

Evaluating Student Experiences in Developing Software for Humanity

Gregory W. Hislop
Drexel University
College of Information Science and
Technology
Philadelphia, PA 19104-2875 USA
00-1-215-895-2179
hislop@drexel.edu

Heidi J. C. Ellis
Trinity College
Computer Science Department
Hartford, CT 06106 USA
00-1-860-297-4175
heidi.ellis@trincoll.edu

Ralph A. Morelli
Trinity College
Computer Science Department
Hartford, CT 06106 USA
00-1-860-297-2220
ralph.morelli@trincoll.edu

ABSTRACT

Student involvement in Humanitarian Free and Open Source Software (HFOSS) projects holds the potential to provide a rich education experience to undergraduates. This paper discusses educational use of HFOSS including identification of the potential for HFOSS to impact curricula, an overview of an implementation of HFOSS in education and a description of an evaluation framework that includes measures of success of such a program. The paper also presents results from initial surveys of students involved in HFOSS development in several U.S. academic institutions. The paper concludes with a discussion of work in progress.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education – *Computer Science Education*.

General Terms

Human Factors.

Keywords

Open source software, student motivation, computing education.

1. EDUCATIONAL POTENTIAL OF HFOSS

Humanitarian Free and Open-Source Software (HFOSS) is software that is built to serve humanitarian purposes and made available under a free and open source license. The purposes of this software can vary widely and examples range from disaster management applications to software used to manage volunteers for non-profit social service organizations.

HFOSS shares the technical characteristics common to FOSS projects in general. It has additional qualities and educational potential because of its humanitarian purposes. Such software can be developed by distributed groups of subject matter experts and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITiCSE '09, July 6–9, 2009, Paris, France.

Copyright 2009 ACM 978-1-60558-381-5/09/07...\$5.00.

software development professionals, and students can participate as legitimate contributors.

Experience with large scale software development is important to students' overall professional preparation. One of the primary objectives of academic computing programs, including computer science, software engineering, and information technology, is to provide students with the skills and knowledge required to step into a professional position in computing [1, 2, 3]. In addition, students need to go beyond the simple acquisition of skills and knowledge to learn how to apply these skills and knowledge in a real-world environment that includes a community of developers and users [4,5].

The CC2005 curriculum guidelines suggest several mechanisms for conveying technical skills and professionalism to students including capstone courses, project courses, co-ops, and courses on ethics and professionalism [2]. While any of these approaches can be highly successful, instructors often struggle with finding ways to strengthen these courses. The courses may suffer due to short duration of the projects, isolation from a professional environment, lack of participation by professional developers, or scarcity of potential client organizations accessible from campus.

The FOSS community provides an excellent option to give students broader experience. The open-source movement has been gaining momentum within the software industry [6] and has been suggested as one way to provide students with industry experience in an academic setting [7]. Indeed, open-source software provides a uniquely transparent and flexible development environment [8].

The humanitarian purpose of HFOSS projects can bring additional elements to the student experience. The hope is that HFOSS will give students a new understanding of the relevance of computing majors, and be attractive for students interested in majors that allow them to make a positive difference in society.

2. IMPLEMENTATION

Several years ago Trinity College began using Sahana [9], an open-source disaster management software project, as an implement for software engineering education. Sahana, Sinhalese for relief, is an HFOSS disaster management system. It is an international project to develop a web-based application that addresses the IT coordination problems that typically occur in trying to recover from a large-scale disaster. Undergraduate students have been working with the Sahana core team and volunteers from Accenture Corporation to develop production-

level software that has been incorporated into the most recent Sahana release.

The success of the initial experience with Sahana has led to several other projects at Trinity and a handful of other institutions. These projects have included student work on OpenMRS [10], as well as some efforts to start humanitarian FOSS projects to serve the needs of local social service organizations such as Literacy Volunteers.

The initial efforts were conducted without major funding support. Assistance was provided by Accenture, Partners in Health, and AidMatrix. These organizations provided some funding, suggested project ideas, helped define requirements, and reviewed documents. The student work has been conducted within the structure of courses and also using some summer internship programs.

More recent work has been organized as “The HFOSS Project” and has received some external funding. Other discussions of the work include [11-18] and more information is available at <http://hfoss.org>.

