IMPROVING CS EDUCATION AT THE UNIVERSITY OF VIRGINIA BY INTEGRATING ELECTIVES

EFFORTS TO SUCCESSFULLY ADD ETHICS TO CS EDUCATION

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
Matt Koehler

October 27, 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.

ADVISORS

Catherine D. Baritaud, Department of Engineering and Society

Briana Morrison, Department of Computer Science

Computer science (CS) is a field of study that is constantly going through change and because of this the education of computer science is also forever changing. Since computer science is a relatively new field, there are many problems within its education. The effect of these problems within education has an impact on computer science graduates as they often lack skills to enter the workforce and have a lack of knowledge on ethics within software engineering. This has a trickle down effect that eventually impacts society as less qualified software engineers can lead to bad software with ethical problems being developed and used by society. The motivation for this research is to look at computer science education and propose a better way to teach computer science so these problems never show up in the first place.

At the University of Virginia (UVA), computer science electives are just an assortment of different computer science topics. The objective of the technical report is to propose integrative electives in the Department of Computer Science at the University of Virginia that better prepare computer science graduates for the workforce as software engineers. Also, learning ethics along with computer science is as important as ever so successfully adding ethics learning into computer education is vital. The Science, Technology, and Society (STS) project will look at how to successfully add ethics to a computer science education. It will do this by examining teaching computer science ethics through the handoff model framework. These topics are loosely coupled as they both look at computer science education and ways to improve upon it; however, the topics look at two different problems within computer science education. The primary author for both the technical report and STS project is Matt Koehler. Other personnel for these reports will be the advisor for the technical report, Briana Morrison an associate professor in the Department of Computer Science, the reviewer of the technical report, and Catherine Baritaud a senior lecturer in the Department of Science, Technology and Society. The work for the technical report will be accomplished in the Fall 2022 semester while the work for the STS project will begin in the Fall 2022 semester and finish in the Spring 2023 semester.

IMPROVING CS EDUCATION AT THE UNIVERSITY OF VIRGINIA BY INTEGRATING ELECTIVES

There are a lot of gaps between what is required of a software engineer and the education of computer science majors. In particular, engineering computer science students at the University of Virginia can have a hard time fully incorporating into their work all the knowledge they have gained during their study. This problem is not unique to the University of Virginia as many computer science graduates can often lack the necessary skills to enter the workforce as software engineers (Liebenberg , Huisman, & Mentz, 2014, p. 2604). This is significant as allowing for a smooth transition from college to post-college life is vital to the success of a young person. Furthermore, the first job a person gets post-college can have a massive impact on their life, so giving students at UVA the best education to prepare them for this job would be very beneficial (Nova, 2018). The overall objective of this technical report will be to propose a better curriculum by specifically integrating electives to better prepare students from the University of Virginia for the workforce.

The University of Virginia currently has a curriculum for the BS major in computer science that attempts to give its students the best education and preparation for the workforce (University of Virginia, 2022). The problem with this curriculum is that the electives leave a lot to be desired as they are not very cohesive to the core curriculum CS classes and other CS electives a student will take. In order to combat this, integrating the electives to be more cohesive with the rest of the learning a person does within the BS major could be very beneficial to learning the necessary skills to become a successful software engineer after graduation. As seen below in Figure 1, a comparison between the current computer science curriculum and the proposed computer curriculum is made to illustrate how currently the CS curriculum has electives that have no connection to each other and to the core CS curriculum. However, the proposed plan is shown to have electives that are integrated with other electives, and the integrated electives also have a stronger connection to the core CS curriculum. A class that could possibly be offered under this proposed curriculum would be machine learning in mobile app development. This is because many top mobile applications in use today such as Facebook and Snapchat use machine learning and students would be able to utilize both skills at once which could be very beneficial to future software engineers who could possibly work with these types of apps (Bushkovskyi, 2022). The approach on how to integrate electives would be taking two current electives offered at UVA and combining them into one class that would allow UVA students to tie multiple topics together which will ultimately help their overall education.

