables us to analyze the reflection content and evaluate the learner's knowledge and understanding.

7.3 Lesson Learned

There were findings that were consistent throughout all the releases and classroom studies. Students preferred to utilize the self-assessment opportunities in sessions and as preparation as formal assessment. The system activity noticeably decreases after the first month of the release, which usually coincide with the first formal assessment. Then it picks up as the second formal assessment approaches. From analyzing how often students engage and the pattern of self-assessment they had, I was able to predict their performances and identify alarming signs that may hinder their progress. As for reflection, the learners showed that they were not able to reflect on their own when given a free form of reflection. They required an assistant to practice and benefit from reflections. As for personalization,

learners preferred to engage more in the personalized questions. Students who showed SRL behavior demonstrated higher engagement with the system and the self-assessment process. With no incentive of pressure to perform, we saw how self-motivated students represented a larger subset of the data set.

CHAPTER 8

CONCLUSION

This dissertation focusses on enhancing the learners' self-assessment experience and understanding how it impact their learning. To achieve that goal, I chose and evaluated different elements that goes into the self-assessment sessions. I designed QuizIT system, to support that cause and conducted a series of classroom studies. Within QuizIT system, I provided the learners with distributed self-assessment and reflection opportunities from which I observed how the learners engaged with these opportunities and impacted their learning goals. I also designed, integrated the personalized experience, and evaluated how the student reacted to that. I saw how this integration positively enhanced the students' performance in the system as well as the course in general. Additionally, with the recommender system integration, I was also able to show the learners' preferences and increase in engagement from the personalized self-assessments. However, some aspects of the system gave mixed messages. When I examined the engagement in the system, I saw that it increased in the reflection procedure but the interest in finding the correct answer has decreased. The reflect procedure also mitigated the significant impact of the open reflections from the initial release. The latest study focused on the student reflection throughout the semester by setting a point for them to reflect thoroughly on a specific topic, as a weekly reflection design. I anticipated having enhanced utilization for the weekly reflections which would confirm the positive impact reported in the initial release. The weekly reflections were able to show that students who utilized that opportunity were more engaged with the system and showed better overall performance.

8.1 Limitations

One of the main limitations I faced during the studies is that, even though QuizIT was designed to provide daily opportunities, I was not able to measure the impact of the daily assessment. As I anticipated and later found, learners preferred to do practice in sessions. It was also hard to measure the learning impact of the system because I had no control over the available sources of practice that the students could utilize. I wanted to capture the natural behavior of the students, however, since the usage of the system was voluntarily, the number of users varies between studies and the participation drops after the first month. To enhance the turnout of active users in future studies, I would strengthen the link between the system and the course. In an experimental controlled study, I significantly enhanced participation by utilizing a subset of QuizIT questions as in-class quizzes. When it comes to reflections, I was faced with a challenge of limited feedback from the learners on both the quality and volume of reflection. Most students had limited participation in the reflection procedure. I evaluated different approaches to promote reflection and increase learners' engagement in it which resulted in improving the reflection quality.

8.2 Contributions of The Dissertation

The contribution of this dissertation came from different fronts. First, the design and implementation of the QuizIT system including all releases. To this day, QuizIT versions have been used in twenty-four courses in different countries with over 1800 learners benefited from it. It helped generate data that enables researchers to advance their research objectives, such as the questions dataset which included over two thousand questions and the learners attempts which was used in the AI-assisted programming

question generation dissertation by Cheng-Yu Chung (2022). Another contribution is the evaluation of personalized self-assessment, which showed the value of such features for similar systems. Also, in identifying distinct self-assessment patterns and demonstrating how the student's behavior can be used to predict student's performance. Lastly, this dissertation showed how setting the reflecting prompts impacted the self-assessment process and how different reflection methods impacted the learner choice to reflect.

REFERENCES

Altadmri, A., & Brown, N. C. (2015, February). 37 million compilations: Investigating novice programming mistakes in large-scale student data. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 522-527).

