Analyzing Societal Transitions in a Computer Science Curriculum Revamp

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Parker Hutchinson

Fall 2023

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

Advisor

Kathryn A. Neeley, Associate Professor of STS, Department of Engineering and Society

STS Research Paper

Introduction

There is high demand for software engineers. At the same time, the industry reports that the graduates produced by universities are lacking in key areas required for productivity in the workplace. The discrepancy in values present in this system creates societal deficiencies for students, employers, and consumers of technology products. Universities play a key role in alleviating this issue, as they provide the most significant form of education for these workers and can significantly change the trajectories of their knowledge with the curricula they provide. UVA specifically is at an inflection point as they educate more and more computer scientists; the total number of undergraduates enrolled in the major nearly doubled from 2017-2022 (UVA Engineering 2022).

Although the field is a mature one, computer science curricula are not explicitly tied to industry insights and needs. Students may be educated on old technologies and platforms or frameworks that are not applicable to their later roles. In addition, the educational system may fail to endow students with knowledge that they are expected to have by the time they start their jobs, like the ability to reason through problems without specific guidelines given.

Inadequately educated students are less likely to succeed and progress up to higher positions in software-related roles. They will have a harder time holding their job if they lack critical programming skills that others possess. They will have even more difficulty progressing up to leadership and management positions if they are unable to master the basics of entry level positions. Students who are provided a stellar education that prepares them with the skills they need to succeed on the job, on the other hand, will be more confident in their abilities and more capable of contributing to their teams immediately.

I will analyze the present state of computer science education by using the University of Virginia computer science curricula as a sample. I will investigate how the department addresses the needs of the various stakeholders who interact with it. I will seek to find improvements to be made that would better educate students in their thinking processes as full time software engineering employees. I will seek to consider the demands of students, professors, and industry stakeholders in a computer science curriculum. I will also look to consider how changes implemented in classrooms can propagate upwards into changes in the computer science department in which they take place and beyond.

Problem Definition: Computer Science Education Fails to Prepare Students for the World

Computer science is a unique field in its rapid growth and rapidly changing environment: new frameworks, languages, and technologies appear and enter the mainstream yearly, and professionals are expected to rapidly assimilate these concepts into their skillsets to remain competitive. These innovations and breakthroughs have enabled many advancements, and this trajectory will likely continue in the future. As technology has permeated all areas of modern life, employment opportunities and demands for students trained in computer science has also risen. The Bureau of Labor Statistics estimates that “overall employment in computer and information technology” will grow 14.6% from 2021 to 2031 (2023). This growth is not due to a small starting size, either; Data USA states that the size of the computer science workforce was “over 2M” in 2021 (2021). This quickly growing workforce has a unique challenge, in that the professionals must readily adapt to a changing landscape of new technologies and tools as they are invented and then adopted. This continuous learning necessity makes it difficult to adequately train students for the field based on a static set of curricula. Indeed, a Tufts report notes that there is a widespread “skills gap” in computer science, and although many students graduate into the field and apply for jobs each year there are still existing shortages which “will grow as organizations become more reliant on technology” (2022).

Although computer science is a field with huge demand for students, the educational curricula provided to students are often deficient in many areas. These deficiencies can culminate in professional errors and shortcomings for graduates. Garousi et. al note that many fresh computer science graduates “face difficulties when beginning their professional careers”, and that these difficulties are a result of a “misalignment” between the topics they are taught in their university education “and what is needed in industry”. Garousi et. al noted that they saw the “greatest reported knowledge gaps in the areas of configuration management, SE [software engineering] models and methods, SE process, design (and architecture), and testing” (2020). This poses a major problem, as society depends more and more on computing in every sector of the economy and personal life. These deficiencies may propagate at an increasing rate as more and more students enroll in computer science classes, and as more and more graduates take on employment in the software field. In addition to this mismatch of education and industry needs, the educational field has difficulty in teaching various topics with the same rigor, and some topics are taught with more effectiveness than others. Authors in a study of software engineers found that subjects like “computer science fundamentals, software design, and mathematical concepts” were adequately addressed by the prior education of the engineers studied. However, the authors also noted that topics like “the World Wide Web, software engineering (SE) components, and computer graphics” were lacking in terms of educational preparation (Liargkovas et. al 2022). Clearly, computer science curricula can improve in their actual procedural teachings, but improvements can also be made in how they teach their computer science students to think. In "Cognitive Biases In Software Engineering: A Systematic Mapping

