The Effects of Bite-size Distributed Practices for Programming Novices

Mohammed Alzaid, Devanshi Trivedi, I-Han Hsiao
School of computing, Informatics & Decision System Engineering
Arizona State University
Tempe, AZ, USA
Malzaid1, Datrived, Sharon.Hsiao@asu.edu

***Abstract* — Programming novices usually find acquiring the ability to write programs challenging at first, however, they overcome this obstacle as they encounter more opportunities in the learning process. Providing learners with distributed practices and the ability to self-assess their programming knowledge is key to measure their development and guide them towards programming proficiency. In this work, we introduced QuizIT, a programming learning tool designed for novices. We conducted a classroom study and collected a semester long data to measure the effectives of the tool to achieve the design objectives. We analyzed the study data and provided the preliminary results from statistical perspective, as well as evaluating the effectiveness of the tool from learners' outcome. The data showed the positive effect of learners' usage of the tool on their course performance. We reported correlations exists in the data between effort (by actively benefiting and reflecting to the small learning opportunities) and the course outcome.**

Keywords—QuizIT, Programming learning, self-assessment.

I. INTRODUCTION

Learning programming languages and the ability to write programs have 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 instructor to orchestra the classes with the same instructional methods. The manual distribution of practices and grading process may not be scalable and may prevent learners from receiving enough learning opportunities and feedback. Not only that, if anything prevents the student from attending the class, subsequently they miss an opportunity to learn the latest content, assuming it was presented during that class time. Therefore, we now have the technology and capability (i.e. MOOCs) to provide alternative opportunities to learners that is accessible, distributed, graded and open for learner anywhere, any time.

We argue that having a daily quiz in the course will make learners be aware of their performance and provides a consistent opportunity to assess their mastery of the course units as discussed by (Braun, & Sellers, 2012). These small bites can be distributed to the learner as supplementary learning opportunities over the web, where it can instantly be graded and reported to student and teacher for more convenient self-assessment option. Also, moving the learners to online resources, ease the instructor task and open possibilities. The instructor could now benefit from monitoring student progress. Since classes differ from one to another, an interactive instructor may adjust the flow of the course as it proceeds. For example, when the instructor observe that most learners achieve mastery level of a given subject from daily quizzes, he could accordingly adjust the planned number or complexity of future questions on that subject.

Many researchers considered the power of small chunk of programming practices (Rohrer, 2015) and the strength from immediate reflects (Butler, & Winne. 1995) to enhance the learner opportunity in mastering the targeted objective. We were motivated by the positive effect these principles had and hypothesized if we built a framework combining selected learning principles in the domain of programming, it will enhance programming learning process. Our design rationales were based on the following learning science principles, such as 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). To surpass the challenge of learning programming, self-assessing is also considered as an essential aspect of the learning process which we utilized to ease programming for learners. Here, we present QuizIT, a new homegrown educational technology, it's a tool designed to monitor programming learners progress through quizzes. Both the learner and the instructor are consistently monitoring the progress of the course. The learner can compare his performance among the class for each learning opportunity provided, while instructor subsequently view the aggregated unit success ratio for the whole class for each subject. To evaluate the tool, we conducted a classroom study on how students use the tool and collected a semester long data. The daily learning opportunities followed the course organization

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

varying in topics and complexity. Students were encouraged to use QuizIT as a supplemental non-mandatory tool for the course. We logged the answer attempts, reviews, among other activities and made a comprehensive analysis from the data. By analyzing the attempt sequences, we found that actively using the tool, may lead to increase learner's performance in the course. The results of the study showed an indication of a positive correlation exists between students reflects and overall performance they made throughout the course, similarly, between number of learning opportunity encountered and the overall success ratio. We monitored the transition in course topics and observed that students struggled with decision and control topics while perform the best for methods, which was introduced later in the course. 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 require investigating the reflects triggers to explain the results. The small chunk of programming practices and the ability for immediate reflects had a positive effect in overall course performance.

The main objectives and contributions in this work are:

- Investigate the effectiveness of the homegrown QuizIT tool with bite-size multiple choice questions (MCQ) per day.
- Design and conduct a classroom study to collect learning and usage data from QuizIT.
- Analyze QuizIT usage logs for students' effort and performances.

The rest of the paper is organized as follows. Section II presents related work. Section III focuses on QuizIT and design rationale of the tool. In Section IV we present the class study and the data collection. Section V covers the experimental study and evaluation results and section VI concludes with future work.

