Adopting a New Practice: Open Source Experiences in the Classroom

Sebastian Dziallas
University of the Pacific
Stockton, CA, USA
sdziallas@pacific.edu

Sally Fincher
University of Kent
Canterbury, Kent, UK
s.a.fincher@kent.ac.uk

Heidi J. C. Ellis
Western New England University
Springfield, MA, USA
heidi.ellis@wne.edu

Lori Postner
Nassau Community College
Garden City, NY, USA
lori.postner@ncc.edu

Gregory W. Hislop
Drexel University
Philadelphia, PA, USA
hislop@drexel.edu

***Abstract*—This Research Full Paper describes a case study of educators changing practice as a result of interactions with an open source software community. It examines an NSF-funded initiative designed to expose educators to humanitarian free and open source software (HFOSS) communities and, in turn, to support them in involving their students in these communities as part of their classroom experience. To date, more than 150 faculty from over 120 different institutions have participated in the initiative's faculty development workshops. In this work, we conducted in-depth interviews with 24 workshop participants. The interviews explored how faculty had adopted HFOSS in the classroom, the hurdles and successes they encountered, and how their teaching had changed, among other questions. Some of the themes we identify in our data – obstacles to adoption, such as a lack of time, and the importance of the institutional context – confirm prior findings in the literature on pedagogical change. However, this work also identifies several additional nuances that have not previously been reported and emphasizes common aspects among educators who successfully adopted a practice.**

***Index Terms*—open source software, qualitative methods, pedagogical change**

I. Introduction

This paper examines an NSF-funded initiative designed to expose educators to humanitarian free and open source software (HFOSS) communities and, in turn, to support them in involving their students in these communities as part of their classroom experience. This is not a trivial undertaking. In addition to the challenges ordinarily associated with pedagogical change, involving students in open source communities requires instructors to understand a sizeable real-world project and to be comfortable with the uncertainty of depending on external parties in the classroom [1], [2].

The focus of this work is the Professors' Open Source Software Experience (POSSE) [2], an NSF-funded faculty development initiative. The POSSE workshop was originally developed in 2009 and became part of an NSF-funded project in 2013. It consists of three stages [1], [2]:

- online activities that emphasize factual content about open source communities,
- a 2.5 day face-to-face workshop that discusses open source culture and exposes participants to open source communities,
- academic and FOSS mentoring after the workshop.

Since the beginning, more than 150 faculty from over 120 different institutions have participated in the workshops.

The study presented in this paper is part of an effort designed to explore instructors' adoption of open source practices in the classroom. A previous study reported on the first phase of this work which used survey responses to investigate challenges and opportunities of adopting HFOSS in the classroom [3]. It reported that many participants had successfully adopted HFOSS in their classrooms, but also identified common challenges, such as finding the time to cover relevant material in an existing course. The second phase, which this paper focuses on, draws on semi-structured interviews to provide a more in-depth perspective. Specifically, we wanted to:

- gain a deeper understanding of the hurdles and obstacles instructors faced in adopting open source experiences in their classrooms,
- explore in what ways adopting open source experiences affected instructors' practice.

II. Related Work

Relevant prior work in the literature includes efforts to incorporate open source communities into the classroom experience, as well as studies designed to explore professional development in computing education more broadly.

A. Open Source Communities

Open source software projects have successfully been used in a variety of ways at different institutions, including in software engineering courses and as part of capstone projects

This material is based on work supported by the National Science Foundation under Grants DUE- 1225738, 1225688, 1225708, 2012966, 2013069, 2012979, 2012999, and 2012990. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

[4]–[6]. More recently, Choi et al. have reported on different models of incorporating open source projects into CS2 courses [7].

Software that somehow benefits the human condition, Humanitarian Free and Open Source Software (HFOSS), has been shown to be attractive to underrepresented groups in computing while supporting real-world learning. Early research into student participation in HFOSS has demonstrated that students gain practical experience with a professional project via such participation [8]–[10]. A multi-institutional study by Hislop et al. reports that "students perceived that participating in an HFOSS project made them more comfortable with computing and improved their perceived ability to maintain a project and interact with professionals." [11] Ellis et al. found that "[...] student involvement in an HFOSS project has a positive impact on perceived learning of software engineering knowledge." [12].