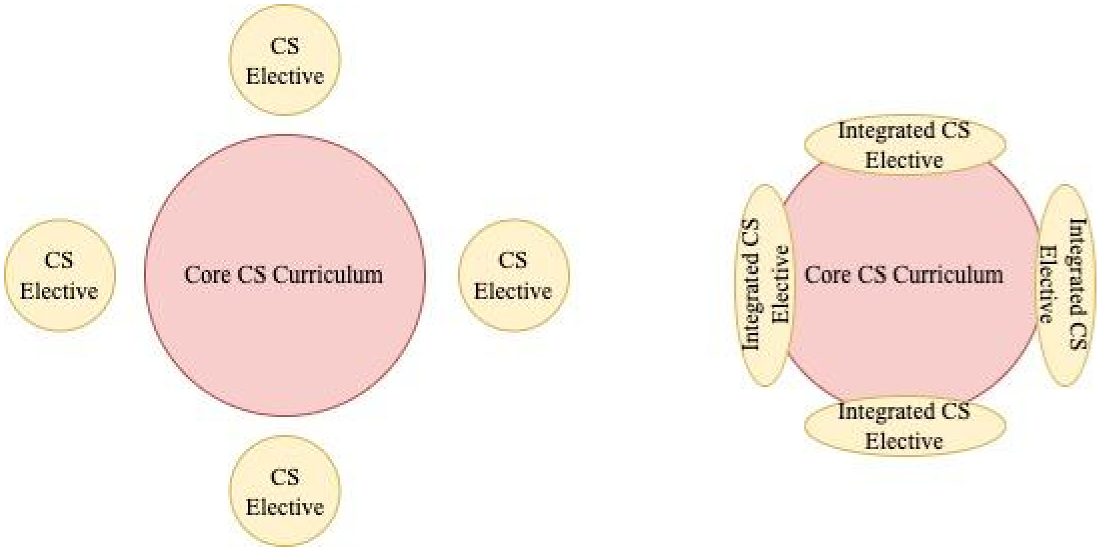

Figure 1: UVA CS Curriculum. The figure visualizes a comparison between the current CS curriculum and the proposed CS curriculum (Koehler 2022).

Proposals for classes like this already exist such as a class proposed by Hollingsworth and Powell that integrates mobile apps and cloud computing (2011, p.20). There is a lot of evidence to suggest that this proposal could be successful because integrating the electives would be akin to interdisciplinary learning which has been proven to be very successful in a lot of cases (Jones, 2010, p. 76). The anticipated outcome would be that UVA CS students would be better prepared to be software engineers post-graduation since they would be able to relate many different computer science topics, such as machine learning, together with their core computer science knowledge which can give them a good basis from which to work with. This outcome would happen instead of the current outcome which is current computer science graduates having deep knowledge of diverse CS topics but having no way of relating them to their other CS experiences. This project will be written as a technical report.

EFFORTS TO SUCCESSFULLY ADD ETHICS TO CS EDUCATION

Ethics in software engineering has always been a pressing topic because of how much software affects society. For example, there are many ethical dilemmas facing software engineers today such as privacy, algorithmic bias, and data collection (Sweeney, 2022). Furthermore, even if a person makes a great app it runs the danger of the app becoming too addictive which can lead to disastrous consequences such as a lot of the problems with social media (Lawton, 2020). So, it is important to get to the root of this beginning with the education of software engineers on ethics (Narayanan & Vallor, 2014, p. 23). Gülcan in particular emphasizes the importance of ethics and its education as education is more than just concepts but becoming a better human (2014, 2625). It is imperative that future software engineers are taught ethics and while many computer science programs have classes for ethics it is necessary to find the best way to implement ethics into computer science education. This STS topic seeks to find out how to successfully add ethics to computer science education. There is an ongoing conversation on how exactly to implement ethics in schools; for example, The University of Virginia which has classes just for ethics on the side and not included within the core CS curriculum (University of Virginia, 2022). However, Melo and de Sousa (2017) are critical of this and say ethics needs to be implemented within the computer science classes themselves (p. 45). A new curriculum made a suggestion similar to this as it said to implement ethics within the core computer science classes (Hadfield et al., 2019). This is seen in Figure 2, where there are many differing ideas on how to implement ethics into the computer science curriculum and whether to just have classes on the side or have the core CS classes themselves have ethics within them. This subject needs further research as there is no consensus on what method of teaching ethics is the most effective for students. It is imperative to learn a better method for the teaching of ethics in CS as it will lead to future software engineers being ethical in their futures which will lead to more ethical technologies being created than previously.