Alzaid, M., Trivedi, D., & Hsiao, I. H. (2017, October). The effects of bite-size distributed practices for programming novices. In 2017 IEEE Frontiers in Education Conference (FIE) (pp. 1-9). IEEE.

Alzaid, M., & Hsiao, I. H. (2018, October). Effectiveness of Reflection on Programming Problem Solving Self-Assessments. In 2018 IEEE Frontiers in Education Conference (FIE) (pp. 1-5). IEEE.

Aronson, L. (2011). Twelve tips for teaching reflection at all levels of medical education. Medical teacher, 33(3), 200-205.

Azcona, D., & Smeaton, A. F. (2017, September). Targeting at-risk students using engagement and effort predictors in an introductory computer programming course. In European Conference on Technology Enhanced Learning (pp. 361-366). Springer, Cham.

Baernstein, A., & Fryer-Edwards, K. (2003). Promoting reflection on professionalism: a comparison trial of educational interventions for medical students. Academic Medicine, 78(7), 742-747.

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In Learning analytics (pp. 61-75). Springer, New York, NY.

Bernard, A. W., Gorgas, D., Greenberger, S., Jacques, A., & Khandelwal, S. (2012). The use of reflection in emergency medicine education. Academic Emergency Medicine, 19(8), 978-982.

Benjamin, A. S., & Tullis, J. (2010). What makes distributed practice effective? Cognitive psychology, 61(3), 228-247.

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. AcM SIGcSE Bulletin, 39(2), 32-36.

Bernardini, A., & Conati, C. (2010, June). Discovering and recognizing student interaction patterns in exploratory learning environments. In International Conference on Intelligent Tutoring Systems (pp. 125134). Springer, Berlin, Heidelberg.

Blikstein, P. (2011, February). Using learning analytics to assess students' behavior in open-ended programming tasks. In Proceedings of the 1st international conference on learning analytics and knowledge (pp. 110-116). ACM.

Blasco-Arcas, L., Buil, I., Hernández-Ortega, B., & Sese, F. J. (2013). Using clickers in class. The role of interactivity, active collaborative learning and engagement in learning performance. Computers & Education, 62, 102-110.

Boyer, K. E., Phillips, R., Ingram, A., Ha, E. Y., Wallis, M., Vouk, M., & Lester, J. (2011). Investigating the relationship between dialogue structure and tutoring effectiveness: a hidden Markov modeling approach. International Journal of Artificial Intelligence in Education, 21(1-2), 65-81.

Boud, D., Keogh, R., & Walker, D. (Eds.). (2013). Reflection: Turning experience into learning.Routledge.

Braun, K. W., & Sellers, R. D. (2012). Using a "daily motivational quiz" to increase student preparation, attendance, and participation. Issues in Accounting Education, 27(1), 267-279.

Brusilovsky, P., Hsiao, I. H., & Folajimi, Y. (2011, September). QuizMap: open social student modeling and adaptive navigation support with TreeMaps. In European Conference on Technology Enhanced Learning (pp. 71-82). Springer, Berlin, Heidelberg.

Brusilovsky, P., Somyürek, S., Guerra, J., Hosseini, R., & Zadorozhny, V. (2015, June). The value of social: Comparing open student modeling and open social student modeling. In International conference on user modeling, adaptation, and personalization (pp. 44-55). Springer, Cham.

Bull, S. (2004). Supporting learning with open learner models. Planning, 29(14), 1.

Buffardi, K., & Edwards, S. H. (2013, August). Effective and ineffective software testing Behaviors by novice programmers. In Proceedings of the ninth annual international ACM conference on international computing education research (pp. 83-90). ACM.

Butler, D. L., & Winne, P. H. (1995). Feedback and self-regulated learning: A theoretical synthesis. Review of educational research, 65(3), 245-281.

Chen, Y., Yu, B., Zhang, X., & Yu, Y. (2016, April). Topic modeling for evaluating students' reflective writing: a case study of pre-service teachers' journals. In Proceedings of the sixth international conference on learning analytics & knowledge (pp. 1-5).