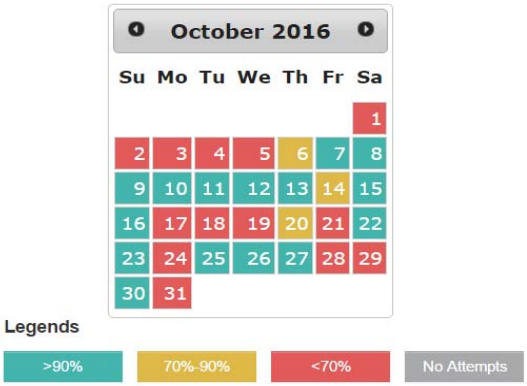

Figure 1. High active Student's calendar with attempt history. The colors show the performance for each daily quiz compared to the class.

II. Related work

In this section, we review some of the published work related to this research objective.

A. Programming (Self)-assessments in Learning

To examine quizIT self-assessment aspect in the programming domain, we introduced the tool to programming novices learning Java during a semester and analyzed their progress. (Spacco, Jaime, et al., 2015) conducted a similar study to analyze student work patterns using programming exercise data. The authors used data from a programming questions system, namely CloudCoder (Hovemeyer, & Spacco, 2013), to do preliminary analysis using linear regression on the data collected from five introductory programming courses in three universities. They were not able to identify whether the system help students learn the material, however they reported in the findings that there was a clear sign of students' improvement in programming skill as the semester progressed. They also found from the system usage a weak correlation between the students' effort and their final grade. Another finding was related to question difficulties. Even though challenging questions might require more time and effort, it may not lead to generating more errors than simpler questions. (Rivers, Harpstead, & Koedinger, 2016) considered the code produced by students' dataset from CloudCoder and applied learning curve analysis on it. For that purpose, they modified the traditional method of knowledge component KC modeling, to make it usable for code-writing data and generate KC. They also computed learning curves using real data and made categorizations of different syntax-based programming concepts. The subject that they noted student struggle with the most were in the no-learning category, which included but not limited to, math operations and strings. However, unlike their work, we are studying the aggregated topics as the students proceed in the courses.

WEB-CAT (Edwards, & Perez-Quinones, 2008) and ASSYST (Jackson, and Usher, 1997) are two examples of the automated assessment tools, which use pattern matching techniques to verify students' answers by comparing them with the correct answers. QuizJET (Hsiao, Sosnovsky, & Brusilovsky, 2010) is another example program that utilizes parameterized exercises to create a sizeable collection of questions to facilitate automatic programming evaluation; PeerWise (Denny, et al. 2010) utilizes student cohorts to leverage mass production of MCQs.

B. Feedback & Metacognition in Learning

The effect of distributed practices was studied by (Rohrer, 2015) and he showed that learner perform better when the learning opportunity is distributed over longer periods. In the study they conducted, the results showed that having longer and more distribution of learning material reflected with high test scores. (Roediger, & Butler. 2011) reviewed the effect of testing on learners and concluded to the importance of feedback to gain of learner from testing. They also argued that having more retrieval practices lead to an increase in retention

Figure 2. Answer attempt showing peer reflects. On the left, the quiz was successfully answered by the student in the first attempt, The right side show students' reflects for the same question.

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 may not be substantial because of focusing on delivering knowledge, both peers will gain from the that interaction. However, when this interaction is not formal, which can take place by other means, that negative effect of focusing on delivering knowledge was minimal. (Butler, & Winne, 1995) discussed self-regulators in which they identified the significant value of feedback. They reported that Self-regulators will use the feedback to evaluate their learning objectives and the estimate the learning outcome.

III. METHODOLOGY

QuizIT is a learning science principle grounded tool, provides the students with bite size distributed opportunities to master their programming knowledge. Not only it distributes MCQ per day, it also aims to solicit students' self-explanation as the learning reflection after answering a question. Additionally, it supports peer learning opportunities, including reviewing peers' comments, reflections and annotations on the quizzes. We designed the tool with these principles in mind, 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). The set of questions were provided as a bite size chunks, to enable learners steadily building on what they already learned in the process to gain the knowledge of programming. The questions vary from conceptional knowledge question to debugging and execution questions. The daily quiz gives an opportunity for the learners to reflect on what they learned recently in the class. It also increases the exposure to the programming concepts and provides a consistent option to self-assess. To explore QuizIT's learning effects, specifically the associations of learning and temporal distribution of practices, the following sections describes the system design rationale.

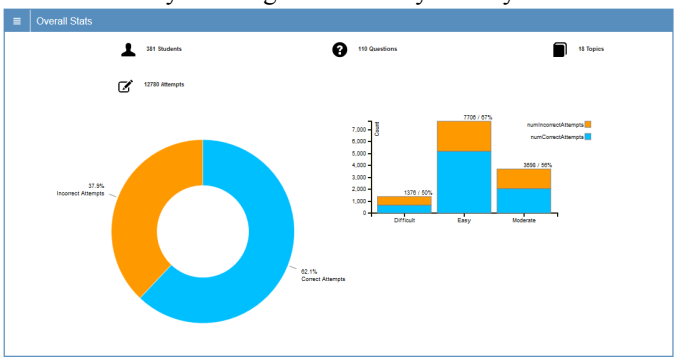

Figure 3A. Instructor analytics for students attempts. Top part shows number of students, attempts and questions. The success ratio is shown to the left and complexity ratio to the right.

