Work in Progress - Challenges to Educating Students within the Community of Open Source Software for Humanity

Heidi J. C. Ellis, Ralph A. Morelli, Gregory W. Hislop
heidi.ellis@trincoll.edu, ralph.morelli@trincoll.edu, hislop@drexel.edu

Abstract - This WIP describes the challenges faced by the SoftHum (Student Participation in the Community of Open Source Software for Humanity) project in incorporating undergraduates in Humanitarian Free and Open Source Software (H-FOSS) projects. The goal of SoftHum is to develop course-level support for the use of H-FOSS as a foundation for software engineering education. Such support will take the form of course materials and a process to support student success in a community-based software experience using H-FOSS, and the documentation of a classroom environment that supports student open source experience. We present a brief overview of the project, discuss the challenges we face in involving students in H-FOSS projects, and present our current progress.

Index Terms - Open source software development, software engineering education, humanitarian software, software engineering courseware.

INTRODUCTION

The sharp decline in enrollment in computing programs since about 2001 is well documented as is the loss of gains that had been made in attracting women students and other traditionally under-represented groups. While there appears to be some signs that enrollments in computing programs are starting to turn around, the demand for software engineers is still rising at a faster rate than that of students entering software engineering programs. Clearly, the mismatch of a low number of software engineering students and high demand for these graduates will not meet national needs.

An approach that has been used to successfully attract computing students is involving students in the development of Humanitarian Free and Open Source Software projects (H-FOSS). A collaborative program between Trinity College, Wesleyan University and Connecticut College called the Humanitarian Free and Open Source Software (Humanitarian-FOSS) project (www.hfoss.org) was established a year ago and received NSF funding to create the high-level infrastructure required to support direct student involvement in H-FOSS projects.

The Sahana project [1] and the OpenMRS [2] project are two representative projects in which students have become involved. The Sahana project is a disaster relief application for use in disasters world-wide and the OpenMRS project supports development of customized medical record systems to aid in medical informatics efforts in developing countries. By participating in these H-FOSS projects, students have gained real-world experience, grown in professionalism and increased their social consciousness. Our experience with Sahana and OpenMRS suggests an approach to computing education that can attract students to computing majors including software engineering, provide an excellent learning vehicle, and achieve social benefits [3, 4, 5]. This education has taken different forms including independent study courses, summer internships, and special topics courses. We have even experimented with a course on developing H-FOSS for non-majors.

The SoftHum project (Student Participation in the Community of Open Source Software for Humanity) [6] grew out of the Humanitarian-FOSS project. The SoftHum project focuses on the classroom infrastructure required to scaffold student learning such as a software development process, course assignments, and assessment.

THE CHALLENGES

Involving students in an open source project has several inherent challenges. A key to understanding these challenges is that SoftHum involves engaging the development community that supports various H-FOSS projects. This is quite different from simply using the code base available in open source projects as a basis for further development, in other words, to have students build on this code but without submitting their results as project additions. SoftHum takes a major step beyond that approach by making students full participants in the H-FOSS community. Challenges to accomplishing this community involvement include:

Student inexperience: Students typically lack sufficient skill and knowledge to be able to enter into a large-scale software project and make an immediate contribution. Areas in which students may lack skills include technological background, experience working in teams, and project management techniques. How do we structure the course or internship so as to allow students to make a real contribution to an H-FOSS project?

Limited course duration: Student involvement in H-FOSS projects must be accomplished within the constraints of the academic environment. This typically means that students will be involved with projects for only one or two terms and will work only a limited number of hours per
The current software process used in courses involving students in ongoing H-FOSS projects typically follows one iteration through a standard software development cycle (requirements-design-implement-test). Students spend a brief period of time at the beginning of the course learning key technologies including PHP and Subversion. Then students are required to develop a Problem Statement document for an enhancement to or customized usage of an H-FOSS project. Next, students develop a Software Requirements Specification (SRS) which is reviewed for completeness. Once the SRS has been completed, students develop a Software Design Specification (SDS) to document the design of their project. During the most recent course offerings, students have also been required to develop Customer Documentation and Programmers Guide documents, after they have implemented their system. Students also use PHPUnit or JUnit to organize test cases.

Many of the documents that support the courses are loosely based on IEEE documentation specifications [7]. However, these documents have been pared down and modified to support student learning. Additional modified documents exist for implementation documentation, test case specification, project tracking, etc. but have not yet been incorporated into the process supporting courses using H-FOSS. Other documents such as Risk Assessment and other project management documents may need to be developed.

Our next step is to do an in-depth analysis of the current software process including investigating aspects such as selection of projects, ideal project characteristics, student assignment to projects, and grading. A software process for developing open source software in an academic environment will be specified. A minimal set of components required to support the process will be developed and a description of how each component may be established will be provided. Grading metrics and descriptions will also be supplied to aid instructors in using the process in courses.

ACKNOWLEDGMENT
This material is based upon work supported by the National Science Foundation under Grant Nos. 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 (NSF).

REFERENCES

AUTHOR INFORMATION
Heidi J. C. Ellis Assistant Professor, Trinity College, heidi.ellis@trincoll.edu
Ralph A. Morelli, Professor, Trinity College, Ralph.morelli@trincoll.edu
Gregory W. Hislop, Associate Professor, Drexel University, Hislop@drexel.edu

Session S3H

October 22 – 25, 2008, Saratoga Springs, NY
38th ASEE/IEEE Frontiers in Education Conference
S3H-8