Study", Mohanani et. al analyze cognitive biases, defined as "systematic deviations from optimal reasoning", and how they can become a source of "software project challenges and failures" (2020). The authors argue that there is a "scarcity of research on mitigation techniques" and a lack of understanding for these biases. They also argue that "specific bias mitigation techniques" are needed for software engineers to avoid the negative effects of these shortcomings in the thought process on their work (2020). A modern computer science curriculum can address these biases before they become intertwined with a student's work by proactively teaching bias mitigation techniques and other methods. Full-time software engineers are also expected to "work with uncertainty" which requires "working with people" to gather necessary requirements, unlike the assignments students complete in school with "proper, structured, simple instructions" (Kravcenko 2023). Some of these skills may be difficult to teach, but they will provide great benefits to students in their preparation for the workplace.

Universities are spending lots of time, money, and effort to try to address the issues outlined above. More students are taking computer science courses than ever before, and the trend of growth is expected to continue. Over 100K bachelor's degrees were awarded in computer science and IT in 2020-2021 (NCES 2021). Narrowing down to computer science specifically, 59,565 degrees were given in 2021 (Data USA 2021). With such a high amount of students enrolled in these programs, any improvements are certain to benefit many both immediately in the educational process as well as later, when they apply their education as professionals in the industry.

This research will seek to find an improved way of educating computer science students in today's society, with its dynamic values and demands. Work will need to be done to ensure that the values of all stakeholders are balanced and needs are met. For example, efficiency is a key aspect of education. However, care should be taken to not create an excessively burdensome curriculum for the students, while also ensuring that they receive an adequate education as a simultaneous priority. Steps should also be taken to continuously incorporate feedback from industry professionals and other employer stakeholders into the curriculum.

Research Approach

I seek to analyze the key actors in the UVA computer science educational system, and to propose possible improvements to be made which would benefit the actors at play in that context. Multiple sources were considered as I researched how the computer science curriculum at the University of Virginia can be revised in an ethical manner. Of the articles under consideration, Geels' paper titled "The Multi-Level Perspective on sustainability transitions: Responses to seven criticisms" holds the most promise for my future research. In this paper, Geels analyzes the necessarily complex nature of sustainability transitions, which include "multiple actors" and "deep-structural changes" in existing societal frameworks. Introducing significant changes to the computer science curriculum would create a destabilizing condition for the existing educational system and the many actors who play a role in it. Such a departure from the status quo bears important similarities to the changes in society which Geels studies and reports on when analyzing sustainability transitions, and thus this source is extremely promising as a foundation to generate an approach for my research. Such transitions, however, would be necessary in order to provide the changes necessary to solve the current problems with computer science education. Students would need to be taught in a different manner, and the changes would need to be replicable enough to spread to other institutions to support the rapidly rising number of students majoring in computer science.

In “The multi-level perspective on sustainability transitions” Geels presents a theory which “conceptualizes overall dynamic patterns in socio-technical transitions” so as to better understand how societal changes can begin and take root. Geels’ insights are valuable to understanding computer science education and the transitions possible within it because it has many similar key actors as the domains he analyzes, and it allows for the analysis of structural change. Geels notes that MLP is similar to an analysis of techno-economic paradigm (TEP) shifts, but also points out that the two systems are different in scope: TEP focuses on economies, whereas MLP is more pragmatic, “focusing on concrete energy, transport, agri-good systems” and so on. At a high level, Geels describes the MLP system as a tool for conceptualizing “overall dynamic patterns in socio-technical transitions” (3). In this framework, transitions are “non-linear processes” which feature interplay between niches, socio-technical regimes, and a socio-technical landscape. Regimes in this system refer to the set of rules which actors abide by and which guide the activities of social groups in socio-technical systems. Niches are areas with special demands and protections, and where innovations and breakthroughs typically emerge. The socio-technical landscape is the “wider context” behind everything which “influences niche and regime dynamics” (5). Notably, MLP has no notion of single causality in its transitions. Transitions are influenced and “caused” by a variety of factors on multiple dimensions, a concept Geels refers to as “circular causality” (6). Changes in society begin as seeds in the niches, align and stabilize as they become integrated with society, and eventually break through to become incorporated into the existing socio-technical regime with an adjustment. This new form of the regime in turn goes on to influence the socio-technical landscape of which it is a part. This landscape produces external influences on niches, and the cycle continues. A visualization of this process is provided in Fig. 1.

