

Bridging the Diversity Gap in Computer Science with a Course on Open Source Software

Judy Weng and Christian Murphy
Department of Computer & Information Science
University of Pennsylvania
Philadelphia, PA
{judyweng, cdmurphy}@seas.upenn.edu

Abstract—Stereotype threat, impostor syndrome, lacking a sense of belonging, and misconceptions about the field are just some of the reasons that contribute to the increasing diversity gap in Computer Science. To address this, our institution has developed an undergraduate course in which students contribute to Free and Open Source Software (FOSS) projects under the guidance of a dedicated mentor. By contributing to FOSS projects, students can: find a mentor or role model; collaborate with, participate in, and contribute to a welcoming and supporting community; and see that they can have real-world positive impact. This paper describes the course and our experiences in teaching it, and provides evidence that it can have a positive impact on diversity by increasing retention and improving students’ confidence.

Keywords—Diversity, Computer Science Education, Free and Open Source Software

I. INTRODUCTION

Diversity and inclusion (D&I) has long been a problem within Computer Science (CS). Over the past few decades, as underrepresented minority (URM) groups within CS dwindle, the problem has become more prominent. Even URM groups who have relatively more advocates and initiatives, such as women, are still underrepresented and their representation has decreased [1]. Other URM groups, e.g. based on socioeconomic background or sexual preference, have little to no community, supporting initiatives, or even statistics to get an idea on their representation within CS. D&I is commonly seen as a pipeline problem that starts well before one enters the workforce, and therefore, it is more beneficial to start more academic initiatives designed to recruit and retain URM students.

To that end, we have developed an undergraduate course aimed at addressing D&I issues by having students contribute to Free and Open Source Software (FOSS) projects. The course exposes students to the cultural, technical, and legal aspects of FOSS development by covering topics such as: the need for and benefits of open source software; open source licensing and business models; intellectual property; and humanitarian free and open source software (HFOSS).

Most importantly, there is a semester-long project in which students become involved in and make meaningful contributions to a FOSS project under the guidance of a mentor who is part of the project’s community. This allows us to attract and retain more URM students and improve D&I by

giving students the opportunity to find a mentor or role model; collaborate with, participate in, and contribute to a community; and see the opportunity for real-world positive impact.

This paper describes the course and our experiences in teaching it, and provides evidence that it can have a positive impact on diversity by increasing participation and improving URM students’ confidence.

II. MOTIVATION AND BACKGROUND

Here we present an overview of some of the challenges with recruiting and retaining URMs in the field of computing.

A. Literature Review

With the increased attention to the CS diversity gap, many researchers have tried to understand its underpinning causes. There are a few common reasons most researchers and educators believe could be a factor.

1) *Stereotype Threat*. Stereotype threat is a social phenomenon in which people believe they are at risk of conforming to stereotypes based upon their social groups, and is often considered a reason people are discouraged from CS [2]. If people are exposed to negative stereotypes about their social group, they will end up performing more negatively, as has been shown in fields such as athletics, math, and even standardized tests [3]. In general, marginalized and disadvantaged groups face more negative academic stereotypes, especially in the STEM fields, and are at a higher risk of stereotype threat [4]. In addition, URMs are usually portrayed as not as capable in STEM and other quantitative fields, which has a correlation in both situations where it has shown a decrease in confidence [5] [6]. The effects of stereotype threat from such negative stereotypes can cause individuals to be less confident and doubt if they made the right decision in their choice of study.

2) *Imposter Syndrome*. Imposter syndrome is a common social phenomenon within CS, meaning people do not experience an internal sense of success because they attribute their achievements to external factors, like luck [7] [8]. URMs often believe they are less competent despite similar comfort levels, grades, and abilities in CS to their white male peers [7]. Many leaving CS to another major or career path believe they are not smart enough and URMs have a higher chance of having imposter syndrome [9].

3) *Sense of Belonging*. The sense of belonging is intrinsic to human nature and the need for it is stronger in adverse and stressful environments, such higher education [8] [9]. URM students may be discouraged from CS because they do not see others like them. The lack of community makes them feel unwelcomed and alienated from their peers [2] [7], and are a better fit for those who are part of the majority [2] [10]. Those who manage to find a supportive community claim that this support kept them from leaving during moments of doubt [7] [11].

4) *Lack of Diversity in Representation*. The lack of diversity in the representative figures within CS is a severe problem. Studies trying to find causes of the diversity gap suggest that increasing the diversity of representation within the field can help alleviate the effects of stereotype threat and lack of sense of belonging [2] [7]. When students choose a major, it is also crucial that they can envision themselves in that occupation [2] [9]. Having a relatable role model can inspire URMs to continue studying CS, while an overwhelming representation of successful people different from them can induce a stronger effect of stereotype threat [5] [8] [10]. Seeing their role models succeed can show students possibilities they never envisioned for themselves before.