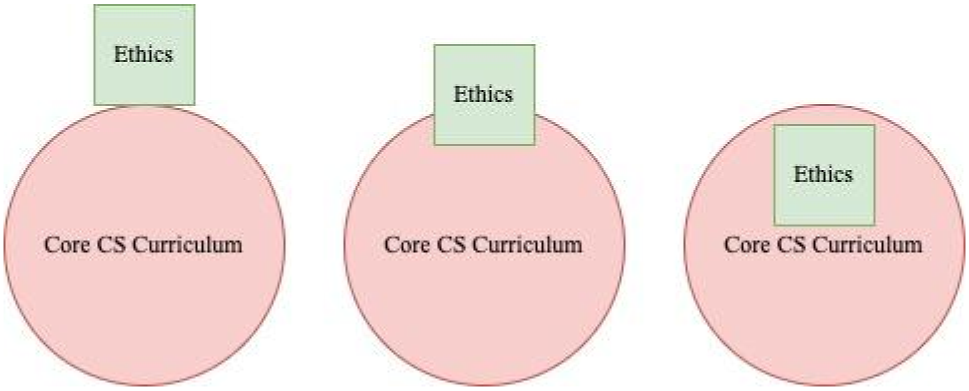

Figure 2: Differing Ideas On Ethics In CS Curriculum. The figure illustrates the differing ideas on how to implement ethics into the core computer science curriculum (Koehler 2022).

The objective of this STS project is to evaluate the different approaches to implementing ethics within computer science education and come to a conclusion on the best method. As seen in Figure 3, this can be further investigated using the STS Framework: Handoff Model from Carlson. This shows how learning is passed down in computer science and more importantly learning ethics. In the model, each chain is possible to stop but as it goes along it is harder and harder to stop the effects that come with it. This is important when considering the diffusion of technology as it is very important for ethics to be spread in the correct way because if it is not the future software engineers in this chain could potentially produce unethical software which is very hard to reverse the effects of. Ideally, were each handoff to be done correctly, the resulting effect would be the creation of ethical software as each future software engineer would have an incorporated basic understanding of the method of how to apply ethics to their work. This research will focus mostly on the first thing in the chain, the curriculum, as this will have the biggest effects going forward because if the curriculum is wrong everything after in the handoff model will also suffer. However, the other parts of the chain will not be ignored as looking at how these parts of the chain are affected by the CS curriculum must be a part of the research. This is because this will be the best way to evaluate a certain CS curriculum. The Handoff Model will be the main approach taken on this STS project.

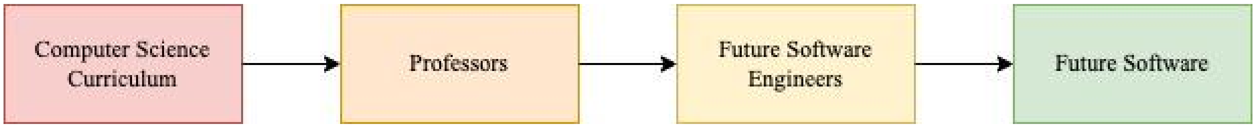

Figure 3: Computer Science Learning Using The Handoff Model. This figure illustrates how computer science knowledge and more importantly the ethics that come along with it are passed along. The colors represent when the effects can be stopped. When a future software is made it would be hard to stop thus it is green but a CS curriculum is much easier to stop (Adapted by Koehler (2022) from Carlson, 2009).

The desired result is that all ethics incorporated within the computer science curriculum is successful as this means that a massive amount of change will be necessary to the curriculum at least. However, this is not likely as one of the many different approaches to incorporating ethics to CS learning will likely be the best when compared to the rest. This will be written as a scholarly article that hopes to convince universities what is the best method on how to incorporate ethics into computer science learning.

