Enhancing Virginia Computer Science to Improve Real-World Experience

Analysis of the University of Virginia's Old Computer Science Curriculum

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Zachary Lieberman

December 9, 2022

Technical Team Members:
Zachary Lieberman

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Zachary Lieberman

ADVISORS

Benjamin Laugelli, Department of Engineering and Society

Rosanne Vrugtman, Department of Computer Science

Introduction

The goal of any university computer science (CS) education is to provide students with the skills and knowledge to succeed in their careers and contribute to society (*Computer Science (B.S.)*, 2021). These curriculums include foundational courses in wide-spread data structures, algorithms, and theory among other niche topics such as cybersecurity and artificial intelligence. These topics are important to learn, yet most students' priority in college is to acquire knowledge of skills and technologies currently used in an internship or full-time role. Despite this emphasis on real-world experience, universities are having difficulty implementing classes to teach current technologies and concepts for a multitude of factors. These factors include the difficulty recruiting industry professionals to academia and an increasing number of students taking computer science courses with limited classroom space in regard to both physical classrooms and enrollment capacity.

My experiential learning experience as a software engineer intern allowed me to discover those learning gaps between the material learned at the University of Virginia (UVA) and the tools I needed to use for my internship. Through this experience, I was able to identify key limitations in the UVA Computer Science curriculum which caused me to reflect upon potential improvements the Department could make. My technical project will describe some potential solutions to the issue of student readiness for career opportunities drawing from my own experiences and successful solutions from other institutions in this matter. In order to truly understand the reasons that change is needed in UVA's CS department, I will be analyzing the limitations of the old CS curriculum and the social aspects, such as different groups of students' priorities and the changing technologies used by employers, which transformed current courses that inevitably resulted in the formation of a new modern curriculum.

Both the social aspect of analyzing the failure of the old CS curriculum and the technical aspect of constructing a better curriculum are important in addressing the sociotechnical challenge of improving student's success in the field by providing a complete picture of the cause of the problems as well as some potential solutions to tackle this important issue. The remainder of this paper goes into further detail about two projects: a technical one that introduces ideas for altering the department's focus for the CS curriculum and an STS project that looks at the shortcomings of the previous CS curriculum as a result of stakeholder priorities.

Technical Project Proposal

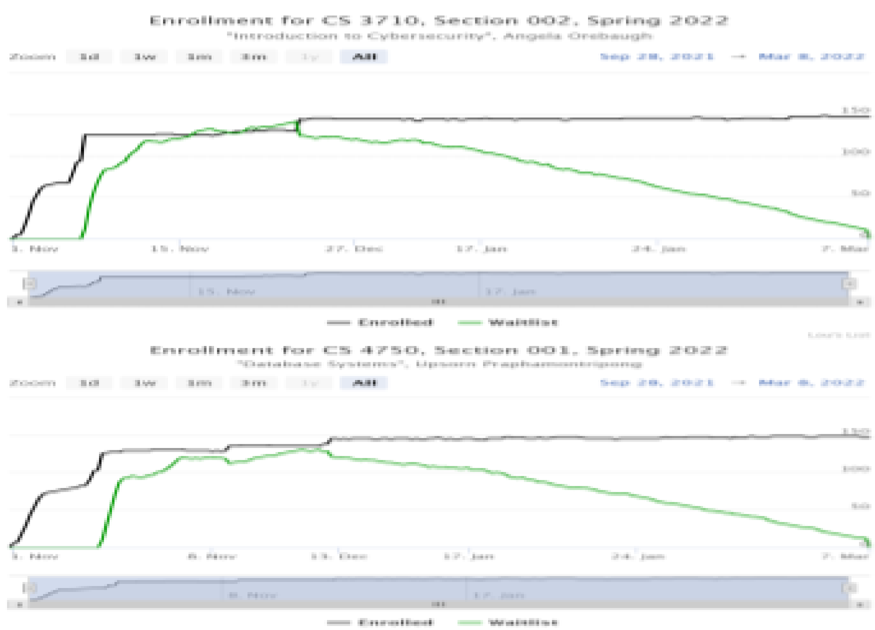
In the Summer of 2022, I held a full-time position as a Software Engineering Intern with the DC-based defense contractor, Leidos, in their Charlottesville Software Factory. The main tasks required of me in this role were to design, develop, troubleshoot, and debug automation solutions for development, test and production environments (Leidos: Careers, 2022). This includes understanding and developing source code and artifacts, tracking tickets, writing documentation, and maintaining continuous integration and continuous development (CI/CD) pipelines for various applications. The Agile work environment consisted of daily Scrum meetings with bi-weekly sprints, a period where tickets are defined for completion within sprint planning meetings to accomplish a project milestone completed by a sprint review (*What Is Scrum?*, 2022). The role consisted of a collaboration of frontend, backend, and cloud container technologies to accomplish the tasks at hand. Some courses that aided in my preparation for these duties were Advanced Software Development (CS 3240), Software Development Methods (CS 2110), Program & Data Representation (CS 2150), and Algorithms (CS 4102). Advanced Software Development was helpful in giving me an introduction to an agile work environment and some web development experience. From the Software Development Methods, Program &