Prior research has also shown that HFOSS domains are attractive to underrepresented groups [13]–[15]. Results indicate that HFOSS has broad appeal to women and can bolster enthusiasm for studying computing. A study of the participants in the Open Source Day at the Grace Hopper Celebration of Women in Computing found that students were motivated by the ability to "do good" as well as the opportunity to learn new technologies and skills [14].

As a result of observing the benefits of student learning within an HFOSS project, several academic institutions have developed local HFOSS applications. These projects fill a local community need via a professional application and associated community while also providing a rich learning environment for students [4], [16]. Projects range from campus food pantries [16], to support for campus organic farming [4], to a dashboard to track campus building energy use.

B. *Professional Development*

Prior work in professional development has often focused on teachers, in part due to recent efforts to bring computing into K12 classrooms in countries around the world [17], [18]. However, there are also several examples in higher education. For instance, Fincher et al. collected "change stories" in which educators describe a time they changed their teaching practice. They found that: "Of the 99 change stories analyzed, only three demonstrate an active search for new practices or materials on the part of teachers, and published materials were consulted in just eight of the stories. Most of the changes occurred locally, without input from outside sources, or involved only personal interaction with other educators." [19]

In the first phase of another study, Barker et al. drew on interviews and observations with 66 faculty to examine how faculty find out about new teaching practices, why they decide to try them, and why they keep using (some of) them [20]. They report that faculty typically do not seek out findings from educational research and instead find educational innovations either to address a particular problem they observe or because they hear about them at conferences or from colleagues [20]. In the second stage of this work, Hovey et al. conducted a survey of 821 CS faculty at 595 institutions. They report that: "faculty who tried an innovation were motivated primarily by concerns for students' learning and course experience, including their engagement and participation." [21]

There are also reports related to the adoption of individual practices in the literature, similar to the one we are examining in this work. For instance, Hu et al. conducted a survey with 32 faculty respondents who had attended process-oriented guided inquiry learning (POGIL) workshops or were using POGIL in their classrooms [22]. Through mainly quantitative analysis, they identify familiar obstacles to adoption, such as a lack of time and relevant materials.

Finally, in 2018, an ITiCSE working group conducted a comprehensive review of the adoption of educational innovations in computer science [23]. In their working group report, Taylor et al. observe: "There appears to have been relatively less research on the propagation of educational innovations in computer science compared to other STEM disciplines, but overall, findings seem to be similar." [23] This work then contributes an additional perspective grounded in the context of computing education. The patterns we observe in our data are broadly similar to those reported in the literature. However, this work also identifies several additional nuances related to the role of the instructor that have not previously been reported and emphasizes common aspects among educators who successfully adopted a practice.

III. Methods

For the first phase of this work, previously reported in [3], we initially invited 39 workshop attendees to complete a survey. We used purposive sampling and selected these participants to ensure that they came from different institutions and had a range of experiences in incorporating open source experiences into their classrooms. Of the 39 participants who were contacted, 26 chose to complete the survey. (See [3] for more information, including details about the adoption of open source in the classroom among participants.) Subsequently, for the second phase of this work, which this paper focuses on, we invited them to participate in follow-up interviews for this study and 24 (11 women and 13 men) chose to do so. Respondents from a wide range of institutions across the United States participated in the interviews. Two came from community colleges, six from liberal arts colleges, six from private universities, and 11 from public universities.

The interviews lasted from 30 to 60 minutes and were conducted in-person, where possible, or by phone. They included questions about how faculty had adopted HFOSS in the classroom, the biggest hurdles and successes they had encountered, what they would do similarly and differently in the future, as well as questions about the contributions their students had made to the HFOSS community, what they had learned from interacting with the community, and how their teaching had changed, among others.

The interviews were recorded, transcribed, and imported into the qualitative data analysis package NVivo. We obtained ethics approval for both phases of this work and use pseudonyms throughout this paper. We performed a thematic analysis, with one of the authors repeatedly reading through the data to identify themes. While these themes were related to our area of interest – instructors' adoption of open source practices in the classroom – they were not pre-defined and we remained open for new themes to emerge as well. These themes were then presented to, and discussed with, the other authors to clarify the thematic categories and to resolve any disagreements.