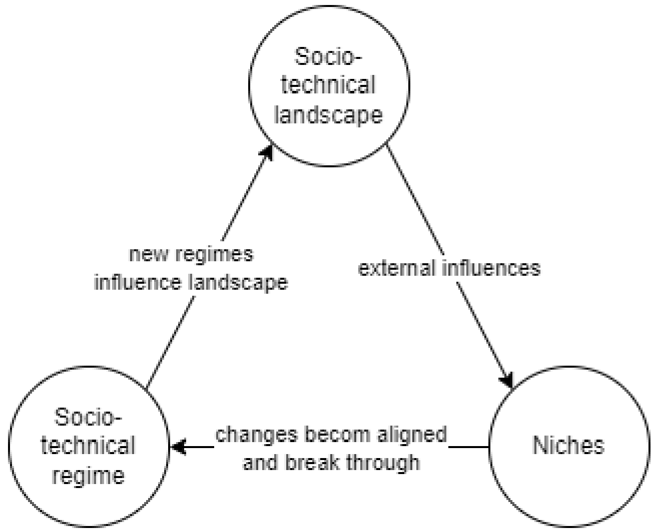

Figure 1: Interdependent interactions in MLP.

Although Geels originally discussed MLP as being a tool to analyze transitions in environmental sustainability, the categories he utilized are relevant to analyzing the players in the UVA computer science curriculum to later consider how improvements could take place. To transpose his categories and locate the relevant actors and their relationships in an educational setting, I will call upon the "Figurational sociology" method utilized by Scanlon et. al in "A Figurational Analysis of Teachers and Students as Policy Actors in Policy Enactment". In their article, the authors seek to understand the educational actors in "pluralities" or "figurations". They note that the mutual dependencies between groups of students and educators and their "socially generated reciprocal needs" means that they only truly exist in these "pluralities". Thus, by accurately locating them within these groups, a deeper understanding can be reached on how these groups enact policy and how their interdependent relationships influence such enactment. These groups and their relationships are of paramount importance to my research, since I hope to discuss the actors and relationships themselves and to note how policy change could be made, and synthesizing Geels' MLP with Scanlon et. al's findings would be of great use to understand both of those.

I will discuss how Scanlon et. al perceive teachers as fulfilling multiple roles throughout the policy enacting process to relate their analysis to my research. I have analyzed and visualized the most prominent figurations from Scanlon et. al's paper relevant to the UVA computer science department in the visualization below. I have specifically chosen the roles from the research which are most relevant to my research: narrators are related to the UVA administration and outreach teams, outsiders could be parents, employers, and other educational leaders, and insiders could be junior teachers and others directly involved with the educational process.

Role	Description
Narrators	Interpret and select meanings (primarily senior leadership team)
Outsiders (entrepreneurs & enthusiasts)	Partnership, ideation, satisfaction
Insiders (Transactors, Receivers)	Accounting, monitoring, junior teachers, implementation, student learning

Table 1: Teaching Roles in Scanlon et. al's figurations.

To apply these roles to my research, I will note subdivide the roles of Geels' socio-technical regime with them, allowing for more fine-grained analysis of how changes can be initiated and later enacted. This will allow me to paint the relationship teachers and students have with the overall context with a finer brush, allowing their unique abilities and roles in the policy creation and implementation process to be appreciated.

Results

As discussed, Geels' MLP is an effective tool for analyzing societal transitions, and a change to the UVA computer science curriculum is another use case. I will utilize MLP to discuss how changes and influences may take place alongside a curriculum change at UVA, and more specifically to identify the barriers and catalysts for such a transition.

The curriculum change will originate from the middle of the social organization, with the socio-technical regime (which in this case embodies the UVA academic system). The CS department and professors would choose to take on a new approach to CS education which addresses the demands and needs discussed before which are not currently met. This initiative would create an influence on the "niches" of the social domain as modelled by Geels.