COMPUTER SCIENCE EDUCATION

The field of computer science is always expanding. It is necessary to keep updating the curriculum in order to prevent future software engineers from getting too far behind. This updating of the curriculum needs to happen in a practical sense as computer science is always evolving with new languages and operating systems but also in an ethical sense because as computers become more powerful the effects of being slightly unethical can lead to a very disastrous outcome. As computers and computer science continue to grow that means the education of computer science must also continue to grow. This is imperative as new computer science graduates are behind not only on technical skills but also on ethical skills as well. A change is necessary here as continuing down the same path will lead to the creation of both bad and unethical software being created from these inexperienced software engineers. Fixing this problem of bad and unethical software can be done from the root, which is computer science education. This is the most efficient way to fix these problems as if we wait until the software is already made it will be too late. Allowing electives to be integrated will allow for computer science students to incorporate more of their base CS knowledge into their work which will better prepare them for the workforce. Additionally, incorporating ethics into computer science learning in a better way will also help these computer science graduates as they will be better prepared for the workforce. The combination of these two solutions combined will create entry-level software engineers that are far better than those produced by universities currently. Overall, computer science education must adapt just as much as computer science is adapting or it risks leaving those learning computer science behind.

REFERENCES

Bushkovskyi, O. (2020, June 24). The best applications of machine learning in mobile development. The Best Applications of Machine Learning in Mobile Development. Retrieved October 21, 2022, from https://theappsolutions.com/blog/development/machine-learning-in-mobile-app/

Gülcan, N. Y. (2015). Discussing the importance of teaching ethics in Education. Procedia - Social and Behavioral Sciences, 174, 2622–2625. https://doi.org/10.1016/j.sbspro.2015.01.942

Hadfield, S., Weingart, T., Coffman, J., Caswell, D., Fagin, B., Sarmiento, T., & Graham, P. (2019). Streamlining Computer Science Curriculum Development and assessment using the new ABET student outcomes. Proceedings of the Western Canadian Conference on Computing Education. https://doi.org/10.1145/3314994.3325079

Hollingsworth, J. and Powell, D. 2011. Requiring web-based cloud and mobile computing in a computer science undergraduate curriculum. *In Proceedings of the 49th Annual Southeast Regional Conference (ACM-SE '11).* Association for Computing Machinery, New York, NY, USA, 19–24. https://doi-org.proxy01.its.virginia.edu/10.1145/2016039.2016054

Jones, C. (2010). Interdisciplinary approach-advantages, disadvantages, and the future benefits of interdisciplinary studies. Essai, 7(1), 76-81.

Koehler, W. (2022). *Computer Science Learning Using The Handoff Model.* [Figure 2]. Adapted from Carlson, B (2009). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Koehler, W. (2022). *Differing Ideas On Ethics In CS Curriculum.* [Figure 3]. *Prospectus* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Koehler, W. (2022). *UVA CS Proposal.* [Figure 1]. *STS Research Paper: Improving CS Education At The University Of Virginia By Integrating Electives* (Unpublished undergraduate thesis). School of Engineering and Applied Science, University of Virginia. Charlottesville, VA.

Lawton, G. (2020, December 22). 5 examples of ethical issues in software development. SearchSoftwareQuality. Retrieved December 1, 2022, from https://www.techtarget.com/searchsoftwarequality/tip/5-examples-of-ethical-issues-in-software-development

Liebenberg, J., Huisman, M., & Mentz, E. (2014). Knowledge and skills requirements for software developer students. International Journal of Computer and Information Engineering, 8(8), 2612-2617.

Melo, C. D., & de Sousa, T. C. (2017). Reflections on cyberethics education for Millennial Software Engineers. 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM). https://doi.org/10.1109/secm.2017.10

Narayanan, A., & Vallor, S. (2014). Why software engineering courses should include ethics coverage. Communications of the ACM, 57(3), 23–25. https://doi.org/10.1145/2566966

Nova, A. (2018, July 2). Why your first job out of college really, really matters. CNBC. Retrieved October 27, 2022, from https://www.cnbc.com/2018/06/25/why-your-first-job-out-of-college-really-really-matters.html

Sweeney, M. (2022, February 11). Ethical dilemmas in computer science. ZDNET. Retrieved October 27, 2022, from https://www.zdnet.com/education/computers-tech/ethical-dilemmas-computer-science/

University of Virginia. (2022, March 31). Computer Science Undergraduate Advising. University of Virginia School of Engineering and Applied Science. Retrieved September 22, 2022, from https://engineering.virginia.edu/departments/computer-science/computer-science-undergraduate-advising#accordion656360