IV. Findings

The themes we identified can broadly be categorized into two groups: Those that are generic to any pedagogic change (which confirm prior findings in the literature) and those that are more intimately related to incorporating open source communities into the classroom experience. The former category includes problems and challenges educators faced in implementing open source experiences in their classrooms, as well as the institutional context and pressures. The latter category includes themes such as the importance of mentors and external support, acknowledging uncertainty, comfort with ambiguity and risks, and giving up authority. We discuss each of these categories (and their themes) in detail below.

A. Obstacles to Adoption

1) Work & Worry: Time constraints are often cited as a reason for the non-adoption of pedagogical practices [24], [25]. We saw a similar pattern in our data, as we had also observed in the first phase of this work [3].

> "I was hoping to find time to create projects for students to help them get more involved, but I haven't had time to do that." (Coburn)

Participants were also concerned about the effect of a new practice on their teaching evaluations. As Taylor et al. observe: "Instructors are highly sensitive to the fact that even a successful innovation can result in lower course evaluations for the first term it is used, due to implementation difficulties [26]." [23]

> "For a class, it's risky. It's time consuming and if you do something that doesn't work it is going to reflect real badly. [...] Students are really picky. If not everybody loved it you're in trouble. And they feed off each other." (Norwood)

This was particularly the case for instructors who were about to or had just applied for tenure.

> "I need to know that it is going to work. I applied for tenure this year, so I wasn't willing to take risks before this. I might be more willing to take risks next year." (Norwood)

Adopting a new practice then not only requires faculty to take the time to make it work, but also to overcome their own concerns about how it would be received [27]. However, not all participants were able to overcome their concerns.

> "[I] didn't feel comfortable having them do an open source project because I didn't feel comfortable enough to help them with it." (Parish)

2) Fit & Visibility: We also saw the influence of the local institutional context and associated pressures in our interviews. Faculty were keenly aware of what "worked" in their context.

> "Here is the thing with my students: if it's something that they have to do outside of classwork, I can't expect them to get it done. My students work full-time at various jobs just to make ends meet, so their time is very limited." (Coburn)

This was not only the case in relation to the student population, but also in terms of how the department works:

> "Any time we change, especially CS1 and CS2, it really is a joint decision. So even though I do the teaching for CS 1, I would not want to introduce git unless the department didn't object." (Clarkson)

Indeed, the importance of departmental culture as a factor in the adoption of new practice is currently underreported in the literature. As Taylor et al. observe: "A narrow focus on prescribed changes for individual faculty may miss key points of friction, such as how departmental culture, and the environment in which faculty work, can prevent highly motivated faculty members from adopting new innovations." [23]

In the case of the particular instructor above, attending the POSSE workshop appears to have encouraged them to introduce version control into their CS1 course.

> "Introducing git in our CS1 class is not a direct result of POSSE, but that certainly spurred me to talk to my colleagues who were open to it. I don't know how other departments do it, but we probably do way more collaboration than we need to." (Clarkson)

For some faculty, being the first to propose and implement a new practice in their department was a challenge:

> "I definitely have not sold my colleagues on open source software, so whatever I do I would be the pioneer. And I think the way I would persuade them is by example. [...] I don't think I'm going to be able to persuade them until I'm doing it myself. Because it is risky. And it is a big investment." (Newton)

Convincing other faculty in the department to support a change in practice then requires a certain level of "visibility" — being seen as adopting the practice oneself.

Others reported a lack of institutional support, both in terms of technical infrastructure and the ability to make curricular changes.