Data Representation, and Algorithms courses I grew the foundations for various data structures and algorithms that are pertinent to the role and technical interviews.

The material from these various courses was important for some portions of my role, however, they did not cover a substantial portion of the responsibilities, skills, and tactics that required additional training, research, and expertise. The limitations in the CS Curriculum left me without experience in JavaScript and React.js, the main language and framework used in frontend development, and unfamiliar with expectations for large documentation efforts for the code base and investigation tickets at the industry level. The CS Curriculum was also structured to have various different topics covered and revisited in an order that made it challenging to remember concepts or have recent experience when preparing for interviews. For example, Java is one of the main programming languages used in backend development, however, the language was used almost exclusively in Software Development Methods which I took in my first

Figure 1

Enrollment Figures For CS Electives Spring 2022



Note. Enrollment data for popular CS courses from *Lou's List*.

semester causing me to forget key mechanics of the syntax in my interview preparation. Another common issue students like myself have is the omission of education on new emerging technology used in industry in our coursework. The results in Table 1 show that even if there are courses that do include these technologies, such as the special topics electives, there are not enough enrollment slots to accommodate many students, which cannot retroactively help prepare for internships.

The existing curriculum structure and computer science courses do not adequately prepare students for experiential learning events like my summer internship. Therefore, improvements should be made to properly prepare students for these experiences. Students should be able to access course materials and lecture recordings for popular courses so they are able to adequately learn and prepare for upcoming experiential learning events if they are unable to take the course due to capacity restraints. This has been incorporated in some courses such as Operating Systems (CS 4414) and can be restricted to only UVA students if the department wants to protect its proprietary resources and materials (Reiss, 2020). This would be an important implementation to the CS Curriculum as the foundational courses, usually taken as a first and second-year, cover material necessary for technical interviews, however, electives that are taken as a third and fourth-year student cover more material that is required during the experiential learning event (*Computer Science (B.S.)*, 2021). The CS Curriculum should also include more materials and assignments focused on code documentation and developing soft skills. These focuses are both important for internships and other events and can be easily integrated naturally and effectively with the other required concepts (McCauley et al., 1996). Finally, creating a more specialized education similar to Stanford's multi-path curriculum would allow students to have a strong advantage when entering the workforce in their respective fields

(*Computer Science Program | SoE Undergrad Handbook*, 2022). I will utilize syllabi from courses I have taken as well as courses from other universities, UVA's old and new CS curriculums and CS curriculums from other universities such as Stanford University, and materials on how institutions such as the Association for Computer Machinery (ACM) recommend creating and revising CS courses (*Computer Science Program*).

STS Project Proposal

The last significant change to the University of Virginia Computer Science (CS) curriculum occurred in 1989 with the introduction of the course *Program and Data Representation* (PDR) which is referred to as the Old Curriculum (Tychonievich and Sherriff, 2022, Curriculum History). This course was the only prerequisite for most of the curriculum as pre-PDR courses covered object-oriented coding and digital logic and post-PDR courses covered most systems and algorithmic concepts (*Computer Science (B.S.)*, 2021). Additionally, at the time of this curriculum overhaul, the only degree in the CS department was the Bachelor of Science (BS) Computer Science degree in the School of Engineering and Applied Sciences (Tychonievich and Sherriff, 2022, Curriculum History). These changes were highly regarded by both faculty and students and garnered high praise from accreditors and industry partners mainly due to the inclusion of PDR as the focal point of the curriculum (Tychonievich and Sherriff, 2022, Curriculum History). The UVA Department of Computer Science Brochure (2000) claimed: "[it] has been highly successful, providing a superior education for those students who will work in industry, as well as for those who will go on to graduate school" (p. 10).