A. QuizIT Design Rationale.

QuizIT's implementation has two main end points, the instructor side and the student side. Although the design objective of the tool was to help students, the instructor side provides an insight to class performance This can lead the instructor to adjust the course flow to recover weak performing topics. Thus, helping students optimally by connecting them to the instructor. The technologies used in the implementation of quizIT were: Django, Python, Javascript, JSON, jQuery, Bootstrap and D3.js. These technologies were chosen to provide flexibility both in development and management of the data with Django admin interface.

1) Student Interface

QuizIT student interface allows students to actively participate in the course activities while providing access to the daily quizzes. Through this interface, Students can answer the daily quiz, review and retry the past quiz attempts and participate in the class activities while posting the comment in the discussion section. One of QuizIT initiatives is positive encouragement

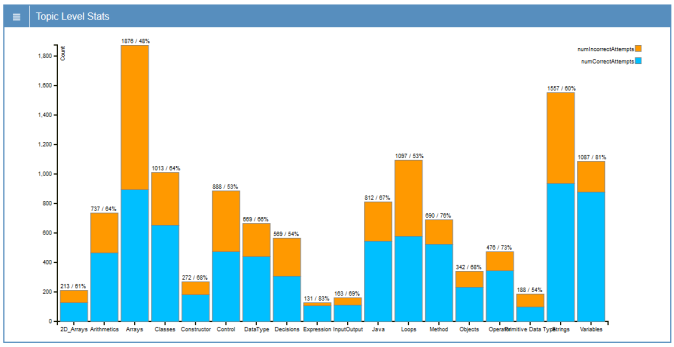

Figure 3B. Instructor view of topic progress showing success ratio for each topic

for students to reflect. The system is developed on the consideration of the objective of positive encouragement by creating pressure to perform. To achieve this, QuizIT implements unique color scheme for question performance instead of overall performance. The Red color represents below average, Orange for average and green for high performance as can be seen in fig. 1. Here we see the progress of an active learner during a month. The color scheme is visible on the calendar navigation. The calendar navigation provides easy form of navigation with quick review of question details which can be further fully reviewed or retried.

From this calendar, the student can revisit the quizzes they didn't perform well in the first attempt. Another way to review the progress history is through the question history section, which we call the list view. It provides quick view of past question with the status of correctness, date, topic of the question with possibility to reattempt in tabular sortable form.

2) Instructor Interface

QuizIT provides flexible interface for the instructor including creating multiple courses, self-explanatory dashboard visualization, Setting and editing question, viewing question status and flexible Calendar navigation for quick access. The QuizIT instructor dashboard visualization provides a clear view of the course performance using minimal color scheme and simple form of graphs. In each graph, the orange color describes the negative aspect whereas the blue color represents the successful aspect of the described information as seen in fig. 3a and fig. 3b. The dashboard gives the analysis of the course in three main section. The main graphs in this section are the overall analysis and topic progress status. The overall analysis graph visualizes the course progress with detailed information about student enrolled in that course, number of questions, number of topics and 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 correct vs incorrect attempts whereas the bar chart compares the success of the class performance with respect to the question complexity. Fig. 3a shows the course interactive visualization. The topic progress Status seen in fig. 3b compares the topic wise success in the form of bar graph. The clear representation and the side-by-side comparison gives immediate idea about students' performance over course subjects. This can immediately indicate which topics requires providing more learning opportunities. Topic level Status - Table: This tabular view has more detailed representation of the data including topic, number of questions, total attempts, number of correct attempts and success percentage. The number of students participating in any learning opportunity is also provided.

B. Overview of QuizIT concept

Each day, a new programming MCQ is posted by an expert instructor to the course profile, the students will be directed to the daily learning opportunity as they log into the system. once an attempt is made. Questions are easily accessible to review and retry using calendar or list view, were in both options, we show the peers performance for every question as well. The opportunity to reflect is provided with the ability to review and rate peers reflects during answering attempt. Each attempt to answer a given question is marked with the appropriate flag indicating the review category (review, attempt & retry) and whether the question was correctly answered or not. When a question is shown to the user, immediate reflection is encouraged throughout the attempt process. Once a reflect is made, the student can review peers' comments and given the opportunity to retry the question independently of the outcome of the first attempt. In fig.2 we see a snapshot of a successful first attempt were he also immediately reflected in the comment section. Learner can only review peers' comments if they submit their own reflect to encourage them to reflect more often. We adapted a coloring scheme that indicate the user's performance with overall class performance as we described earlier. This is used to raise learner's awareness of their performance on any learning opportunity and encourage them to revisit questions they incorrectly answered on the first attempt. Using the list view to show attempt history, they can clearly see their progress and performance over time and topic. Next, we present the study and key elements in data collection.