> "[...] our IT support is not geared towards software developers. We're having trouble even having a Linux presence. [...] The college-wide IT is much more concerned about security and security is the enemy of downloading and installing things." (Ferraro)

> "Lack of institutional support in the sense that I know that if I were to say to the chair or curriculum committee that I wanted to modify the course to

> insert the stuff into it, I would get push-back. I did get push-back so I decided I would silently slip it in.” (Parks)

In this, we can see an example of how instructors navigate their institutional context in order to adopt new practices. Indeed, as Fincher et al. observe: “Practitioners do not exist in isolation nor work in identical circumstances. For transfer [of practices] to occur, not only must you want to change your practice but you must also be able to do so. Your context has to permit you to change.” [28]

B. Open Source Context

1) Drawing Strength & Courage: We found that faculty drew strength and courage to adopt new practices from external sources. One example of this was through the community of workshop attendees. While not unique to this initiative, it is uncommon in pedagogic change efforts [29].

> “POSSE is a really big piece of this. If I was on my own, I probably would have given up. Its just too hard to invent all this stuff out of nowhere, out of a void. So having the POSSE group and structure has really helped me [...].” (Wegner)

Many participants also expressed the need for external support from the open source community, for instance through mentors.

> “One of the things that has helped is having a person that students could contact. [. . .] Somebody to who you can just say “I’m just stuck. Maybe I should know what I’m doing but I don’t. Could you help?”” (Clarkson)

This included support both with more technical challenges and to involve students in the open source community itself:

> “I needed more help on the application, not how to put it in the course. If someone had just told me that the build wouldn’t work on Windows 10 that would have been a huge help. It took me a huge amount of time to arrive at the conclusion that it wouldn’t work.” (Ferraro)

> “Where I struggle is the piece where I say “OK, now go get involved in the community.” That is always my struggle. Having a mentor with the community has helped mitigate that.” (Wegner)

Efforts to involve professionals and community members in classrooms have previously been reported in the literature. For instance, as part of the *Industry Fellows* project, a faculty member and professional work closely together on the delivery of a course [30]. However, the participants in our work also highlighted problems of scale with adopting such models at a larger number of institutions.

> “The problem with that becomes scalability. I have [name of the mentor] from Mozilla and that works really well if I have 20 students. But if there are 65 universities using Mozilla, I don’t think that [they] can do all of them. For me, what works really well I don’t think is scalable.” (Wegner)

Nevertheless, the findings in this section emphasize the importance of external support when adopting a new practice, such as from the community of workshop attendees or external mentors.

2) Acknowledging Uncertainty: Some of the faculty who involved their students in open source communities reported that it was a learning experience not only for the students, but also for the instructors themselves.

> “One thing — embarrassing to admit, I learned a lot about workflow, particularly in open source projects, because I’ve never done professional software development. The GitHub workflow was new to me, and that was a skill we should be teaching our students.” (Dunn)

> “Sitting in on design meetings, seeing how they organize a sprint, seeing what they did during a sprint planning meeting, seeing what they did at the end of a sprint. This was the first time I had actually experienced these things on a real project. It makes you realize what you were doing wrong when you were trying to translate what you have read or watched in videos into the classroom.” (Forrest)

This is an example of instructors identifying relevant practices for their courses in a way similar to that described by Barker et al. and Fincher et al. [19], [20]. These practices are not exclusively about technical content knowledge, but about real-world practices that are used in professional software development. Exposure to these practices also affected how projects were run in the classroom:

> “Seeing how that [open source] project was run has fundamentally shifted how I run a project. [. . .] [It] forced me into learning those technologies. I have gotten better at being willing to jump into a new technology and tinker away. Yes, more confident, that comes from a competence. Almost an information literacy skill, because I know where to look for help.” (Forrest)

These participants acknowledge an uncomfortable experience (that is “embarrassing to admit” and “makes you realize what you were doing wrong”), but appear to be glad to have had it, as they recognize the importance of these practices.

3) Comfort With Ambiguity & Risks: Not all participants were comfortable with the ambiguity involved in introducing their students to open source projects.

> “I typically do course tutorials or other things to help students anticipate problems and reduce frustration [. . .] and I need to be really familiar with the material in order to feel sufficiently like an expert or a resource for the students — [the] real world is messy.” (Leslie)

Fincher and Dziallas identified a pedagogical stance as a form of embodied knowledge that is grounded in prior experience and informs instructors’ practices in the classroom [31]. It may also provide an explanation for responses to new practices, including “deal-breaking behavior” (a rejection of a practice as incompatible with one’s approach or context).