Once the socio-technical regime has emplaced the new curriculum and set the stage for the classes to take place, the system will be tested, evaluated, and refined. The classrooms in which these new classes and teaching methods take place are the 'niches' which Geels discussed in his model. The changes that could take place are general, and may be different from classroom to classroom. Some professors might try a new teaching style in an effort to achieve the goals laid out by the department, whereas others might keep their usual routine but utilize new teaching materials. Professors may also choose to use entirely new technologies to aid their teaching. The industry could serve as a source of tried-and-true ideas in this area; for example, CodeSignal assessments, described as the "leading technical interview and assessment solution" for the industry, are commonly used to assess software engineering candidates' skills in algorithmic development and general computer science knowledge, and these assessments might gain traction in the classroom because of their objective and quantitative measurement of a candidate's aptitude in computer science skills (CodeSignal 2023). A professor may choose to administer this exam at the beginning and end of the semester to quantitatively measure their student progress and assess their readiness for work in the industry.

As Geels shows in his MLP on societal transitions, some of the changes which take place in the classroom will gain traction, whereas others will die out. New ideas, methods, and practices in teaching will “become aligned” with what works for society, and continue to iterate upon themselves until they finally break through into a societal change, taking place in the socio-technical regime. For example, if CodeSignal assessments were found to be useful by one professor, discussions about this success would likely result in more professors trying the idea and finding new ways to refine the idea. If the CodeSignal assessments continued to provide benefit to all parties involved, the change would continue to spread and develop until it finally becomes a foundational aspect of the UVA computer science department. Changes would also prompt more changes, allowing others to recursively redefine the curriculum as they see fit in the spirit of creativity.

Once changes have ‘broken through’ to the socio-technical regime of the computer science department and curriculum at UVA, they would be foundational ideas which other developments build off of. This socio-technical regime would be pressured and influenced by the college and society at large in various ways. We have already discussed some of these pressures as those coming from industrial leaders and stakeholders. Specifically, universities are being pressured to educate more and more students while simultaneously educating them in an effective and thorough way such that they are ready to produce for their teams upon becoming employed for a software engineering role. Society’s demands may further change, however, and these shifts will in turn affect the regime’s policies towards niches, creating opportunities for new ideas to grow and break through.

Likewise, the regime will influence the broader society at large. As policies take place and are evaluated, their successes and failures will be noted. Ideas present in the policies which align well with the goals and values of stakeholders will break through to the regime, creating a new structure entirely which will prompt influence on society. This could, for example, take the form of other institutions borrowing beneficial aspects of UVA's new computer science curriculum, eventually resulting in a shifted educational method at college institutions across the nation.

A visualization of the dynamics at play with a change to the UVA computer science curriculum is presented below. The implementational actors are listed on the right-hand side, and their roles in Geels' MLP are shown simultaneously on the left. Societal changes are shown as progressing primarily upwards, from the niches of classrooms up to the broader nationwide educational system.

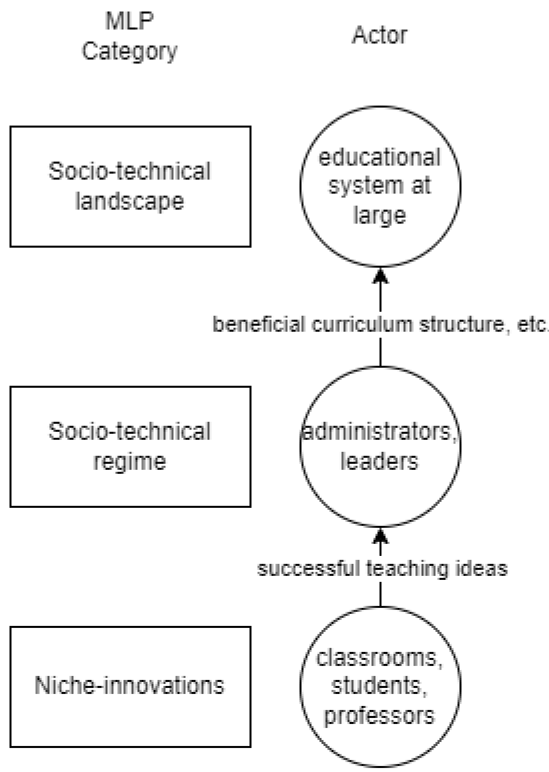

Figure 2: Visualization of influence pathway from classrooms to society at large.