However, the views held over 20 years ago overlooked how stakeholders' priorities and concerns shifted with the expansion of the program and changing ideologies in the industry. The

Department of Computer Science has grown drastically from 2014 having 28 faculty and around 250 majors to 45 faculty and approximately 700 majors in 2021 (Tychonievich and Sherriff, 2022). This includes majors from the Bachelor of Science in CS (BS CS), Bachelor of Arts in CS (BA CS), and the Bachelor of Science in Computer Engineering (BS CpE), but does not consider additional stakeholders such as students in the Computer Science Minor, transfer students, or other students taking CS courses (*Interdisciplinary Major in Computer Science*, 2021). This increased influx of new students caused the CS Department to marginally update the curriculum by changing requirements for new degree programs and revising and creating courses with current industry trends. However, this did not constitute a complete restructuring of the curriculum which disadvantaged students from gaining a cohesive and modern education plan (Tychonievich and Sherriff, 2022).

I argue that the design of the Old CS Curriculum did not succeed because it did not address the priorities and concerns of the faculty in the CS department, transfer students, students in the college of arts and science, administrators, accreditors, and industry partners. The curriculum design aimed only toward BS CS students limited transfer students and study abroad opportunities, differentiated course content between BS CS and BS CpE students, and hindered opportunities for some students with less computing experience among other concerns (Tychonievich and Sherriff, 2022, Why Make a Change? 4.0.1-4.0.5). I will use Social Construction of Technology (SCOT) to help frame my analysis as it looks at how human behavior shapes technology rather than the other way around. SCOT implores that relevant social groups or stakeholders exhibit varying interpretive flexibility, how different parties have used or interpreted a technology, due to their different priorities and concerns until there is a stabilization of the technology that accounts for a majority of the stakeholders (Pinch and Bijker, 1987).

Through the lens of SCOT, this project seeks to explain why closure was not accomplished because of shifting opinions and stakeholders within the University of Virginia's Department of Computer Science over the past 20 years. To support this claim, I will analyze evidence from Luther Tychonievich and Mark Sherriff's analysis of the Old Curriculum to determine the role that relevant social groups and stakeholders performed in the overhaul of the CS Curriculum (Tychonievich and Sherriff, 2022).

Conclusion

The technical report will focus on presenting effective improvements to the UVA CS curriculum drawn from my own experiences and previous implementations of curriculum improvements from other top engineering programs. The STS research paper will seek to analyze the social impact that led to the failures of the old UVA CS curriculum using the SCOT framework as a basis for the restructuring of the curriculum within the technical report. These reports together will address the underlying causes of the curriculum's failure from a sociotechnical view while introducing alternative measures to help improve CS education at UVA.

References

Bloomfield, L. (2022, December 9). *Enrollment for CS 3710 001 - Introduction to Cybersecurity*. Louslist.org. https://louslist.org/enrollment.php?Semester=1222&ClassNumber=16608

Computer Science Program | SoE Undergrad Handbook. (2022, December 9). Stanford Engineering. https://ughb.stanford.edu/majors-minors/computer-science-program

Department of Computer Science - University of Virginia. (2000). Computer Science Brochure [Brochure], Charlottesville, VA, United States

Leidos: Careers. (n.d.). Leidos. Retrieved December 9, 2022, from https://careers.leidos.com/jobs/10861719-associate-software-engineer

McCauley, R. A., Jackson, U., & Manaris, B. (1996). Documentation standards in the undergraduate computer science curriculum. *Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education - SIGCSE '96*. https://doi.org/10.1145/236452.236548

Pinch, T. J., & Bijker, W. E. (1984). The Social Construction of Facts and Artifacts: or How the Sociology of Science and the Sociology of Technology might Benefit Each Other. Social Studies of Science, 14(3), 399–441. https://doi.org/10.1177/030631284014003004

Reiss, C. (2020). *CS4414: Schedule*. CS 4414 Operating Systems. https://www.cs.virginia.edu/%7Ecr4bd/4414/S2020/schedule.html

Tychonievich, L., & Sherriff, M. (2022). Engineering a Complete Curriculum Overhaul. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3478431.3499287

University of Virginia - Computer Science (B.S.). (n.d.). Records.ureg.virginia.edu. Retrieved December 9, 2022, from http://records.ureg.virginia.edu/preview_program.php?catoid=52&

University of Virginia - Interdisciplinary Major in Computer Science. (n.d.). Records.ureg.virginia.edu. Retrieved December 9, 2022, from http://records.ureg.virginia.edu/preview_program.php?catoid=52&

What is Scrum? (2022, August 13). Agile Alliance. https://www.agilealliance.org/glossary/scrum/