3. MEASURES OF SUCCESS FOR HFOSS

The impact of HFOSS projects can be considered from multiple perspectives. These perspectives can be organized into three broad themes, each of which is discussed below.

Attracting and engaging students: There is a pressing need in the US to attract and retain many more students in computing. The sharp decline in enrollment since about 2001 is well documented. This has been accompanied by reversal of gains that had been made in attracting women students and other traditionally under-represented groups [19].

There are many aspects of the use of HFOSS that have the potential to attract new students to computing as well to enhance the interest of existing students including:

Relevance – Conveying the relevance and practical application of computing seems to be important to attracting the current generation of students to the computing disciplines. Female students in particular appear to desire to understand the applicability of software as well as to create software that helps others [20].

Compassion - The nature of HFOSS, which focuses on helping people, can engage students who respond to the incentive of helping others in a visible and concrete manner.

Community - The HFOSS community is diverse, comprised of people from various countries, backgrounds and professions. HFOSS has a potential user base that spans the globe. Student participation in this community provides opportunities to expose students to developers and also humanitarian professionals and users from a variety of walks of life.

Enhancing student learning: Student FOSS projects provide learning opportunities in key technical areas including programming skills, testing and debugging techniques, and use of software development tools such as version control systems. However learning in these areas is not particularly unique to FOSS projects. The real advantage in FOSS projects relates to computing degree program objectives intended to prepare

students for professional positions. HFOSS is very attractive for the potential breadth of professional experience that students can gain. Key aspects of this experience that relate to software development include:

Software process – Students participating in an HFOSS project must conform to the development process used by a particular FOSS project. Students gain an understanding of the steps necessary to plan and carry out a software project in a real-world environment for a project with substantial impact.

Distributed development –The global complexion of the HFOSS community provides a unique opportunity for students to develop software in collaboration with professionals in a variety of locations and from a variety of cultures. This experience prepares students for increasingly distributed professional environments.

Project scale and complexity – HFOSS provides students with a project of significant size and corresponding complexity, allowing students to understand the impact of working with a project containing sizeable intricacy.

Projects with long product life – In order to better understand the development lifecycle, students need to be exposed to a project that is ongoing and has a longer life-span than a single-semester project. HFOSS projects can help students gain an understanding of issues such as maintenance and understand the impact of decisions made by developers earlier in the product’s lifespan.

FOSS – Beyond humanitarian FOSS, the open-source movement is gaining momentum and open source products are becoming widely used in a variety of domains. Students need to understand the role of OSS and its impact on business.

Creating social benefit: A core result of successful HFOSS in education is to make real contributions to the HFOSS products. This result fits the academic environment in several ways:

Student social awareness – A critical aspect of a student’s education is gaining an understanding of social responsibility. Educational institutions have long worked to help students prepare for lives as good citizens and useful members of the community and the world at large. In a manner similar to pro bono work in law, HFOSS allows computing students to employ their computing skills to help others.

Institutional outreach and social service – Most academic institutions include contributions to the community as part of their mission. The participation in and support of an HFOSS project allows the institution to fulfill this mission in a unique manner.

Leveraging academic resources – The involvement of multiple institutions collaborating on HFOSS projects could bring substantial resources to bear on social needs. In this manner, the academic community at large can have a positive impact at a global level.

In summary then, these three themes provide a broad perspective on what constitutes success for HFOSS educational projects. At the core, the goal is to use HFOSS to enable student learning. More generally, HFOSS may help attract students to the discipline. Finally, HFOSS provides a potentially interesting contribution to the social mission of academic institutions. These themes in turn inform the evaluation of the projects.

4. INITIAL EVALUATION

The themes discussed earlier as defining success for this effort are too broad to be covered by a single evaluation approach. Therefore, the evaluation of HFOSS efforts will consist of a variety of methods including focus groups, structured interviews, surveys, and evaluation of work products. Not all of these methods have been applied at this time, but several are discussed in the following sub-sections. Additional results are included in [13-15, 17, 18].

4.1 Structured Interview

A structured interview was constructed to elicit the opinion of instructors on the educational aspects of involving students in HFOSS projects within the environment of an academic course. Questions cover the technical background of students, the software development process used by students to develop HFOSS applications, and the educational infrastructure used to support that process.