5) *Misconception of Computer Science*. Students feel that the academic work within a CS curriculum is usually different from the real-world experience in industry. Moreover, some say they started CS with the misconception of what working in a CS related role was. They envisioned themselves sitting in front of a computer and rarely interacting with others [2] [12]. Students who left CS said their new major was a better fit because there was more interpersonal communication and social interaction [7] [12]. This is a common misconception of CS [13], but CS actually requires working in teams and also interpersonal soft skills [14]. This is a sign that students do not get enough exposure to the industry they are studying for [12].

B. Student Survey at Our Institution

In addition to reviewing the existing literature, we sought to get a better understanding of the situation at our own institution, particularly whether students who identify as URMs perceive aspects of CS differently from those who identify as non-URMs when it comes to making a major feel inclusive. This includes mentorship, relatable role models, collaboration opportunities, sense of community, ability to have an impact on society, opportunities to express ideas and creativity, and networking opportunities.

We surveyed students in our CS1 course as well as an upper-level undergraduate course, and asked them to respond using a 5-point Likert scale (with 1 labeled “Strongly Disagree” and 5 labeled “Strongly Agree”) to eight questions stated in the positive form: “*To what extent do you agree with each of the following: Computer Science is a field that offers me the opportunity to... (1) Find a mentor, (2) Find a role model, (3) Collaborate with others, (4) Participate in a community, (5) Contribute to a community, (6) Have a positive impact on society, (7) Express my creativity, (8) Connect with professionals.*”

Additionally, we asked whether the student identifies as member of an underrepresented minority in computing, with possible answers being “Yes,” “No,” and “I don’t know.” This prompt came at the end of the survey. Note that we did not explicitly ask for gender, race, ethnicity, etc., only whether the student identified as a URM.

In total, 33 students from the CS1 course and 15 students from the upper-level course completed the survey. Of the 48 respondents, 21 stated that they considered themselves part of an underrepresented group, 19 stated that they did not, and 8 answered “I don’t know.”

Because of the small sample size, we are unable to state the statistical significance of the results, but they do generally confirm many of our expectations:

- Across both courses, students who identified with a URM group felt that they were less likely to be able to find a mentor, have a positive impact on society, express their creativity, or connect with professionals.
- These differences were more pronounced in the students in the upper-level course than they were in the CS1 course, indicating that these misconceptions worsen over time.

Somewhat surprisingly, we also found that students who identify as URMs were *more* likely to agree that CS is a field in which they can collaborate with others, participate in a community, and make a contribution to a community, perhaps because those students have already formed peer communities through organizations such as WiCS, NSBE, SHPE, etc. Even though this did not match our expectations, surely it is important to continue to create interventions for our students that reinforce the notion that CS is a collaborative field.

Overall, these results demonstrate that there is need for our institution to attempt to address URMs’ perspectives of computer science and improve diversity and inclusion.

III. A COURSE ON OPEN SOURCE SOFTWARE

This section describes an undergraduate course that combines best practices of successful academic initiatives and contributions to open source software in hopes of maximizing their benefits and effects to retain URM students within CS and increase their confidence.

A. Why Teach Open Source Software?

Free and Open Source Software (FOSS) is software that is licensed to be free to use, modify, and distribute. The contributors are a mix of paid and voluntary programmers. Since FOSS is developed by a variety of developers, the product requires contributors to learn to communicate in a professional and realistic environment despite most of the communicate being virtual. Since the source code is public, there is complete transparency in all code, commit history, and documentation. The open source licensing allows the software to be modified by end-users. These factors make FOSS perfect for student contributors. Contributing to FOSS benefits students by teaching them technical and professional skills and

allows them to learn within a professional community and distributed development environment [22].

B. Course Overview

The course is offered at our institution as an undergraduate special topics course entitled “Open Source Software Development,” and is targeted toward students who have completed our traditional software engineering course.¹

Students indicate their preferences from a curated list of FOSS projects for which a mentor in the community has been identified, and then are assigned to a project based on their preferences and availability of resources. Students must necessarily work in pairs and may suggest projects not on the curated list, as long as they are able to find a mentor in the community. Once the project is under way, the mentor specifies the approved development process, coding standards, etc., then gets the students ramped up on the project and assigns the tasks that need to be completed. Over the course of the semester, the mentor continues to support the students by conducting weekly check-in meetings, reviewing code contributions, and helping to coordinate work through issue tracking systems.

