

Students Satisfaction with Mastery Learning in an Introductory Programming Course

Luiz Augusto de Macêdo Morais¹, Jorge C. A. Figueiredo¹, Dalton D. S. Guerrero¹

¹Software Practices Laboratory
Federal University of Campina Grande
Aprígio Veloso Ave., SPLab, Bodocongó
58429-900 – Campina Grande - PB – Brazil

luizaugusto@copin.ufcg.edu.br

{abrantes, dalton}@computacao.ufcg.edu.br

Abstract. *This paper investigates the satisfaction of students with the application of Mastery Learning (ML) in CS1. We applied the method to allow students master the concepts and progress at their own pace. We performed a survey and discovered that subjects are very satisfied with ML and they prefer to use it instead of a traditional method.*

1. Introduction

Experienced programming teachers believe there are at least two populations in CS1 with distinct experiences, paces and learning strategies [Dehnadi and Bornat 2006]. Some students cannot keep pace with the course and fall behind. Others, with prior experience or who learn faster, have to wait for their classmates.

In traditional teaching methods, teachers try to keep a pace to benefit all the class. However, differences in students' learning and the tightly integrated nature of CS1 curriculum increases the gap between students along the course [Robins 2010].

We applied Mastery Learning (ML) in CS1 to provide a personalized instruction, allowing students to master the concepts and progress at their own pace. This study is the first step to investigate the effectiveness of the method in our course. We explored the satisfaction of students with ML and their opinion about its continuation and expansion.

We discovered that students approved the method (Section 4.1) – specially feedback provided by automatic tests – and agreed with its continuation and expansion to other programming course (Section 4.2).

2. Background

Mastery Learning is a philosophy about teaching and learning, which asserts that under appropriate instructional conditions all students can master the concepts being taught [Block and Burns 1976].

One of the most successful approaches of Mastery Learning is Personalized System of Instruction (PSI), created by Fred Keller [Keller 1968]. It allows students master the concepts at their own pace and increases communication between students and instructors [Fox 2004]. Its main characteristics are: (a) *unit mastery* — students only advance a

unit when they master the concepts; (b) *self-pacing* — students can progress at their own pace; (c) *tutoring* — tutors are responsible for guiding and assessing students; (d) *emphasis on written word* — textual materials are used for primary delivery of course content; and (e) *lectures for motivational purposes* — lectures are used for demonstrations and motivational purposes.

Another ML approach is Learning for Mastery (LFM), developed by Benjamin Bloom [Bloom 1968] to allow the majority of students effectively master the concepts. Its main characteristics are: the *feedback, corrective, and enrichment process* — students need to understand their weaknesses and correct them as well as comprehend their strengths and expand their learning; and *instructional alignment* — all stages and objectives of the teaching-learning process should be explicit and well-defined [Guskey 2005].

We implemented the method using characteristics from both PSI and LFM. In addition, we included computer-aided activities to adapt ML to the context of programming classes.

3. Methodology

We performed a survey to investigate if students are satisfied with the application of Mastery Learning in our course. Although surveys are usually used in large-scale studies, the contributions of this work are specific to the context of our institution.

3.1. Course design

The course consists of 10 units, separated in two groups: *basic units*, which contain fundamental concepts of programming and *advanced units*, which address subjects such as data structures and functions.

Students have to master a unit to progress to the next. It is done scoring at least 6.0 in two exams of the same unit, which are administered in different weeks. If students have prior experience or feel confident with the topics, they can take the proficiency test and finish the course faster.

The course has parallel classes, which means that students in different units study distinct lessons. The same classes are offered several times, so that students can reinforce weak topics. They are also assisted by tutors, which are responsible for assessing, diagnosing and correcting their difficulties.

Students can learn at their own pace thanks to online materials called *roadmaps*, which are designed to cover the concepts of each unit so that students can study or reinforce topics, without the help of instructors. They also have access to an online submission system that allows them to submit solutions and receive automatic feedback [Gaudencio et al. 2013].

3.2. Participants

The course comprised 92 Computer Science students at Federal University of Campina Grande, which attended classes during the second semester of 2013. Among the subjects, 39 passed the course, 26 failed it, 6 passed the proficiency test and 21 dropped out of the course. We discarded the last ones because they may have not experienced the method for enough time.

We selected students using quota sampling [Cohen et al. 2007], which is a non-probability sampling strategy that tries to maintain the same proportion of subjects in both population and sample.

Students were separated in two groups: (A) those who mastered the units and (B) the others who did not. Group A is composed of subjects with average grade of units greater than or equal to 6.0 and group B comprises students who scored less than 6.0. We used the number 6.0 to separate the groups because it is the boundary that defines if a student mastered or not a unit.

The population was composed of 41% students in A and 59% ones in B. The sample contained 54 students, separated in the same proportion. We carried out the Mann-Whitney's test and accepted the null hypothesis ($p\text{-value} = 0.91$), confirming the representativeness of the sample.

3.3. Data collection and analysis

We used a structured online questionnaire to collect students' data. It was administered before the release of final grades to avoid bias from students who failed the course. It was validated by two professors, which are experts in ML. They guaranteed face validity after a series of discussions about each question.

After data collection, we created contingency tables and classified the answers as follows: dissatisfied (scales 1-3), indifferent/neutral (scale 4) and satisfied (scales 5-7). We used Fisher's exact test to analyze whether results are independent of groups A and B (see section 3.2), which means that both groups have similar opinions.

4. Results

We discovered that students are very satisfied with Mastery Learning and most of them are favorable to its continuation and expansion to other programming course.

4.1. Satisfaction

We asked students, in a 7-point Likert scale, their level of satisfaction regarding the application of Mastery Learning. The majority of students (45/54, 83%) declared being satisfied with the method, whereas only 2/54 (4%) are not satisfied and 7/54 (13%) are indifferent. Figure 1 shows the scale of satisfaction, separated by (A) students who mastered the units and (B) those who did not.