IV. EXPERIMENTS

To capture novices' learning data while working with bite-size quizzes, we conducted a classroom study. QuizIT was introduced to an introduction to Java programming course at Arizona State University of Fall 2016 semester. Students were encouraged to use the system as non-mandatory tool. Among 375 registered students, 176 actively used the system and generated 11484 attempts. A total of 110 unique quizzes throughout the semester. These questions covered a total of 18 topics which include (Java, Primitive Data Type, Method, Datatype, Expression, Variables, Strings, Arithmetic's, Operator, Objects, Control, Decisions, Loops, Classes, Constructor, Arrays, 2D Arrays, InputOutput). There were three level of complexity varying from easy, moderate to more challenging questions and cover conceptual knowledge and programming skill include code and non-code questions. The questions followed the course organization, were quizzes on new topics are introduced once it has been initially covered in the class. We collected data until the final exam date. We used the final class grades to compare the students' performance from using QuizIT and the class performance. We obtained the class grades after the final exams. The class performance was evaluated using homeworks, labs, quizzes and exams.

A. Data Collection

When users access learning opportunities, we mark the time stamp and session id for the event based on the action that immediately follows. If the user makes an answer attempt, the chosen question option and correctness of the answer is subsequently registered. When no answer is provided, we mark that as a review event. We focused on three aspects of the data during our analysis, attempts, reflects and sessions.

Feature		Observed value			
	Max	*Min*	*Mean*	*Median*	*SD*
#Question attempted	***98***	***11***	***34.28***	***35***	***20.6***
# Day accessed	***70***	***2***	***9.2***	***5.5***	***7.1***
Total attempts	***220***	***16***	***80.39***	***61.5***	***19.8***
Correct attempts	***101***	***10***	***44.16***	***36***	***18.4***
Incorrect attempts	***108***	***7***	***36.85***	***24.5***	***22.5***
Reflect size chars	***474***	***2***	***60.20***	***42***	***24.5***
# Reflects	***47***	***1***	***6.8***	***1***	***11.2***
Session actions	***225***	***23***	***86.5***	***66***	***43.9***
# Reviews	***299***	***22***	***69.9***	***53.3***	***33.8***

Table 1. List of features and associated data used in analysis.

1) Answer Attempts

One of the main indicators whether a student learned a given subject is the first attempt success ratio. Throughout our analysis, we used this as an indicator of learner mastery of a given subject. Users can see their attempts for a question and the overall performance for each question compared to the class marked with the appropriate color. If they had more than one attempt, we mark the incorrect answer and they can go for another attempt. If the question was incorrectly answered initially, we don't show how they did on the first attempt to enable them to reflect through retries. The coloring of the attempt is based on the ratio of correct attempts over all attempts for any question. Here, the overall ratio includes re-attempts as well. Moreover, we consider the second attempt using retry feature as another form for reflects. We define Second attempts as attempts that occurs on a different session after the first attempt using the retry functionality. From the instructor perspective, we would only show the overall success rate for the topics in the course and the detailed data for each question.

2) Reflects and comments

QuizIT design focuses on self-assessing. To increase the usefulness of this feature, students were encouraged to reflect as they encounter a new learning opportunity. Reflects are generated by the users as comments after an answer attempt or retries on a different session. We passively monitored the progress to see how students are participating in this procedure. We considered the preceding event before the comment as a trigger for the reflect. For each reflect posted, we associated it with question topic and complexity. We also had an expert labeling comment as constructive and non-constructive based on its relevant to the question and filtered out those that adds no benefit to peers. Although we have not focus on the content of the reflects in this work, we plan to use the data from this feature in the next study. The main reflect features that we analyzed in the study are total number of reflects and the reflect average length.

3) Sessions

The daily quiz gives an opportunity for the learners to test, reflect and assess on what they learned recently in the class. Although the tool and the experiment were designed to provide a question per day, we anticipated the participant to answer multiple questions during one session. Since novices programming courses tend to span on more than one day per week, they may not have the time to allocate for the course each day considering their schedule. In this experiment, we also wanted to keep the learning opportunity accessible even after the posting date and allowed for flexible attempts. As we will discuss in the study results, this indicator showed a behavioral aspect in the system usage with a spike in question attempts during two specific weeks.