Cheng-Yu Chung, I-Han Hsiao & Yi-Ling Lin (2022) AI-assisted programming question generation: Constructing semantic networks of programming knowledge by local knowledge graph and abstract syntax tree, Journal of Research on Technology in Education, DOI: 10.1080/15391523.2022.2123872

Deci, E. L., Schwartz, A. J., Sheinman, L., and Ryan, R. M. (1981). An instrument to assess adults' orientations toward control versus autonomy with children: reflections on intrinsic motivation and perceived competence. J. Educ. Psychol. 73, 642–650.

Denny, P., Hanks, B., & Simon, B. (2010, March). Peerwise: replication study of a student-collaborative self-testing web service in a US setting. In Proceedings of the 41st ACM technical symposium on computer science education (pp. 421-425).

Dimitrova, V. (2003). STyLE-OLM: Interactive open learner modelling. International Journal of Artificial Intelligence in Education, 13(1), 35-78.

Edwards, S. H. (2004, March). Using software testing to move students from trial-and-error to reflection in-action. In ACM SIGCSE Bulletin (Vol. 36, No. 1, pp. 26-30). ACM.

Edwards, S. H., & Perez-Quinones, M. A. (2008, June). Web-CAT: automatically grading programming assignments. In ACM SIGCSE Bulletin (Vol. 40, No. 3, pp. 328-328). ACM.

Fekete, A., Kay, J., Kingston, J., & Wimalaratne, K. (2000). Supporting reflection in introductory computer science. ACM SIGCSE Bulletin, 32(1), 144-148.

Gibbs, G. (1988). Learning By Doing, a Guide to Teaching and Learning Methods. Further Education Unit Oxford Polytechnic.

Gibson, A., Aitken, A., Sándor, Á., Buckingham Shum, S., Tsingos-Lucas, C., & Knight, S. (2017, March). Reflective writing analytics for actionable feedback. In Proceedings of the seventh international learning analytics & knowledge conference (pp. 153-162).

Guerra, J., Sahebi, S., Lin, Y. R., & Brusilovsky, P. (2014). The problem-solving genome: Analyzing sequential patterns of student work with parameterized exercises.

Gleaves, A., Walker, C., & Grey, J. (2008). Using digital and paper diaries for assessment and learning purposes in higher education: a case of critical reflection or constrained compliance?. Assessment & Evaluation in Higher Education, 33(3), 219-231.

Hacker, D. J., Dunlosky, J., & Graesser, A. C. (Eds.). (1998). Metacognition in educational theory and practice. Routledge.

Hartwig, M. K., & Dunlosky, J. (2012). Study strategies of college students: Are self-testing and scheduling related to achievement?. Psychonomic Bulletin & Review, 19(1), 126-134.

Hoepner, J. K., Sather, T. W., Homolka, T., & Clark, M. B. (2021). Immersion learning at an aphasiacamp: Analysing student video reflections. International Journal of Speech-Language Pathology, 1-11.

Hollingsworth, H., & Clarke, D. (2017). Video as a tool for focusing teacher self-reflection: Supporting and provoking teacher learning. Journal of Mathematics Teacher Education, 20(5), 457-475.

Hsiao, I. H., Sosnovsky, S., & Brusilovsky, P. (2010). Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. Journal of Computer Assisted Learning, 26(4), 270-283.

Hsiao, I.H., Sosnovsky, S. and Brusilovsky, P., 2010. Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. Journal of Computer Assisted Learning, 26(4), pp.270-283.

Hsiao, I. H., Bakalov, F., Brusilovsky, P., & König-Ries, B. (2013). Progressor: social navigation support through open social student modeling. New Review of Hypermedia and Multimedia, 19(2), 112-131.

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., ... & Rubio, M. Á. (2015, July). Educational data mining and learning analytics in programming: Literature review and case studies. In Proceedings of the 2015 ITiCSE on Working Group Reports (pp. 41-63). ACM.

Jackson, D., & Usher, M. (1997, March). Grading student programs using ASSYST. In ACM SIGCSE Bulletin (Vol. 29, No. 1, pp. 335-339). ACM.

