

Making a Case for Studying the Effects of Using GitHub on HBCU Students' Computational Thinking Skills

1st Dr. Andrea E. Johnson
Computer and Information Sciences
Spelman College
Atlanta, USA
ajohn308@spelman.edu

Abstract—While preparing students for careers beyond graduation, HBCU faculty often have to balance the priorities of bridging the gap between the level of student preparation and the expected level of preparedness as influenced by industry and discipline standards. Prioritization of these standards does not always align with the experience of students at Historically Black College and Universities (HBCU) and Minority Serving Institutions (MSI). The Association of Computing Machinery (ACM) 2013 Computer Science Curricula guidelines highlight version control as one of many core tools in the Social Issues and Professional Practice (SP) Knowledge Area students should experience in their upper-level software engineering courses. Research has demonstrated that at this level, students receive exposure to the tool but are not able to gain sufficient mastery. Other researchers have advocated for version control being introduced earlier in the computer science curricula. There is limited research that indicates that this particular tool improves the learning outcomes for students or significantly complements the other ACM Knowledge Areas. This paper explores the challenges experienced by an HBCU faculty member while integrating GitHub into an advanced programming class. It is important to note, there is no research that directly addresses the use of version control in predominately African American classrooms.

Index Terms—programming education; computing education; version control systems; GitHub

I. INTRODUCTION

HBCU students typically start introductory computing courses less prepared to handle the same rigor as their peers at PWI serving institutions and may not benefit in the same way from the introduction of version control. Faculty should prioritize improving students' computational thinking skills and thereby improve their ability to develop algorithms, decompose tasks, perform problem abstraction and program incrementally. Currently, the vast majority of introductory [Computer Science] courses are programming focused, in which students learn about concepts in computer science (e.g., abstraction, decomposition, etc.) through the explicit tasks of learning a given programming language and building software artifacts. [1]

The CS2013 guidelines, encourage the development of soft skills (such as teamwork, verbal and written communication,

time management, problem solving, and flexibility) and personal attributes (such as risk tolerance, collegiality, patience, work ethic, identification of opportunity, sense of social responsibility, and appreciation for diversity) as necessary for the workplace. [2] The CS2013 guidelines also list the Professional Practice Knowledge Areas that make up the profile of a Computer Science graduate. One particular skill, version control, is highlighted as a core tool that computer science students should have exposure to as they gain competency in tools and environments.

II. BACKGROUND

"Git has become a widely used tool for enabling collaboration – in an industry-wide survey of over 16,000 data scientists conducted by Kaggle, 58.4% of the 16,000 respondents said that Git was the main tool used for sharing code in their workplace." [3] GitHub is a social coding site that utilizes Git commands for repository management. "Despite GitHub's growing presence in programming classrooms, to date, few studies have explored how GitHub and the design of its implementation shape students' learning outcomes and classroom experiences." [4]

Lawrance, Jung, and Wiseman [5] citing Reid and Wilson, integrated Git into a CS1 classroom for non-CS engineering majors. In their conclusion, "Although distributed version control often confused students, they had little difficulty in adopting its workflows and never asked "when am I ever going to use this?". The paper did not mention any processes for evaluating what students learned. It only highlighted their ability to use Git which would in turn be useful in a future career. To be clear, Git is the version control system underlying GitHub. GitHub presents its usage through a graphical user interface.

Many studies highlight GitHub's usefulness as a mechanism for improving collaboration, supporting group work and introducing students to industry tools. [6] [4] Rocco and Lloyd [7] discuss the advantages of introducing GitHub early in CS1 classes because it creates good software development habits. By following the software development process, it is implied that students develop their incremental, iterative

programming skills. The qualitative analysis of this particular study, however, only focuses on the subjective quality of the code and does not reinforce any measurable learning outcomes.

III. METHODS

At Spelman College, introductory courses' learning objectives include developing computational algorithms for problem-solving and team work skills; demonstrating algorithmic thinking in a variety of contexts and thorough examination of computing fundamentals; translating algorithms into high-level programming languages that execute. The students in those courses are STEM majors but not all major in Computer Science. To justify prioritizing the introduction of version control, through GitHub, into introductory HBCU classrooms there must be evidence that version control improves student learning outcomes. One course, Computer Science I, aimed at first and second year students, introduces GitHub into the classroom to familiarize students with version control. To gain a better understanding of how students' learning objectives were affected, GitHub was introduced systematically over the course of three semesters to observe student outcomes. Namely, the objective of improving their computational thinking skills as measured by improvements in their incremental programming skills. Students were predominately female and African American and the instructor was an African American female as well.

Observations over three semesters indicate that the majority of students had never been introduced to version control, Git, or GitHub. For many of those students, this was their first- or second-time programming. The combination of trying to master the fundamental skills needed for advanced programming and learning to use GitHub proved overwhelming for many. After refining the instruction method in Fall 2020, students were noticeably more capable of utilizing GitHub as an assignment submission tool but its effect on their incremental programming skills was still not clear. For each program the student's repository data was collected. Data points such as commit frequency and assignment grade were compiled for each assignment. Data analysis is currently underway to assess if there is a measurable quantitative impact on the student's learning outcomes.

REFERENCES

- [1] Code.org. More data and talking points for advocacy. [Online]. Available: <https://code.org/promote/morestats>
- [2] The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society. (2013) Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science. [Online]. Available: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf
- [3] Kaggle. (2017, 10) Kaggle 2017 survey results. [Online]. Available: <https://www.kaggle.com/amberthomas/kaggle-2017-survey-results>
- [4] C. Hsing and V. Gennarelli, "Using github in the classroom predicts student learning outcomes and classroom experiences: Findings from a survey of students and teachers," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2019, p. 672–678.
- [5] J. Lawrance, S. Jung, and C. Wiseman, "Git on the cloud in the classroom," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2013, p. 639–644.
- [6] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in github: Transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, New York, NY, USA, 2012, p. 1277–1286.
- [7] D. Rocco and W. Lloyd, "Distributed version control in the classroom," in *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2011, p. 637–642.