However, other participants described an increased willingness to take risks in the classroom.

> “I’ve gotten more comfortable with things not going as planned; better at trying something, taking a risk, making sure the students would learn something but not necessarily what I’d planned.” (Lawson)

> “I take more risks! This whole thing is just a gamble. Are they going to be able to do anything? And even though they weren’t able to do a contribution, didn’t get to do a build, I really feel like they learned a lot with trying and failing and trying and failing again. And so I have become a lot more tolerant of giving students things I’m pretty sure they will fail on several times.” (Ferraro)

A related aspect that is common in project-based learning more broadly is the idea of letting go of the end point of a project or class.

> “I think that when I started larger open source projects in the testing course 4-5 years ago, you’re letting go of setting the end point for students. I think that’s changed me quite a bit in that I’m not exactly sure where I’m going, I’m not exactly sure what I expect from you, but I hope you trust me as an instructor that it will be fine and I hope I can trust you as a student that you’re making progress. It’s not a controlled project that I give them.” (Mercer)

This shift in pedagogic practices could be characterized as moving from “sage on the stage” to “guide on the side” [32].

4) Giving Up Authority: Related to this shift towards being a ”guide on the side” in the classroom is the idea of giving up authority in the classroom. Several instructors also spoke about this in the process of introducing open source projects into their classrooms. One even went as far as to keep themselves from “knowing too much”.

> “[I’m] more willing to cede authority and admit that I don’t know something” (Marshall)

> “This was not a regular course where I told them stuff. In a way I kept myself from knowing too much because I didn’t want to say “no this is how you do it”. So they were trying different things, pulling things off the internet, asking [one of the mentors] things.” (Ferraro)

V. Discussion

A. *Categorization of Obstacles*

In the first part of the findings section, we have seen a number of pitfalls that instructors experience when adopting a new practice. In the following, we provide a brief categorization of such obstacles. These obstacles can be experienced at personal, curricular, and institutional levels.

At a personal level, an instructor may choose not to implement open source practices in their classroom because of concerns about initial implementation difficulties, which may affect the student response in teaching evaluations or an upcoming tenure review, or their own pedagogic stance. As one participant noted: “I need to be really familiar with the material in order to feel sufficiently like an expert or a resource for the students [. . .].”

At a curricular level, they may not have encountered the kinds of tools and techniques commonly used in open source projects. “This was the first time I had actually experienced these things on a real project. It makes you realize what you were doing wrong [. . .].” Exposure to the open source community then provided an opportunity for them to identify relevant industry practices.

Finally, at an institutional level, adopting these practices may require discussion with other faculty in the department, who may not be inclined to support such a change: “I would not want to introduce git unless the department didn’t object.” For some of the instructors, this meant they had to take the initiative themselves: “And I think the way I would persuade them is by example.”

At all of these levels, there is then a certain amount of risk-taking required to implement new practices. While the obstacles we identified here may apply to pedagogic change efforts more broadly, they are also specific to this practice. Many disciplines, including law, medicine, and (at some institutions) engineering, require students to gain supervised workplace experience [33]. However, rather than requiring work-based experiences for a fixed amount of time (e.g. one semester or academic year), the work discussed here allows instructors to embed these practices in their classes, as early as in introductory algorithms and data structures courses [7].

B. *Open Source Culture*

In the second part of the findings section, we have seen aspects specific to the open source context. One important factor here is that student involvement in an open source project within the structure of a classroom requires the melding of two different cultures. While there are some similarities between academic and open source cultures such as freedom of expression [34] and the idea that knowledge is to be shared for the greater good of the community [35], there are considerable differences between the two cultures. Open source communities have short timelines with products being released as often as every few months. Academia has much longer release times with curricular change typically taking at least a year to effect, and often much longer. As one instructor observed:

> “I said I was thinking of creating a sequence of courses that would lead to participating in HFOSS projects and I asked what people thought of a minor. Most of the people thought it was a really good idea. The chair and curriculum committee representative were less enthused due to the paperwork involved.” (Parks)

Open source projects are also opportunistic in that development utilizes resources as they appear which results in a somewhat less predictable path to a final product. Academia sets schedules and learning objectives a year or more in advance with little opportunity for modification. As a result, open source projects are positioned to have flexible and changing project goals such that schedules and deliverables may not align well with academic course outcomes as a term progresses.