B. Descriptive Data

During the analysis, we identified a set of features that we predicted to have significant correlation with student performance. Table 1 shows the detailed data used in the analysis from the dataset in the study. The values are calculated for active users showing minimum and maximum values along with corresponding mean and median and standard deviation to give an understanding of the data distribution. Here, we considered mainly the data that contains active users, were we set the minimum threshold for number of questions and days as seen in the table, which we will discuss next in the result section. Some of the features are self-explanatory, but for better understanding, we will briefly describe the feature list. The total number of attempts represent all the attempts that been performed by the learner for all learning opportunities. This include initial attempts and all consecutive reattempts along with retries. The retries defer from initial attempts by providing the same learning opportunity with shuffled options. Correct and incorrect attempts mark the sum of initial attempts and retries. We can see the maximum number of correct and incorrect is higher than unique question attempts because of the inclusion of retry function. For reflects, the data here represent only active users who reflected at least once. Session actions represent all the activities during a given session including answering reviewing and retries. Lastly, number of reviews represents how many times the user accessed the learning opportunities. Once the question is attempted, it can be accessed again in two ways, simply review or attempt a retry. The choice was left to the user to decide. We saw that students preferred to use review much more than using retry to access the questions again.

V. RESULTS

To measure the effects of distributed practices and reflection on student performance during learning, we performed the quantitative analyses to verify the hypotheses on the following: (Tool usage, effect of effort, effect of reflect, first encounter, question complexity and topics). We examined the data starting from the general system usage and reported the highlights.

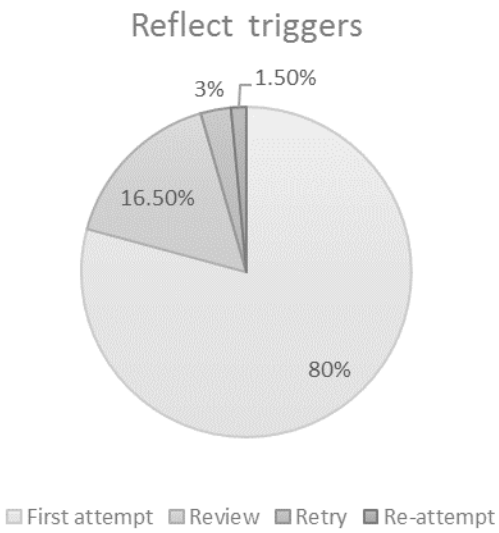

Figure 4. Reflect triggers as seen in the experiment.

Additionally, the efforts and reflections effects on the overall course performance are measured. Moreover, we analyzed the effects from learning opportunities provided, including the first attempt success, question complexity and topical influences.

A. QuizIT usage

To visualize the system usage over time, we plotted all the quizzes attempts spinning across the entire semester fig. 6. Since we allowed access to the questions after they have been posted, from the system usage, we observed higher volume of activity during the week before each exam. The spikes indicated the use of QuizIT to prepare for the upcoming exam. Next, we discuss how students used QuizIT features.

According to QuizIT design, students can access question in three ways, quiz of the day page, list view and calendar view. Students will be directed to the quiz of the day page as they log into the system, which generated 30.6% of all reviews and attempts. This means that over 30% of attempts were on the same day of the quiz. However, since we enabled access to previously posted questions, part of those attempts was the start of a session that included few questions. Therefore, we couldn't see enough evidence of the effect of daily quiz. Over 53.5% of traffic preferred to use list view over calendar view for the remaining 15.5%. 60% of active students mainly used list view to access questions while 27% had mainly used calendar view. The remaining used mixture of both. Although questions with related topics were distributed throughout the semester, we found the patterns of consecutive attempts for related topics, which indicate that students were using the search function provided in the tool to access questions with similar topic. From the number of sessions and attempts, we saw preferences of students to answer multiple questions in the same session. These results provide us with ideas for future enhancements in the system. To understand the impacts of QuizIT on learning, we continue investigating in the amount of work that students did on the system.

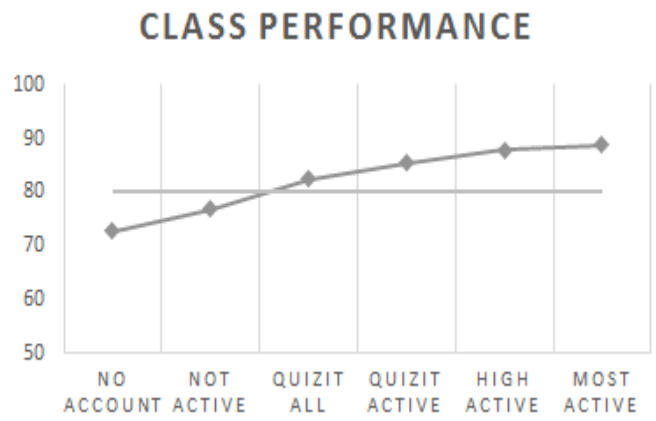

Figure 5. Class performance for quizIT users' categories.