At present, this interview has only been used for a single course offered during the fall 2008 term, so no general conclusions can be drawn. However, that one interview provided a variety of useful observations on the experience of teaching an HFOSS course. Some examples follow.

The instructor thought that students got real-world experience with development tools. "Students can see the tools in use and have a reason to use them as opposed to pure discussion."

This instructor found course materials created for teaching HFOSS courses useful. There was more material available than he could use effectively, due to schedule constraints. However he was able to apply parts of the materials to help provide direction and structure for the course. For example, he liked the grading rubrics "I learned a lot about how to tell students what was I was looking for. That was extremely helpful"

The instructor thought that a repository of education-related FAQs for various software development tools and teaching approaches would very useful to instructors and technical support people for both getting the development environment running as well as for teaching.

The interactions with the core developers on the HFOSS project team were very positive. "... I found that to be an extremely good collaboration. ... <the HFOSS representative> understands that we are teaching students. So while he does have a desire to have a particular functionality added to the software that he manages, he has a good grasp, not necessarily of the specific constraints, but that there are constraints that might not meet his requirements."

Overall, this instructor found the use of HFOSS to be worthwhile, and believes that it is worth continuing the effort.

4.2 Student Post-HFOSS Survey

A 44-statement survey was developed to determine student attitudes in three general areas: 1) the impact of involvement in an HFOSS project on their motivation and learning 2) opinion of learning software engineering knowledge and skills; and 3) impact of involvement in an HFOSS project on academic and professional career plans. The items in these areas will be useful in addressing the overall themes related to attraction and retention of students, and learning outcomes.

The survey used a five point Likert scale from strongly disagree (1) to strongly agree (5). There were also options for "not applicable" and "don't know". The survey also asked questions about student background including age, gender, major and programming ability.

In summer 2008, Trinity College (Hartford, CT USA) offered a 10-week internship program to undergraduate students interested in participating in the development of HFOSS. The internship received over 30 applications and selected 13 participants (10 male, 3 female) from Trinity College, Connecticut College, the University of Hartford, and the University of Connecticut. In the internship, students worked on various projects for five different HFOSS groups.

In fall 2008, a software development course was offered via video conferencing to 17 students at three locations; Trinity College (3), Connecticut College (7) and Wesleyan University (7). Students spent the first half of the course learning software development approaches and then applied what they had learned to develop a project based on the OpenMRS application in the second half of the course.

The survey was given in hard copy form to students during their last week of participating in the internship or course. The summer 2008 had a 100% response rate while there were 14/17 responses for the fall 2008 for an 82.4% response rate and an overall response rate of 27/30 = 90%. It should be noted that one subset of the fall 2008 students did not actually work on an HFOSS project and therefore provided no input into many of the survey items.

The students ranged from sophomores to seniors with 18 students between the ages of 18 to 20 and nine students between the ages of 21 to 23. Of the 27 students, 18 named computer science as their major, and 5 more mentioned CS as a second major.

Students also rated themselves on programming ability using a 1-5 scale with 1 being "Novice" and 5 being "Advanced". The majority of the students (14) rated themselves at a four out of five. Only one student rated themselves as a five and no student self-rated at level one.

The responses for each of the questions in the survey were tabulated and examined for their relation to the evaluation areas discussed above. Some of the interesting results follow.

Learning and Motivation - In the area of the impact of humanitarian FOSS on motivation and learning, the responses were quite strongly positive. In general, students seemed to value the experience of working on a project with humanitarian purpose. The related items all had a majority of responses falling in the "agree" and "strongly agree" categories. Several questions indicated strong support for the notion that students became more aware of the potential benefits of computing to society. For example, two of the statements and percent of students agreeing or strongly agreeing are:

- Working on an HFOSS project gives me a better appreciation for the usefulness of computing. (91%)
- I enjoyed working on an HFOSS project because the project can positively impact the world. (91%)

Similarly, students seemed to be motivated by the real world connection of their projects. They also seem to have had a

positive experience that will prepare them to deal with increasingly diverse, global workforce:

- Knowing that my project will help people motivates me to do my best on the HFOSS project. (86%)
- I enjoyed working on an HFOSS project because it allowed me to participate in a diverse community of HFOSS developers. (71%)

Software development learning – The impact on student learning in this area has not been measured directly yet. However, the student self-assessment of learning indicates a positive impact of participation in an HFOSS project on learning related to professionalism and software engineering. Students agree that they have developed some understanding and ability to use a software process. They also exhibit some confidence in being able to participate on a development team, and in knowing appropriate team behavior. The statements below are shown with the accompanying percent of students agreeing or strongly agreeing with each statement.