Open source culture defaults to open with ideas being freely shared within the community from initial conception through maturation. While academic culture supports the generation of new ideas, typically these new ideas are explored and polished before being presented to the community in the form of a publication. An additional barrier to sharing within academia is that students are discouraged from sharing work products due to a concern about cheating except in specific, limited situations.

We have also begun to see how some instructors overcame this mismatch of cultures through external support, such as from other workshop attendees and mentors in the open source community.

C. Limitations

The work presented here is necessarily limited to its context: participants attending a specific faculty development initiative about introducing open source experiences into their classrooms. Our goal was then not to provide generalizable findings, but to explore participants' experiences in depth. We hope that the detailed description of this work's context and the connections to prior findings in the literature will allow readers to transfer the findings to their own contexts.

An additional limitation is that this work relied on interviews with 24 instructors who were selected using purposive sampling. However, this number of participants is in line with recommendations in the literature [36].

Finally, quality in qualitative research is commonly assessed through credibility and trustworthiness [37]. To address this, we have provided a detailed account of how the data was collected and analyzed. The author conducting the analysis was also not involved in conducting the workshops.

VI. Conclusions

Henderson et al. observe that: “Two commonly used change strategies are clearly not effective: developing and testing ‘best practice’ curricular materials and then making these materials available to other faculty, and ‘top-down’ policy-making meant to influence instructional practices.” [24] The work presented here takes a different approach and exposes instructors to the open source community.

We have identified specific aspects – building a community of participants [38]–[40] and engaging external mentors – that support instructors in adopting this new practice. These aspects then serve as recommendations for others planning to engage in pedagogic change efforts.

We were also interested in gaining a deeper understanding of the hurdles and obstacles instructors faced in adopting open source experiences in their classrooms. In this regard, we have confirmed prior findings in the literature and contributed a categorization of obstacles at a personal, curricular, and institutional level specific to this practice.

Finally, we identified additional themes – drawing strength and courage, acknowledging uncertainty, comfort with ambiguity and risks, and giving up authority – that illustrate instructors' responses to being exposed to the open source community (whether as part of the POSSE workshop or, subsequently, in the classroom).

We believe that some of the changes participants describe came as a result of being involved in a large, “messy” open source project, with a culture that differs substantially from academic culture. Their responses to this may be related to the concept of instructor identity. Exploring in more detail how involvement in such projects impacts instructor identity would then provide a promising opportunity for future work.

References

[1] H. J. C. Ellis, G. W. Hislop, M. Purcell, M. Chua, and S. Dziallas, “Towards a model of faculty development for FOSS in education,” in *2013 IEEE 26th Conference on Software Engineering Education and Training (CSEE&T)*, May 2013, pp. 269–273.

[2] B. Morgan, G. W. Hislop, and H. J. C. Ellis, “Faculty Development for FLOSS Education,” in *Open Source Systems*, ser. IFIP Advances in Information and Communication Technology, F. Bordeleau, A. Sillitti, P. Meirelles, and V. Lenarduzzi, Eds. Cham: Springer International Publishing, 2019, pp. 165–171.

[3] L. Postner, H. J. Ellis, and G. W. Hislop, “A Survey of Instructors' Experiences Supporting StudentLearning using HFOSS Projects,” in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, Feb. 2018, pp. 203–208.

[4] G. Braught, J. Maccormick, J. Bowring, Q. Burke, B. Cutler, D. Goldschmidt, M. Krishnamoorthy, W. Turner, S. Huss-Lederman, B. Mackellar, and A. Tucker, “A Multi-Institutional Perspective on H/FOSS Projects in the Computing Curriculum,” *ACM Transactions on Computing Education*, vol. 18, no. 2, pp. 7:1–7:31, Jul. 2018.

[5] T. M. Smith, R. McCartney, S. S. Gokhale, and L. C. Kaczmarczyk, “Selecting open source software projects to teach software engineering,” in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '14. New York, NY, USA: Association for Computing Machinery, Mar. 2014, pp. 397–402.