B. Effect of effort

Considering students' effort put into these bite-size distributed quizzes throughout the semester, we used their final scores to associate to their QuizIT usage. The class students differ of how they interacted with the tool. Part of the class did not use it, we labeled them as No-Account group; some registered but did not use it actively was labeled as Non-Active; QuizIT All represented all students who register to the system; The active users who passed the 10 questions attempted threshold are marked as QuizIT Active; High Active represent learners who encounter higher than the mean question attempted (34); lastly, Most active represent active students who attempted more than half of available learning opportunity provided. In fig. 5 we compare the class outcome of all class students including quizIT and non QuizIT users. We found that QuizIT had positive impacts on overall course performance according to their activities. We can see in the chart that No-Account group performed the least among other groups; following by the not active group, the class performance then steadily increases as learners encounter more learning opportunities. We saw that, there was three letter grades different between QuizIT register users average and students who opt out of using the system. We also consider the split by the number of sessions they had but it showed no major difference, so we are not showing the result here.

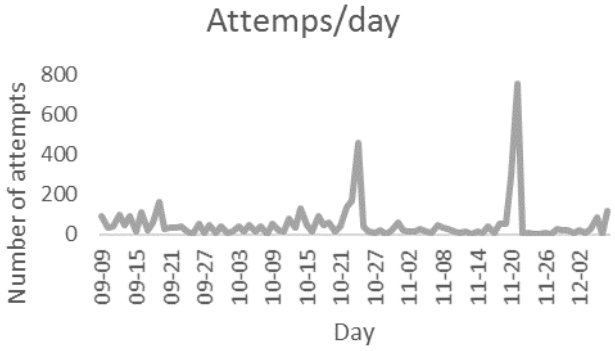

Figure 6. system usage for initial opportunities encounter.

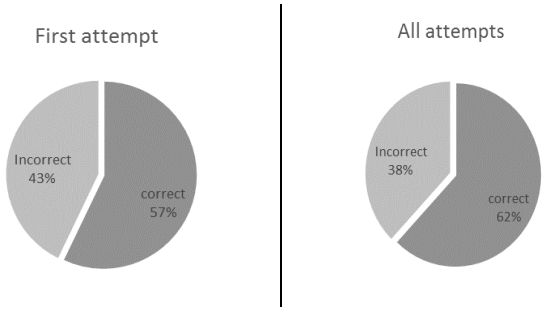

Figure 7. First attempt and overall attempts.

We can see that quizIT learners had above average class performance (80.2% line), with a steady increase in the class outcome, as the learners become highly active and participate in more learning opportunities. The most active group had an equal split over the mean of number of days. This group yield the best performance in the class with around 88% average final grade. With further analysis, we also found a positive correlation exists between total number of attempts and the overall success ratio (P value < 0.003, R > 0.23). We wanted to explore further if active users differ over complexity as well. In Fig. 8 the learners are classified into two groups based on their active usage. We considered the on the mean number of learning opportunities they took as seen in table 1 (#question attempted mean 34). Active students with lower than average was labeled as low active, while highly active were above that. We can see in the figure that both groups had similar success ratio for easy learning opportunities, but clearly split as they encounter challenging questions.

When examine the time stamp of the answer attempts, we saw that low active group had higher tendency to rapidly providing random answer attempts to find the correct answer. It's worth reminding here that in all learning opportunities given to the learners, they always end up submitting the correct answer which indicate that all quizIT users wanted to learn from the learning opportunity by knowing the correct answer.

C. Effect of reflects

As we mentioned before, the students were encouraged by the instructor to reflect on learning opportunities. We wanted to understand what triggers the reflect throughout learning process in the tool and measure their effects. We monitored when the student takes the time to reflect after encountering the learning opportunity. In fig. 4, we see that an overwhelming 80% of reflects were generated immediately after student first attempt, while the remaining 20% were during review and retry activities. To filter these reflects, we classified the textual reflects into constructive and non-constructive comments, were constructive comments relate directly to the learning opportunities while non-constructive were anything but that. 18% of reflects were labeled as unconstructive comments. The positive correlation exists between students reflects by posting comments after first attempts and overall performance they made throughout the course (P value < 0.02, R > 0.24), similarly, between total number of attempts and the overall success ratio. We observed that reflects posted by 22% of active users. 25% of reflects generated on the post day of the question. 57% of reflects were posted in a session along with another question reflect. We saw that learners tend to reflect more when they take more than two learning opportunity in the same session. When examining the consecutive attempts after reflects and found that 58% of students who reflected on incorrect attempts improved and correctly answered on the next opportunity., When we examined the distribution of reflects by topic. Here loops and Strings were the highest to capture students reflects.