- I can list the steps in the software process we used in the HFOSS project. (83%)
- I can use a software process to develop an HFOSS project. (77%)
- I am sure that I can actively participate in an HFOSS community to develop a software project. (87%)
- I can participate in an HFOSS development team's interactions. (95%)
- I can identify when peers in an HFOSS project are behaving in an unprofessional manner. (81%)

Attraction and retention of students – Finally, there is good evidence that the experience with the humanitarian FOSS projects positively affected the students' perception of computing as a major and career choice. This is visible to some extent in the questions discussed above under motivation.

- Participation in an HFOSS project has caused me to consider computing as a major or minor. (63%)
- Participation in an HFOSS project has positively reinforced my decision to make computing my major. (77%)
- Participation in an HFOSS project has caused me to consider taking further computing courses. (52%)

While the questions above show a majority of students agreeing or strongly agreeing, the response is not as strong as in other areas. In part, this may be because of the mix of students represented in the projects being measured. The population includes students relatively early in their studies and also students close to graduation. For the more advanced students, it is reasonable that their feelings about their major are set, and the humanitarian FOSS experience would be less likely to have much influence in this area.

To test this idea, an additional analysis for the above three questions was performed. The population was divided by reported programming ability. One group consisted of students who classed themselves as more experienced (values of 4 or 5 where 5 was "experienced") and the other group was the students with less programming experience (values of 2 or 3 where 1 was "novice" but no student selected 1). The less experienced

students agreed quite strongly with the statements about impact of the humanitarian FOSS experience on their view of computing majors.

- Participation in an HFOSS project has caused me to consider computing as a major or minor. (100%)
- Participation in an HFOSS project has positively reinforced my decision to make computing my major. (100%)
- Participation in an HFOSS project has caused me to consider taking further computing courses. (88%)

The difference between the more experienced and less experienced groups was tested for each of the questions above using a two tailed t test for groups with unequal variance. The difference between the groups was significant at the $p < .05$ level for each of these questions. In short, it appears that the HFOSS experience has a positive impact on getting students to view computing majors and careers more favorably, and that this impact is particularly pronounced among students with less programming experience.

These initial results show promise for the benefits that can be reaped from involving students in HFOSS projects. The overall evaluation is ongoing and expanding with a pre-course survey that corresponds to the post-course survey being administered beginning with spring 2009 offerings of HFOSS courses. The combination of a pre-course and post-course survey will allow the change in attitudes from the beginning to the end of the course to be examined. Additional focus groups of students are also planned in the hopes of more precisely identifying the success factors for the use of HFOSS in education.

5. WORK IN PROGRESS

The prior sections discuss some of the initial results of using HFOSS projects in education. Work is continuing at the initial sites, along with efforts to help expand this effort to additional institutions. To support this expansion, definition of possible approaches to using HFOSS in education is needed. These definitions need to be embodied in course materials and other resources that can be made available to interested instructors. There is also a need to explore use of HFOSS at different points in the computing curriculum. Results of this effort are and will be made available on the Web at [21]. The assessment effort described in this paper includes development and evaluation of the following items.

Software development process – In order for students to produce a product within an HFOSS course, they must employ a software development process that fits in the educational environment as well as complying with the process used by the HFOSS community that supports the project. The degree that any defined process is employed in open source projects varies greatly. Some open source projects have a relatively well-defined process while others use a very ad hoc approach. The process for HFOSS project courses must be defined and evaluated to determine what is effective.

Educational content areas - The courses and internships that involve students in HFOSS must address student learning in a series of content areas. These content areas include software engineering knowledge, concepts of FOSS, principles of developing open source software, and the particulars of an

individual HFOSS project. Students must be able to use appropriate tools and technologies to develop the project. Finally, in most courses, students must understand team dynamics and how to work effectively in a team.