[6] D. Spinellis, “Why computing students should contribute to open source software projects,” *Communications of the ACM*, vol. 64, no. 7, pp. 36–38, Jun. 2021.

[7] E. Choi, L. Meng, and J. Hott, “Open Source Software Practices in CS2,” in *21st Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 1–5.

[8] G. W. Hislop, H. J. C. Ellis, and H. Jackson, “Student contribution to HFOSS: Challenges and opportunities,” *Journal of Computing Sciences in Colleges*, vol. 33, no. 6, pp. 181–182, Jun. 2018.

[9] H. J. C. Ellis, G. W. Hislop, S. Jackson, and L. Postner, “Team Project Experiences in Humanitarian Free and Open Source Software (HFOSS),” *ACM Transactions on Computing Education*, vol. 15, no. 4, pp. 18:1–18:23, Dec. 2015.

[10] G. W. Hislop, H. J. C. Ellis, and B. Morgan, “Student reflections on learning in HFOSS,” *Journal of Computing Sciences in Colleges*, vol. 35, no. 8, pp. 288–289, Apr. 2020.

[11] G. W. Hislop, H. J. C. Ellis, S. M. Pulimood, B. Morgan, S. Mello-Stark, B. Coleman, and C. Macdonell, “A Multi-Institutional Study of Learning via Student Involvement in Humanitarian Free and Open Source Software Projects,” in *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ser. ICER '15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 199–206.

[12] H. J. C. Ellis, G. W. Hislop, S. M. Pulimood, B. Morgan, and B. Coleman, "Software Engineering Learning in HFOSS: A Multi-Institutional Study," in *2015 ASEE Annual Conference & Exposition*, Jun. 2015, pp. 26.1379.1–26.1379.12.

[13] L. Postner, D. Burdge, S. Jackson, H. Ellis, G. Hislop, and S. Goggins, "Using humanitarian free and open source software (HFOSS) to introduce computing for the social good," *ACM SIGCAS Computers and Society*, vol. 45, no. 2, p. 35, Jul. 2015.

[14] C. Macdonell, H. J. C. Ellis, D. Burdge, L. Postner, and G. W. Hislop, "The Use of HFOSS Projects in the Grace Hopper Celebration of Women in Computing Open Source Day," in *2018 ASEE Annual Conference & Exposition*, Jun. 2018.

[15] H. J. C. Ellis, G. W. Hislop, and L. Postner, "The Power of Open Source for Social Good to Increase Diversity in Computing," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, Mar. 2021, p. 1305.

[16] K. R. Wurst, C. Radkowski, S. Jackson, H. J. C. Ellis, D. Burdge, and L. Postner, "LibreFoodPantry: Developing a Multi-Institutional, Faculty-Led, Humanitarian Free and Open Source Software Community," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, Feb. 2020, pp. 441–447.

[17] Q. Cutts, J. Robertson, P. Donaldson, and L. O'Donnell, "An evaluation of a professional learning network for computer science teachers," *Computer Science Education*, vol. 27, no. 1, pp. 30–53, Jan. 2017.

[18] K. Falkner, R. Vivian, N. Falkner, and S.-A. Williams, "Reflecting on Three Offerings of a Community-Centric MOOC for K-6 Computer Science Teachers," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. Seattle Washington USA: ACM, Mar. 2017, pp. 195–200.

[19] S. Fincher, B. Richards, J. Finlay, H. Sharp, and I. Falconer, "Stories of change: How educators change their practice," in *Frontiers in Education Conference (FIE), 2012*, Oct. 2012, pp. 1–6.

[20] L. Barker, C. L. Hovey, and J. Gruning, "What Influences CS Faculty to Adopt Teaching Practices?" in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. Kansas City Missouri USA: ACM, Feb. 2015, pp. 604–609.

[21] C. L. Hovey, L. Barker, and V. Nagy, "Survey Results on Why CS Faculty Adopt New Teaching Practices," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, Feb. 2019, pp. 483–489.