D. First encounter

Another measure for student performance was their first attempts at each learning opportunity. Even though learners will eventually find the correct answer as we observed in the data, the first attempt had significant importance to mark their initial understanding of the question and the related topic. Throughout the dataset, when the first attempt was marked as incorrect, it was usually followed by consecutively guesses to find the correct answer. We also followed the first attempt success ratio in comparison to overall course success ratio, which includes retry, as shown in fig. 7. Here we compare the aggregated success rate for the first attempts comparisons to overall course. First attempts were close to 57%, while the overall course success ratio was 62%. The success ratio for first attempt on same day of posting were slightly higher than aggregated first attempt ratio (58%).

E. Question Complexity

Among the three level of complexity, 70 were easy, 31 moderate and 9 tough questions. The distribution of the question complexity was not equal over all topics which can be seen in fig. 9A. As expected, we saw the success ratio decrease as the question become more challenging. Moderate complexity opportunities were interestingly higher to capture learner reflections. We were not able to justify that, however, after examining the actual reflects, we expect it may be caused by the fact that moderate questions require more concentration and engage the learner more. However, that requires further investigation in reflects triggers to explain such results.

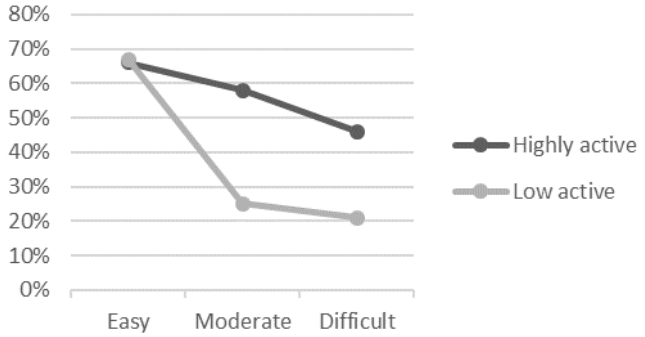

Figure 8. Question complexity success ratio between high and low active users.

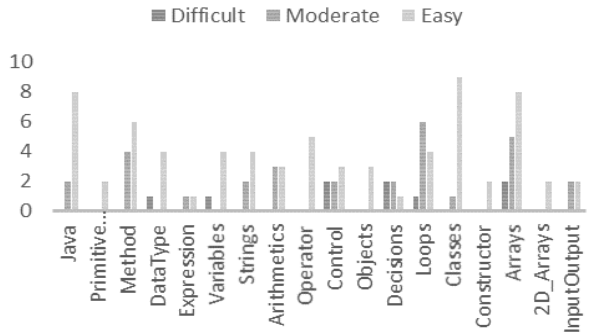

Figure 9A. Distribution of learning opportunities levels complexity level over topics.

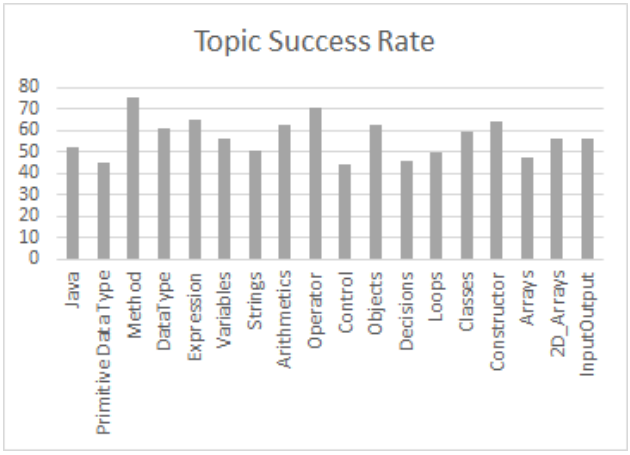

Figure 9B. Success ratio per topic from first attempt.

F. Topic learning.

From fig. 9B, we saw that students struggled with primitive data types, decision and control topics while perform the highest for methods, which was introduced later in the course. To explain that we considered the distribution of the learning opportunities throughout the course and we noticed that both decision and control topics had the highest number of tougher questions as can be seen in fig. 9A. For the primitive data type case, we found that it had the least learning opportunities among all topics. Since it was introduced early in the course, the distribution of questions followed the design of the course syllabus, which may have prevented from giving the instructor enough window to provide more bites for the students to master that topic. However, this is simply an observational result from the data set and we don't argue that we identified the reason behind the difference of student's performance over course topics. We plan to examine that in more details in future study.