Likewise, the visualization shown below shows how influence can travel in the opposite direction. In this example, industrial stakeholders are considered as the socio-technical landscape. Their demands are levied upon educational institutions, which are prompted to revise curricula and courses to meet these goals. The ideas are actualized in classroom niches where professors test out new teaching items.

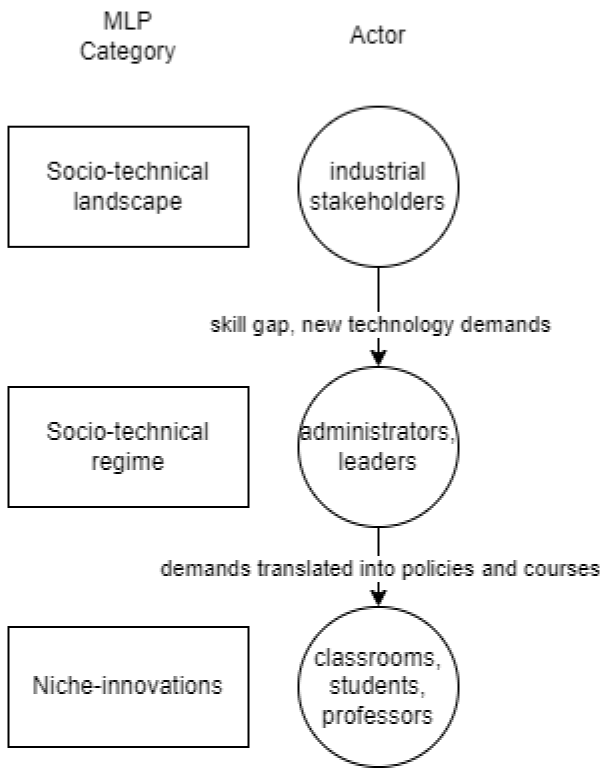

Figure 3: Visualization of influences from industrial stakeholders to classroom implementations.

After analyzing the flow of influences in the educational context of UVA, one may now consider various catalysts and resources for this change taking place in the computer science curriculum. Geel's analysis helps us to understand what factors are at play and what the possibilities are for transition within this socio-technical context.

Administrators at UVA may seek to enlighten themselves with the input of outsiders to gain the best insights of what employers need from their students. This is a method already employed by other departments at UVA to ensure that their graduates exit with the skills necessary to both perform well on the job and maintain competition with other applicants. University administrators may administer surveys to employers on career fair days to gauge what they would like to see more of from applicants based on what they saw in their search. These fairs already take place multiple times a year with hundreds of employers; 8 career fairs took place over late August to September alone at UVA (Palmer 2023). Thus, implementing this survey would not require a fundamental shift in any practices utilized by either the University or the employers interested in hiring graduates.

Another possible source of experimentation for UVA could be the use of a co-op initiative prior to graduation. This is a method used by other institutions to ensure that students graduate with real-world experience in the hopes that such exposure will contribute to their knowledge and readiness after graduation. Such a program is utilized nearby at Virginia Tech, with around 100 students enrolling in their version of the co-op program every semester. In their system, students have the option of either enrolling in a full-time work program for 32-40 hours per week, or a part-time program for 4-31 hours per week (Virginia Tech 2023). Such a split allows students the option to continue taking classes alongside their work program if they desire. This two-option system would likely be a beneficial choice to provide for UVA students as well, since some may want to continue taking classes at the University while working. Co-op programs are also financially beneficial to students, and can help them to pay “a portion of his/her college expernses” while simultaneously having the opportunity to “work with state-of-the-art equipment and use the most current procedures” (Calhoun Community College 2023).

There are barriers in place to such a change to UVA’s computer science curriculum, however. A key goal of industry exposure prior to graduation, as entailed by a co-op program, would be the gain of domain-specific or job-specific knowledge. This enlightenment would be difficult, if not impossible, for professors and administrators at UVA to measure, due to the unique nature of every job as well as the possible presence of proprietary information. Thus, the subjectivity of this experience and the difference in effectiveness for different students would likely make this harder to implement across the entire computer science student population at UVA, though not impossible.

