CSci 393 Open Source Software Development Syllabus and Structure

Overview

This course exposes the student to working on Free and Open Source Software (FOSS) projects. Through a sequence of self-guided activities and frequent, interactive meetings with the instructor and a small group of students, the student will be introduced to the historical context of FOSS and to the technology, tools, culture, intellectual property rights, and conceptual processes and practices that are critical to being a contributing member of an open source software project and its community. To the extent that it is possible, emphasis will be placed on Humanitarian Free and Open Source Software (HFOSS) projects.

Objectives

Among the (deliverable) outcomes of this project are that

- the student will become a contributing member of a software development community and have documented evidence of their contribution; and
- the student will document their efforts on this project by creating a blog that chronicles their work.

Specific technology that the student will explore and learn to use well includes:

- version control systems (e.g., git)
- issue trackers
- communication channels
- documentation and wikis
- linux/unix programming environment

Prerequisites

All students should have completed the first three programming courses, CSci 127, 135, and 235, and Math 150.

Course Structure

The first run of this course is a collection of independent study projects with a group of about ten to fifteen students. It will be run officially in Fall 2018, but there will be periodic meetings over the summer with those students who are available, about once every two to three weeks. The agendas for these meetings will not include anything essential to the course unless all students are present. The purpose of the meetings is mostly to build enthusiasm.

In the fall, there will be meetings every two weeks. Much of the learning will be as independent activities, and much of the communication and delivery of materials will be over the web, using a combination of Piazza, the course website, and GitHub.

Because there is no official class meeting times, the number of hours allocated to each topic is not constrained by the college's contact hour limits. The students have been told to expect to work about one hundred hours for this project over the course of the semester.

Syllabus

1. Context and Overview
   (a) Open-ness in general: open source, open data, open hardware, etc.
   (b) History and background of open source software
   (c) Open source culture and community
2. Contributing to Projects
   (a) Early and Easy Contributions
      - open wikis
      - open maps

   (b) Types of contributions to software projects
   (c) Ways of getting involved
3. Tools and Technology
   (a) Programming Tools and Technology
      - markdown, make, gdb, Unix shells

   (b) Software engineeering tools required for group software projects
      - version control systems (e.g., git)
      - issue tracking
      - documentation tools (e.g., Doxygen)

   (c) Software engineering tools specific to distributed group projects
      - communication tools
      - remote, distributed version control (GitHub)
      - online, web-based issue trackers
4. Intellectual Property Rights and Licensing
   (a) types of licenses
   (b) licensing your own work
   (c) avoiding plagiarism
5. Project Evaluation and Studies
   - Evaluating the suitability of a project for the purpose of their contributing to it.
6. Team Selection
7. Project Selection/Assignment
   (a) Getting involved in the community
   (b) Setting up project development environment
   (c) Picking some issues to work on
   (d) Solving the issues and issuing pull requests
8. Team Reports