CS at UVa: Curriculum's Platform for Future Success

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Ahmed "Dio" Suliman

December 2, 2022

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.

Dio Suliman

Technical advisor: Briana Morrison, Department of Computer Science
Technical Writing Instructor: Rosanne Vrugtman, Department of Computer Science

CS at UVa: Curriculum's Platform for Future Success

CS4991 Capstone Report, 2022
Ahmed "Dio" Suliman
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
diosulimana@gmail.com

Abstract
The computer science department at the University of Virginia tries its best to ensure that every undergraduate can hold their diploma with the knowledge that it will assist them after they walk off the stage. To this end, the program includes subjects such as computer architecture, algorithms, theory of computation, etc. By completing their major, students should know and be able to use various tools, software, algorithms, methodology, and more that will further their status as software engineers. However, this does not mean that the UVA program cannot be improved, as there are several issues that appear throughout the curriculum that need to be addressed. Some ideas for change include altering the pace of some courses and removing or modifying courses, among other ideas to make graduates more career-/industry-ready. Using an updated system would allow students to explore more options and paths that will prepare them for the success they want to explore.

1. Introduction/Background

During my experience in the UVA CS program, I have seen some areas in which the program prepares graduates for the real world and other areas that could use improvement. For example, Advanced Software Development was a project-based course that had students follow a scrum methodology in groups. This course is a basic version of how tech companies would actually run with similar forms of meetings, layouts, communication, deadlines, and more. However, a course with a similar format but less successful, in my opinion, was HCI in Software Development. HCI did not give students an opportunity to understand how the work should flow. Issues included odd deadlines, no rigid system or guidelines, lectures that were not substantial, and more.

2. Related Works

UVA School of Engineering and Applied Science (2002) course websites detail objectives taught and the breakdown of courses required in the computer science program. This information shows what courses can be removed or modified. Interviews from current and former computer science students, as well as articles and videos from former students, are also included. These are first-hand accounts of student opinions about portions of the curriculum that were poorly managed or felt like a waste of time; as well as sections that should be expanded.

3. Proposed Curriculum Design

The past system has tried to keep students on a path that would lead to success. Without electives, the old curriculum path would include:

- Intro to Programming (111x)

- Software Development Methods (2110)
- Discrete Mathematics (2102)
- Program and Data Representation (2150)
- Digital Logic Design (2330)
- Theory of Computation (3102)
- Computer Architecture (3330)
- Algorithms (4102)
- Advanced Software Development Techniques (3240)
- Operating Systems (4414)

Figure 1: Old Curriculum Path

Figure 1 above shows the path along which a student would progress, depending on which classes they have taken at that point. To change the design, I would remove Discrete Mathematics (2102). Discrete, a prerequisite for Theory of Computation (3102), focuses on introducing logic, proofs, and sets overall. This material is all reintroduced and reviewed again in Theory and the 2102 course did not have much impact other than using a programming language that, according to students interviewed, "was a nightmare." This class has little purpose as Theory overall took what was important from the class and taught it to students again.

Another course to remove is Digital Logic Design (2330). The purpose of the class is to teach students about designing a CPU from switches to gates and, last, to components. While it was an interesting course, I would recommend making it an elective instead, since most of the material is forgotten and not used again in the future unless one specializes in the field.

The next change I recommend is moving Operating Systems (4414) to an elective. This course is only mandatory for students getting a BS so BA students do not have to worry about it. The course feels more like an elective as it goes over topics that one sees from Computer Architecture, Digital Logic Design which, along with networks, is being removed in this path. Most students would not need this information in in their respective job spaces, so why require that students take it?

The next course to edit would be dividing Computer Architecture (3330) into two separate parts. Comp Arch dives into the organization and architecture of a computer system's hardware. This course goes over a mountain of material and can overwhelm a student. Breaking the course into two sections would allow students to actually spend quality time learning instead of forcing them to memorize the key points they will may or may not need.

The last change I recommend would be allowing students to take more electives earlier. One of the earliest electives a student can take is HCI, which as shown above was a poorly run course and one I would recommend that students not take. Allowing students to take more electives early would

help broaden their knowledge of the vast sea that is computer science.

4. Results

The computer science field encapsulates services, subfields, and more that spreads over numerous topics. Removing and editing courses opens up space for students to take more electives and understand what they are learning without wasting time on topics that will not be of use to them.

Removing Discrete and Digital Logic Design takes away two mandatory courses that will not help them, opening space for more meaningful courses. Pushing Operating Systems as an elective allows students who wish to take it to do so, but also saves a spot for another course.

Although splitting Comp Arch into two takes another space for a course, it allows the material to be understood and actually be used when it comes to students thinking about how their code runs through hardware. These changes save space for more electives to be taken that will help students prepare for their future, as well as open up paths they may have not known about.

5. Conclusion

The old curriculum path tries its best to nourish and have students mature into a more useful pathway for life. Students use the material taught in the course in their career life after graduation. However, some changes to the curriculum path would allow for an easier transition, along with avoiding the need to learn material that will not be used after the course is over. Adding, removing, or altering courses will better equip a student to handle the next arc of their life.

6. Future Work

Follow-up on this would require testing to determine how this new curriculum would launch. Would productivity increase? Would grades increase? Without further testing, anticipated benefits are all speculative.

References

Sean Lei. 2018. Honest Review of UVA Computer Science - YouTube. YouTube. Retrieved December 1, 2022 from https://www.youtube.com/watch?v=sQwQJ6oWIJ4

University of Virginia, School of Engineering and Applied Science. 2022. Computer Science Undergraduate Advising. (March 2022). Retrieved November 30, 2022 from https://engineering.virginia.edu/departments/computer-science/computer-science-undergraduate-advising#accordion656360

Sean Lei. 2018. Honest Review of UVA Computer Science - YouTube. YouTube. Retrieved December 1, 2022 from https://www.youtube.com/watch?v=sQwQJ6oWIJ4