In addition to contributing to a FOSS project, students learn about the cultural, technical, and business aspects of open source software, with a particular focus on social issues, such as: what are the moral/ethical foundations of FOSS? what motivates people to contribute to FOSS projects? how can a FOSS community be more inclusive, and what happens when it is not inclusive? does FOSS live up to its promises, or can only those who are already at a socioeconomic advantage actually participate in and benefit from it?

Introspection and reflection are important parts of the course. Students are expected to make multiple posts to their public blog each week and respond to the things they are learning and experiencing as they contribute to their project and get a better understanding of FOSS.

C. Meeting Educational Objectives

To conclude this section, we describe how the course meets its objectives of addressing the challenges identified in Section II: stereotype threat, impostor syndrome, lacking a sense of belonging, and misconceptions about the field.

1) *Engagement and Inclusion.* A key pedagogical element of the course is that there are no lectures, as they may be biased against students from underrepresented groups [15]. Rather, class meetings are focused on discussion and interactive, participatory, active learning exercises in which students participate and engage with instructors and peers to work out problems and explore concepts. Most variations of active learning reduce procrastination and promote better study habits by requiring students to come to class prepared to participate in mandatory group activities [16], and the

performance gap between non-URM and URM students, such as those from low socioeconomic backgrounds, is typically closed in such courses [17]. Likewise, there is a correlation in industry that shows that the majority of URMs have a lower level of soft skills when compared to their non-URM peers [18]. Active learning increases students’ soft skills due to its mandatory group work [19]. In addition, students rotate leadership roles during the activities [21], which helps avoid negative effects of stereotype threat and imposter syndrome [20].

2) *Sense of Belonging.* The class is designed to be smaller than the large-scale CS courses that typically have 150+ students at our institution. Limiting enrollment to around 20 helps students feel closer to the peers in this course and get a chance to know them better. The small group discussions and activities ensure the students interact with one another. Moreover, every student gets a chance to integrate themselves into their FOSS project’s community. These activities give students an academic, social, and professional community.

3) *Finding a Role Model or Mentor.* A key difference between this course and traditional CS courses is the opportunity to find a role model or mentor. Instead of just choosing and implementing one large new feature for the project and submitting a pull request at the end of the semester, students are expected to be fully integrated with the community to explore what possible contributions they can make. For a FOSS project, the project’s community is an important factor that influences its growth and development. Thus, every student is expected to regularly communicate with a mentor and the overall FOSS community. They will be exposed to more people and the mentor can help them find successful professional role models.

4) *Increased Confidence.* Increasing confidence can negate the negative effects of stereotype threat and imposter syndrome. Previous graduate courses surveyed their students after they contributed to FOSS projects for a semester and there was significant increase in the students’ confidence in their technical skills [22]. In addition, students’ confidence can increase when they have the chance to provide their own input and ideas, help their peers, and receive validation for their work. The discussions and activities allow all students to share their views and feel valued while the projects will, ideally, accept and merge the students’ contributions. These small steps show them that they are closer to being recognized as a “real world” programmer.

5) *Real World Experience and Impact.* Real world experience can counter the misconception people have about CS. The course project provides an opportunity for students to experience contributing to a project with a team of professionals. In addition, there are guest speakers who speak about their expertise and how it helped their successful FOSS projects. Moreover, topics in the syllabus cover not only technical knowledge, but also essential soft skills and legal knowledge that developers need. Last, students who contribute to Humanitarian FOSS projects understand the potential for social good that can come from the field of CS.

¹ The course website is available at <http://www.seas.upenn.edu/~cdmurphy/foss> and a more complete syllabus can be found at http://foss2serve.org/index.php/FOSS_Course,_UPenn,_Murphy

IV. RESULTS

To date, 54 undergraduate students have completed the course across five offerings (the first author was a student in the course during its most recent offering; the second author is the course instructor) and have made contributions to 33 different FOSS projects, including MongoDB, Mozilla Firefox and Servo, and OpenMRS.

Although the number of students is small, the diversity numbers are encouraging: 43% of the students who took the course are women, 4% are African-American, and 7% self-identify as members of the LGBT community. All of these numbers are higher than the percentages in our department overall. Among members of those URM groups, the end-of-semester feedback was very positive, demonstrating that the course was meeting its goals:

- “I gained confidence in talking to people who I have never met in person.”
- “I feel more confident about contributing to more open source projects in the future.”
- “This course satisfied my confidence that I am indeed a programmer and computer scientist.”
- “The process of working with a large code base, navigating it and adding to it, and having an active dialog with people who spend much more time working with it have immeasurably boosted my confidence and informed my ability as to how exactly I as a computer scientist am contributing not just to existing projects, but the tech world at large.”
- “Open source gave us the group of friends we never had.”

V. CONCLUSION

Although stereotype threat, impostor syndrome, and misconceptions about the field of computing can have a negative effect on diversity, our experience in teaching a course on open source software development has shown that it can increase representation and improve URM students' confidence.