[22] H. H. Hu, C. Kussmaul, B. Knaeble, C. Mayfield, and A. Yadav, "Results from a Survey of Faculty Adoption of Process Oriented Guided Inquiry Learning (POGIL) in Computer Science," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. Arequipa Peru: ACM, Jul. 2016, pp. 186–191.

[23] C. Taylor, J. Spacco, D. P. Bunde, Z. Butler, H. Bort, C. L. Hovey, F. Maiorana, and T. Zeume, "Propagating the adoption of CS educational innovations," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. Larnaca Cyprus: ACM, Jul. 2018, pp. 217–235.

[24] C. Henderson, A. Beach, and N. Finkelstein, "Facilitating change in undergraduate STEM instructional practices: An analytic review of the literature," *Journal of Research in Science Teaching*, vol. 48, no. 8, pp. 952–984, Oct. 2011.

[25] P. Atjonen, E. Korkeakoski, and J. Mehtäläinen, "Key pedagogical principles and their major obstacles as perceived by comprehensive school teachers," *Teachers and Teaching*, vol. 17, no. 3, pp. 273–288, Jun. 2011.

[26] L. Barker and J. Gruning, "The student prompt: Student feedback and change in teaching practices in postsecondary computer science," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. Madrid, Spain: IEEE, Oct. 2014, pp. 1–8.

[27] D. M. Le Fevre, "Barriers to implementing pedagogical change: The role of teachers' perceptions of risk," *Teaching and Teacher Education*, vol. 38, pp. 56–64, Feb. 2014.

[28] S. Fincher, M. Petre, and M. Clark, *Computer Science Project Work: Principles and Pragmatics*. Springer-Verlag, 2001.

[29] B. J. Eib and P. Miller, "Faculty Development as Community Building," *International Review of Research in Open and Distributed Learning*, vol. 7, no. 2, pp. 1–15, 2006.

[30] J. Tenenberg, "Industry fellows: Bringing professional practice into the classroom," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '10. New York, NY, USA: Association for Computing Machinery, Mar. 2010, pp. 72–76.

[31] S. Fincher and S. Dziallas, "Then and Now: Past Experience Echoed in University Computing Teachers' Current Practice," in *Research in Engineering Education Symposium*, Dublin, Ireland, May 2015.

[32] M. Chua and S. Dziallas, "Work in progress: From sage on the stage to guide on the side: Examining shifts in teaching practice through stories of open community participation," in *Frontiers in Education Conference (FIE), 2012*, Oct. 2012, pp. 1–2.

[33] S. Fincher and D. Knox, "The Porous Classroom: Professional Practices in the Computing Curriculum," *Computer*, vol. 46, no. 9, pp. 44–51, Sep. 2013.

[34] W. Scacchi, "Free/open source software development," in *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*, ser. ESEC-FSE '07. New York, NY, USA: Association for Computing Machinery, Sep. 2007, pp. 459–468.

[35] J. Ljungberg, "Open source movements as a model for organising," *European Journal of Information Systems*, vol. 9, no. 4, pp. 208–216, Dec. 2000.

[36] G. Guest, A. Bunce, and L. Johnson, "How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability," *Field Methods*, vol. 18, no. 1, pp. 59–82, Feb. 2006.

[37] Y. S. Lincoln and E. G. Guba, *Naturalistic Inquiry*. SAGE, Apr. 1985.

[38] A. Furco and B. E. Moely, "Using Learning Communities to Build Faculty Support for Pedagogical Innovation: A Multi-Campus Study," *The Journal of Higher Education*, vol. 83, no. 1, pp. 128–153, Jan. 2012.

[39] T. L. Tinnell, P. A. S. Ralston, T. R. Tretter, and M. E. Mills, "Sustaining pedagogical change via faculty learning community," *International Journal of STEM Education*, vol. 6, no. 1, p. 26, Aug. 2019.

[40] M. Tovar, R. Jukier, J. Ferris, and K. Cardoso, "Overcoming Pedagogical Solitude: The Transformative Power of Discipline-Specific Faculty Learning Communities," *To Improve the Academy*, vol. 34, no. 1-2, pp. 319–344, 2015.