VI. CONCLUSION & DISCUSSION

In this paper, we presented QuizIT, a new homegrown educational technology designed to assist and monitor programming learners progress through quizzes. To evaluate the tool, we performed a classroom study and reported the preliminary analysis of the first study dataset. We found correlations between student effort and performance along with identifying active answering patterns associated with positive course outcome. We found some findings in the current study requires more data to be verified, for example we noticed posting reflects were higher for moderate level questions but we would like to know the reflects triggers that could explain such result. Although the use of the tool was not required, the students' participation in the study and the effect it had over their performance is promising. This encourage us to further enhancement of the tool to aid students into programming learning mastery. Limitation of the evaluation in this study generated from the use of data of one class for one programming language. further study is required to provide support for the initial findings. We also did not capture the use of some features such as sorting and reviewing peer's comments. We didn't evaluate the actual content of the learning opportunities which could be broken down to code vs non-code question. The temporal effect of the session and attempts was not presented in this work as well.

We plan to overcome all the limitations in this study and use the data from future studies to confirm the findings of our experiment. Currently we have new courses enrolled in QuizIT and their progress is being monitored. We are studying the effect of daily quiz with access time limit as well. In the meantime, we will introduce new system features for analytic visualization and reflect procedures. The future scope of this system includes the topic wise performance analytics in visual format, more easier navigation, topic wise comparison with overall class performance. Making the comment section compulsory for class involvement and reflection at the student side may as well increase the learners gain from the system. Also, adding interface to visualize question-wise and topic-wise performance of each student at the instructor side to each student will enable instructors to guide and assist effectively.

REFERENCES

[1] Bennedsen, Jens, and Michael E. Caspersen. "Failure rates in introductory programming." ACM SIGCSE Bulletin 39.2 (2007): 32-36.

[2] Braun, Karen Wilken, and R. Drew Sellers. "Using a "daily motivational quiz" to increase student preparation, attendance, and participation." Issues in Accounting Education 27.1 (2012): 267-279.

[3] Butler, Deborah L., and Philip H. Winne. "Feedback and self-regulated learning: A theoretical synthesis." Review of educational research 65.3 (1995): 245-281.

[4] Denny, Paul, et al. "PeerWise: students sharing their multiple choice questions." Proceedings of the fourth international workshop on computing education research. ACM, 2008.

[5] Edwards, Stephen H., and Manuel A. Perez-Quinones. "Web-CAT: automatically grading programming assignments." ACM SIGCSE Bulletin. Vol. 40. No. 3. ACM, 2008.

[6] Hacker, Douglas J., John Dunlosky, and Arthur C. Graesser, eds. Metacognition in educational theory and practice. Routledge, 1998.

[7] Hovemeyer, David, and Jaime Spacco. "CloudCoder: a web-based programming exercise system." Journal of Computing Sciences in Colleges 28.3 (2013): 30-30.

[8] Hsiao, I-H., Sergey Sosnovsky, and Peter Brusilovsky. "Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming." Journal of Computer Assisted Learning 26.4 (2010): 270-283

[9] Jackson, David, and Michelle Usher. "Grading student programs using ASSYST." ACM SIGCSE Bulletin. Vol. 29. No. 1. ACM, 1997.

[10] Rivers, Kelly, Erik Harpstead, and Ken Koedinger. "Learning Curve Analysis for Programming: Which Concepts do Students Struggle With?." Proceedings of the 2016 ACM Conference on International Computing Education Research. ACM, 2016.

[11] Robins, Anthony, Janet Rountree, and Nathan Rountree. "Learning and teaching programming: A review and discussion." Computer science education 13.2 (2003): 137-172.

[12] Roediger, Henry L., and Andrew C. Butler. "The critical role of retrieval practice in long-term retention." Trends in cognitive sciences 15.1 (2011): 20-27.

[13] Rohrer, Doug. "Student instruction should be distributed over long time periods." Educational Psychology Review 27.4 (2015): 635-643.

[14] Roscoe, Rod D., and Michelene TH Chi. "Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions." Review of Educational Research 77.4 (2007): 534-574.

[15] Shute, Valerie J. "Focus on formative feedback." Review of educational research 78.1 (2008): 153-189.

[16] Spacco, Jaime, et al. "Analyzing student work patterns using programming exercise data." Proceedings of the 46th ACM Technical Symposium on Computer Science Education. ACM, 2015.

[17] Topping, Keith J. "Trends in peer learning." Educational psychology 25.6 (2005): 631-645.

[18] Zimmerman, Barry J., and Dale H. Schunk, eds. Self-regulated learning and academic achievement: Theory, research, and practice. Springer Science & Business Media, 2012.