Another difficulty in such a transition would be finding the best way to address transitioning needs of the workplace. Although surveys can be repeatedly given to employers to assess what technologies are taught to students, in reality students cannot be taught everything they'll ever be expected to know in their jobs after graduation. Thus, a key goal of the computer science department (and indeed, one that is likely currently held) would be to endow students with the ability to rapidly adapt to new situations and learn new technologies, integrating these unknown features with a plethora of foundational knowledge. However, this skill would be a hard one to measure over a semester-long course, and thus administrators may have to find an innovative way to help strengthen this cerebral muscle in students while also ensuring that their foundational computer science knowledge is not flaunted.

Conclusion

The effects of a change to UVA's computer science curriculum have been analyzed through the lens of Geels' MLP. Using this, one can thoughtfully play out the effects of a change to the policies in place at UVA, and more critically analyze the factors which would boost or prevent such a change from taking place. Clearly, changes which take place at UVA can go on to impact much more than just the students and professors which take part in them in classrooms. With thoughtful consideration and policy enactment, UVA can allay the skills gap in computer science with a revised educational curriculum.

References

Bureau of Labor Statistics, 2023. “Occupational Projections Overview, 2021–31 : Monthly Labor Review: U.S. Bureau of Labor Statistics.” Www.bls.gov, www.bls.gov/opub/mlr/2023/article/occupational-projections-overview-2021-31.htm#:~:text=Also%2C%20employment%20in%20computer%20and.

Calhoun Community College. “Benefits of the Co-Op Program.” *Calhoun Community College*, 2023, calhoun.edu/student-services/career-services/cooperative-learning/benefits-of-the-co-op-program/#:~:text=Allows%20the%20student%20to%20gain. Accessed 29 Oct. 2023.

“CodeSignal - Technical Interview & Assessment Solution.”, 2023. CodeSignal, codesignal.com/.

Data USA, 2021. “Computer Science”. Datausa.io, datausa.io/profile/cip/computer-science-110701.

Garousi, V., et al. "Closing the Gap Between Software Engineering Education and Industrial Needs." IEEE Software, Software, IEEE, IEEE Softw, vol. 37, no. 2, 1 Mar. 2020, pp. 68 - 77.

Geels, F. W., 2021. The multi-level perspective on sustainability transitions: Responses to seven criticisms. Environmental Innovation and Societal Transitions, 1(1), 24–40. https://doi.org/10.1016/j.eist.2011.02.002

Kravcenko, Vadim. “Things They Didn’t Teach You about Software Engineering.” *Vadim Kravcenko*, 6 Jan. 2023, vadimkravcenko.com/shorts/things-they-didnt-teach-you.

Liargkovas, Georgios, et al. "Software Engineering Education Knowledge Versus Industrial Needs." IEEE Transactions On Education, vol. 65, no. 3, 1 Aug. 2022, pp. 419 - 427.

Mohanani, R., et al. "Cognitive Biases In Software Engineering: A Systematic Mapping Study." IEEE Transactions On Software Engineering, Software Engineering, IEEE Transactions On, IIEEE Trans. Software Eng, vol. 46, no. 12, 1 Dec. 2020, pp. 1318 - 1339.

NCES, 2022. “Digest of Education Statistics”. https://nces.ed.gov/programs/digest/d22/tables/dt22_322.10.asp

Palmer, Heather. “Fall 2023 Networking Nights & Career Fairs.” *University of Virginia School of Engineering and Applied Science*, 26 July 2023, engineering.virginia.edu/fall-2023-networking-nights-career-fairs-save-date.

Scanlon, D., MacPhail, A., & Calderón, A. (2023). A figurational analysis of teachers and students as policy actors in policy enactment: ‘Integrated Learning Experiences’ in examinable physical education curriculum. European Physical Education Review, 29(2), 308-328. https://doi.org/10.1177/1356336X221139945

Tufts, 2022. “How to Stay Competitive in Computer Science.” Onlinesoe.tufts.edu, 2 Mar. 2022, onlinesoe.tufts.edu/blog/is-computer-science-oversaturated.

UVA Engineering. “Enrollment Statistics.” *University of Virginia School of Engineering and Applied Science,* 14 Oct. 2022, engineering.virginia.edu/current-students/current-undergraduate-students/degree-information/enrollment-statistics.

Virginia Tech, “Cooperative Education and Internship Program (CEIP).” *career.vt.edu*, 6 Oct. 2023, career.vt.edu/experience/ceip.html#majors.