Educational infrastructure – Since the working environment for HFOSS projects is similar to the working environment of professional developers, there are usually more complicated requirements to create hardware and software environments that students need to be productive. This infrastructure may confer an additional burden on the instructor to create and maintain, or to teach students to create and maintain.

In short, the on-going work is exploring how to effectively use HFOSS in education and working to provide as much support as possible to instructors interested in pursuing this approach.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0736874 and 0722137. 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.

7. REFERENCES

- [1] Information Technology 2008. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology.
- [2] Computing Curricula 2005: Computer Science. The Overview Report. IEEE CS and ACM 2005.
- [3] ACM/IEEE. 2004. Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE CS / ACM 2004.
- [4] Fernandez, J. D., Garcia, M., Camacho, D. and Evans, A., 2006. Software engineering industry experience: the key to success. *Journal of Computing Sciences in Colleges*, 21, 4.
- [5] Spinellis, D. 2006. Open Source and Professional Advancement. *IEEE Software*. 23, 5, 70.
- [6] Samuelson, P. 2006. IBM's pragmatic embrace of open source. *Communications of the ACM*. 49, 10, 21–25.
- [7] Patterson, D. 2006. President's Letter. *Communications of the ACM*. 49, 3, 27-30.
- [8] Kapor, M. 2005. How is Open Source Special? *EDUCAUSE Review*. 40, 2, 72-73.
- [9] Apikul, C. 2006. Managing Disasters – Sahana. The International Open Source Network, retrieved from <http://www.iosn.net/foss/humanitarian/projects/sahana/> on January 21, 2009.
- [10] Morelli, R.A., Tucker, A.L., Danner, N., de Lanerolle, T.R., Ellis, H.J.C., Izmirlı, O., Krizanc, D., and Parker, G. 2009. Revitalizing Computing Education by Building Free and Open Source Software for Humanity. *Communications of the ACM* – forthcoming 2009.
- [11] Ellis, H.J.C., Morelli, R.A., et al 2007. Can Humanitarian Open-Source Software Development Draw New Students to CS? *SIGCSE* 2007.
- [12] Ellis, H.J.C., Morelli, R.A., and de Lanerolle, T. 2007. Holistic Software Engineering Education Based on an Open Source Project. *20th Annual Conference on Software Engineering Education and Training*. 2007.
- [13] Morelli, R.A., Ellis, H.J.C., de Lanerolle, T., Damon, J., and Walti, C. 2007. Can Student-Written Software Help Sustain Humanitarian FOSS? *The 4th International ISCRAM Conference*. 41-44.
- [14] Ellis, H.J.C., and Hislop, G.W. 2008. Fostering the Community of Software Engineering Educators. *21st Annual Conference on Software Engineering Education and Training*.
- [15] Ellis, H.J.C., Morelli, R.A., and Hislop, G.W. 2008. Support for Educating Software Engineers Through Humanitarian Open Source Projects. *21st Annual Conference on Software Engineering Education and Training*.
- [16] Ellis, H.J.C., Morelli, R.A., and Hislop G. W. 2008. WIP: Challenges to Educating Students within the Community of Open Source Software for Humanity. *The 2008 Frontiers in Education Conference*.
- [17] de Lanerolle T., Morelli R., Danner N., Krizanc D., Parker G., and Izmirlı O. 2008. Creating an Academic Community to build Humanitarian FOSS: A Progress Report in *Proceedings of the 5th International ISCRAM Conference Washington,DC,USA*, 337-341.
- [18] Morelli, R.A., de Lanerolle, T.R. 2009. FOSS 101: Engaging Introductory Students in the Open Source Movement, *SIGCSE* 2009.
- [19] Computing Research Association. 2005. Taulbee Report. <http://www.cra.org> last accessed on January 21, 2009.
- [20] Cohoon, J. McGrath. 2006. Just Get Over It or Just Get On with It: Retaining Women in Undergraduate Computing. in *Women and Information Technology – Research on Underrepresentation*, J. McGrath Cohoon and William Aspray eds. The MIT Press, Cambridge MA, 205-237.
- [21] Software for Humanity site: <http://edudev.hfoss.org>