Jadud, M. C., & Dorn, B. (2015, July). Aggregate compilation Behavior: Findings and implications from 27,698 users. In Proceedings of the eleventh annual International Conference on International Computing Education Research (pp. 131-139). ACM.

Karpicke, J.D. and Aue, W.R., 2015. The testing effect is alive and well with complex materials. Educational Psychology Review, 27(2), pp.317-326.

Kann, V., & Högfeldt, A. K. (2016, February). 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 (pp. 510-515).

Kember, D., McKay, J., Sinclair, K., & Wong, F. K. Y. (2008). A four-category scheme for coding and assessing the level of reflection in written work. Assessment & evaluation in higher education, 33(4),369-379.

Kovanović, V., Joksimović, S., Mirriahi, N., Blaine, E., Gašević, D., Siemens, G., & Dawson, S. (2018,March). Understand students' self-reflections through learning analytics. In Proceedings of the 8th international conference on learning analytics and knowledge (pp. 389-398).

Law, L. C. (1998). A situated cognition view about the effects of planning and authorship on computer program debugging. behavior & Information Technology, 17(6), 325-337.

Lew, D. N. M., & Schmidt, H. G. (2011). Writing to learn: can reflection journals be used to promote self-reflection and learning?. Higher Education Research & Development, 30(4), 519-532.

Lew, M. D., & Schmidt, H. G. (2011). Self-reflection and academic performance: is there a relationship?. Advances in Health Sciences Education, 16(4), 529.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., ... & Simon, B. (2004, June). A multi-national study of reading and tracing skills in novice programmers. In ACM SIGCSE Bulletin (Vol. 36, No. 4, pp. 119-150). ACM.

Lu, Y., & Hsiao, I. H. (2016). Seeking Programming-Related Information from Large Scaled Discussion Forums, Help or Harm?. International Educational Data Mining Society.

M. Dorodchi, A. Benedict, D. Desai, M. J. Mahzoon, S. MacNeil and N. Dehbozorgi, "Design and Implementation of an Activity-Based Introductory Computer Science Course (CS1) with Periodic Reflections Validated by Learning Analytics," 2018 IEEE Frontiers in Education Conference (FIE), 2018,pp. 1-8, doi: 10.1109/FIE.2018.8659196.

McCrindle, A. R., & Christensen, C. A. (1995). The impact of learning journals on metacognitive and cognitive processes and learning performance. Learning and instruction, 5(2), 167-185.

McDermott, R., Brindley, G., & Eccleston, G. (2010, June). Developing tools to encourage reflectionin first year students blogs. In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (pp. 147-151).

McDrury, J., & Alterio, M. (2002). Learning through storytelling: Using reflection and experience in higher education contexts. Dunmore Press Limited.15

Mitrovic, A., & Martin, B. (2002, May). Evaluating the effects of open student models on learning. In International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems(pp. 296-305). Springer, Berlin, Heidelberg.

Mitrovic, A., & Martin, B. (2007). Evaluating the effect of open student models on self-assessment. International Journal of Artificial Intelligence in Education, 17(2), 121-144.

Montalvo, O., Baker, R., Sao Pedro, M., Nakama, A., & Gobert, J. (2010) Identifying Students' Inquiry Planning Using Machine Learning. Educational Data Mining Conference, Pittsburgh, PA

M. Alzaid, and I. H. Hsiao, "Personalized Self-Assessing Quizzes in Programming Courses" Proceedings of Workshop on Educational Data Mining in Computer Science Education (CSEDM) in conjunction with The 11th International Conference on Educational Data Mining, 2018.

Moon, J. A. (2006). Learning journals: A handbook for reflective practice and professional development. Routledge.

Paliadelis, P., & Wood, P. (2016). Learning from clinical placement experience: Analysing nursing students' final reflections in a digital storytelling activity. Nurse Education in Practice, 20, 39-44.

Papancea, A., Spacco, J., & Hovemeyer, D. (2013, August). An open platform for managing short programming exercises. In Proceedings of the ninth annual international ACM conference on International computing education research(pp. 47-52). ACM.