As part of future work, we intend to do a longitudinal study to determine the long-term effects of the course on the students who take it, and to measure its overall effects on representation within our department. Additionally, since the course is currently targeted at upper-level undergraduates, future work could evaluate whether such a course targeted at introductory-level students would also be effective, thus increasing the number of URM students who enter the field of computing.

ACKNOWLEDGMENT

We would like to thank Heidi Ellis, Gregory Hislop, Stoney Jackson, Darci Burdge, Lori Postner, and other members of the foss2serve team for their encouragement and for allowing us to participate in the POSSE series. We would

also like to thank Jay Borenstein and the Facebook Open Academy program for starting us in this direction, and all the FOSS project mentors who supported our students.

REFERENCES

- [1] "Data Tables", *Women, Minorities, and Persons with Disabilities in Science and Engineering*, 2016.
- [2] S. Cheryan, A. Master and A. Meltzoff, "Cultural stereotypes as gatekeepers: increasing girls' interest in computer science and engineering by diversifying stereotypes", *Frontiers in Psychology*, vol. 6, 2015.
- [3] M. Smith, "Perceptions of females in undergraduate computer science/technology programs toward the field of computer science: A mixed methods study", Ph.D, Saint Louis University, 2012.
- [4] N. Dasgupta, "Ingroup Experts and Peers as Social Vaccines Who Inoculate the Self-Concept: The Stereotype Inoculation Model", *Psychological Inquiry*, vol. 22, no. 4, pp. 231-246, 2011.
- [5] D. Simmons, *How Students of Color Confront Imposter Syndrome*. 2017.
- [6] A. Kumar, "The effect of soliciting demographic data on the performance of students on online tests", *2010 IEEE Frontiers in Education Conference (FIE)*, 2010.
- [7] M. Roberts, T. McGill and P. Hyland, "Attrition from Australian ICT degrees: why women leave", *ACE '12 Proceedings of the Fourteenth Australasian Computing Education*, vol. 123, pp. 15-24, 2012.
- [8] P. Clance and S. Imes, "The imposter phenomenon in high achieving women: Dynamics and therapeutic intervention.", *Psychotherapy: Theory, Research & Practice*, vol. 15, no. 3, pp. 241-247, 1978.
- [9] C. Steele, "A threat in the air: How stereotypes shape intellectual identity and performance.", *American Psychologist*, vol. 52, no. 6, pp. 613-629, 1997.
- [10] L. Irani, "Understanding gender and confidence in CS course culture", *ACM SIGCSE Bulletin*, vol. 36, no. 1, p. 195, 2004.
- [11] C. Steele and J. Aronson, "Stereotype threat and the intellectual test performance of African Americans.", *Journal of Personality and Social Psychology*, vol. 69, no. 5, pp. 797-811, 1995.
- [12] I. Miliszewska, G. Barker, F. Henderson and E. Sztendur, "The Issue of Gender Equity in Computer Science--What Students Say", *Journal of Information Technology Education*, vol. 5, pp. p107-120, 2006.
- [13] NY Times, "Missing From Science Class", 2017.
- [14] E. Greenwald, "5 Myths About Learning to Code You'd Be Better Off Ignoring", *The Muse*, 2017.
- [15] A. Ramirez, "Is Lecturing Culturally Biased?", *Edutopia*, 2017.
- [16] M. Prince, "Does Active Learning Work? A Review of the Research", *Journal of Engineering Education*, vol. 93, no. 3, pp. 223-231, 2004.
- [17] K. Spurr, "Active learning in large science classes benefits black and first-generation college students most", *University of North Carolina at Chapel Hill, College of Arts & Sciences*, 2014.
- [18] J. Snyder, J. Sloane, R. Dunk and J. Wiles, "Peer-Led Team Learning Helps Minority Students Succeed", *PLOS Biology*, vol. 14, no. 3, p. e1002398, 2016.
- [19] S. Berenson, K. Slaten, L. Williams and C. Ho, "Voices of women in a software engineering course: reflections on collaboration", *Journal on Educational Resources in Computing*, vol. 4, no. 1, p. 3-es, 2004.
- [20] P. Moss and C. Tilly, ""Soft" Skills and Race:: An Investigation of Black Men's Employment Problems", *Work and Occupations*, vol. 23, no. 3, pp. 252-276, 1996.
- [21] H. Phillips and R. Powers, "The College Seminar: Participation Under Instructor-Led And Student-Led Discussion Groups", *Teaching of Psychology*, vol. 6, no. 2, pp. 67-70, 1979.
- [22] R. Marmorstein, "Open source contribution as an effective software engineering class project", *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education - ITiCSE '11*, 2011.