We performed Fisher's exact test ($p\text{-value} = 0.49$) and confirmed that both groups are satisfied with the application of ML. However, Figure 1 reveals higher satisfaction among subjects who finished the course with a better performance (group A).

Students are also satisfied with some characteristics of ML, such as: feedback, tutoring, self-pacing, roadmaps and unit mastery. The detailed results are showed in Table 1.

According to Table 1, the majority of students are satisfied with all characteristics of Mastery Learning. We performed Fisher's exact test and confirmed ($p\text{-value} > 0.05$) that both groups A and B have similar levels of satisfaction.

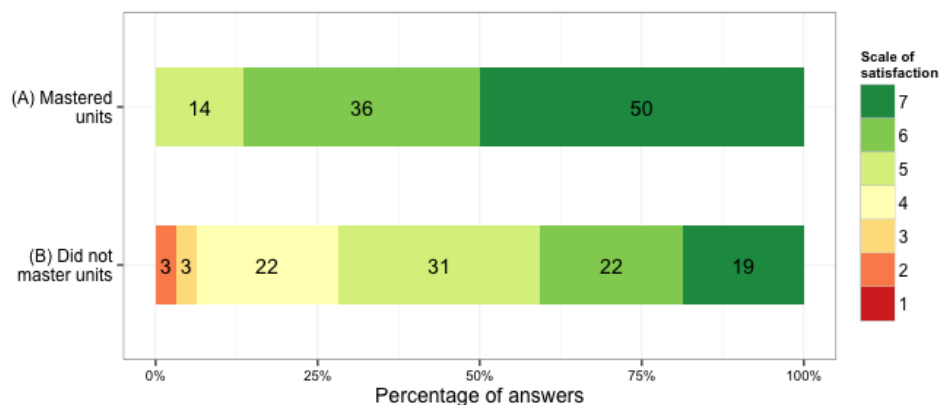


Figure 1. Satisfaction of students with Mastery Learning

	Feedback	Tutoring	Self-pacing	Roadmaps	Unit mastery
Satisfied	98.1	90.7	81.5	79.6	66.6
Indifferent	1.9	5.6	11.1	13.0	16.7
Dissatisfied	0.0	3.7	7.4	7.4	16.7

Table 1. Percentage of students' satisfaction with characteristics of ML

Feedback in exercises is the aspect with the higher level of satisfaction (*median* = 7) among students (98.1%). One interviewee said: "Feedback is important because it highlights my difficulties".

Although unit mastery was rejected by 16.7% subjects, it was accepted by 2/3 (66.6%) ones, with an average of 5.5 points in Likert scale. We discovered that some subjects who did not master the units are dissatisfied with this characteristic because they do not like of falling behind while their classmates advance to next units.

4.2. Continuation of ML

We analyzed multiple choice questions to investigate if students support the continuation of ML in the course and if they want to study other programming course based on the method.

Most students (94%) affirmed we should continue using Mastery Learning, but 68% of them think the method needs to be improved. On the other hand, only 3 (6%) students prefer a traditional method. Fisher's exact test ($p\text{-value} = 0.12$) confirmed that groups A and B have similar opinions.

We asked students if they would study with ML again. Forty-three (80%) subjects prefer ML, whereas 11 (20%) ones opted for studying in a traditional method. Fisher's exact test showed that students in group A tend to prefer studying with ML ($p\text{-value} = 0.03$).

We also asked students if they wanted to study the next programming course with ML. The majority (80%) were favorable and 20% prefer studying with a traditional method. Fisher's exact test rejected the null hypothesis ($p\text{-value} = 0.03$), so there is a relation between the answers and the performance of respondents.

5. Conclusion

This work investigated the satisfaction of students with the application of Mastery Learning in an introductory programming course. We discovered that students are satisfied with the method and they are favorable to its continuation and expansion.

The results of this paper contributed to justify the use of Mastery Learning as a pedagogical strategy for teaching programming. We believe that improving the satisfaction of students with the method is the first step to promote an effective learning.

We plan to replicate this study in further classes to confirm the results and to explore the advantages and disadvantages of applying Mastery Learning in a programming course.

References

- Block, J. H. and Burns, R. B. (1976). Mastery Learning. *Review of Research in Education*, 4(1):3–49.
- Bloom, B. S. (1968). Learning for Mastery. Technical report, UCLA.
- Cohen, L., Manion, L., and Morrison, K. (2007). *Research Methods in Education*. Routledge, New York, 6th edition.
- Dehnadi, S. and Bornat, R. (2006). The camel has two humps. Technical report, Middlesex University.
- Fox, E. J. (2004). The Personalized System of Instruction: A Flexible and Effective Approach to Mastery Learning. In Moran, D. J. and Malott, R. W., editors, *Evidence-based Educational Methods*, chapter 12, pages 201–221. Elsevier, San Diego.
- Gaudencio, M., Wanderley, L. F., Lemos, F. W., de Araújo, E. C., Figueiredo, J. C., and Guerrero, D. D. (2013). Eu sei o que vocês fizeram (agora e) na aula passada: o tstview no acompanhamento de exercícios de programação. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 24.
- Guskey, T. R. (2005). Formative Classroom Assessment and Benjamin S. Bloom: Theory, Research, and Implications. In *American Educational Research Association*, number April, pages 1–11, Montreal.
- Keller, F. S. (1968). Good-Bye Teacher... *Journal of Applied Behavior Analysis*, 1(1):79–89.
- Robins, A. (2010). Learning edge momentum: a new account of outcomes in CS1. *Computer Science Education*, 20(1):37–71.