Pee, B., Woodman, T., Fry, H., & Davenport, E. S. (2002). Appraising and assessing reflection in students' writing on a structured worksheet. Medical education, 36(6), 575-585.

Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012, February). Modeling how students learn to program. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 153-160). ACM.

Pintrich, P. R. (2004). A conceptual framework for assessing motivation and self-regulated learning in college students. Educational psychology review, 16(4), 385-407.

Quinton, S., & Smallbone, T. (2010). Feeding forward: using feedback to promote student reflection and learning–a teaching model. Innovations in Education and Teaching International, 47(1), 125-135.

Räisänen, M., Postareff, L., Mattsson, M., & Lindblom-Ylänne, S. (2020). Study-related exhaustion:First-year students' use of self-regulation of learning and peer learning and perceived value of peer support. Active Learning in Higher Education, 21(3), 173-188.

Rivers, K., Harpstead, E., & Koedinger, K. R. (2016, September). Learning curve analysis for programming: Which concepts do students struggle with?. In ICER (pp. 143-151).

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teahing programming: A review and discussion. Computer science education, 13(2), 137-172.

Roediger, H.L. and Butler, A.C., 2011. The critical role of retrieval practice in long-term retention. Trends in cognitive sciences, 15(1), pp.20-27.

Roediger III, H.L. and Karpicke, J.D., 2006. Test-enhanced learning: Taking memory tests improves long-term retention. Psychological science, 17(3), pp.249-255.

Rohrer, D. (2015). Student instruction should be distributed over long time periods. Educational Psychology Review, 27(4), 635-643.

Roscoe, R. D., & Chi, M. T. (2007). Understanding tutor learning: Knowledge-building and knowledge telling in peer tutors' explanations and questions. Review of educational research, 77(4), 534-574.

Sharples, M. (2013). Shared orchestration within and beyond the classroom. Computers & Education, 69, 504-506.

Shashkov, A., Gold, R., Hemberg, E., Kong, B., Bell, A., & O'Reilly, U. M. (2021, June). Analyzing Student Reflection Sentiments and Problem-Solving Procedures in MOOCs. In Proceedings of theEighth ACM Conference on Learning@ Scale (pp. 247-250).

Shute, V. J. (2008). Focus on formative feedback. Review of educational research, 78(1), 153-189.

Spacco, J., Denny, P., Richards, B., Babcock, D., Hovemeyer, D., Moscola, J., & Duvall, R. C. (2015, March). Analyzing Student Work Patterns Using Programming Exercise Data. In SIGCSE (pp. 18-23).

Stone, J. A. (2012, February). Using reflective blogs for pedagogical feedback in CS1. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 259-264).

Tang, C. (2002, July). Reflective diaries as a means of facilitating and assessing reflection. In Quality conversations: Proceedings of the 29th HERDSA Annual Conference Perth (pp. 7-10).

Topping, K. J. (2005). Trends in peer learning. Educational psychology, 25(6), 631-645.

Turns, J. A., Sattler, B., Yasuhara, K., Borgford-Parnell, J. L., & Atman, C. J. (2014, June). Integrating reflection into engineering education. In 2014 ASEE Annual Conference & Exposition (pp. 24-776).

Ullmann, T. D. (2015). Automated detection of reflection in texts: A machine learning based approach. Open University (United Kingdom).

Van Gog, T. and Sweller, J., 2015. Not new, but nearly forgotten: the testing effect decreases or even disappears as the complexity of learning materials increases. Educational Psychology Review, 27(2), pp.247-264.

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. International Journal of Artificial Intelligence in Education (IJAIED), 10, 151-182.

Winne, P. H. (1997). Experimenting to bootstrap self-regulated learning. J. Educ. Psychol. 89, 397–410.

Yost, D. S. (2006). Reflection and self-efficacy: Enhancing the retention of qualified teachers from a teacher education perspective. Teacher Education Quarterly, 33(4), 59-76.

Zimmerman, B. J., & Schunk, D. H. (Eds.). (2012). Self-regulated learning and academic achievement: Theory, research, and practice. Springer Science & Business Media.