Syllabus
CSCI 4830-009
Open Source Development
Spring, 2017
M, W 4:00 – 5:15
ECCR 151

Instructor

Ned McClain
Department of Computer Science
Office Hours: Tu 11:00-12:00 @ ECCS 122

Course Objectives

This course provides both a practical and theoretical foundation of the technical, social and legal frameworks of (free/libre) open source software projects. The course is designed to provide hands-on mentoring and support to students as they learn to use the tools and procedures to make a first contribution to a real project. The course also covers theoretical aspects such as the history, technologies, common practices, and legal and organizational considerations when joining existing or starting new open source projects. Students are expected to join and make contributions to existing open source projects as part of this course.

The course is designed to benefit both newcomers and experienced developers and contributors to the world of Open Source. Basic programming experience is assumed, but not required (non-programming contributions can be made). Some familiarity with common software development methodologies (software engineering) might be helpful.

Topics to be Covered

- Open Source History and Culture
- VCS (Git), Repository Structure, Workflows
- Tooling and Using Open Source
- Open Source Business Models
- Licensing and Legal
- High Availability and Disaster Recovery
- Project and Community Management and Standards
- Lean, Agile, Kanban, DevOps
- Automation, CI, Monitoring, Analytics
- Security, IaaS, Microservices, Containers

Course Materials

The Cathedral and the Bazaar, Eric Steven Raymond, 1997
http://www.unterstein.net/su/docs/CathBaz.pdf

Prerequisites
CSCI 2400 or ECEN 3350 (minimum grade C-).

Course Requirements
You are expected to attend all lectures, complete assigned readings, complete 4-6 homework assignments, and contribute to a public open source project. Your semester grade will be based 40% on the homework assignments and 60% on the open source project.

Students average 1-2 hours of effort per homework, and as little as 8 hours for their public open source contribution. Some students may find open source to be personally and professionally rewarding, and choose make significantly larger contributions. If you find yourself spending too much time, not understanding the concepts, or have a scheduling emergency, you should email Ned immediately or chat after class.

Student Outcomes

a) Apply Knowledge
b) Computing Requirements
d) Teamwork
e) Professional Issues
f) Communicate Effectively
g) Analyze Impacts
h) Professional Development
i) Current Techniques
k) Design & Development

Academic Honesty
Working with your fellow students—for example, studying together for exams—can be very helpful in learning. Studies have shown that the single most important factor in determining the success of students was whether or not they had developed intellectual relationships with other students. On the other hand, some ways of working together can strongly interfere with your learning—for example, copying their answers. We will adhere to the University's Honor Code and US Copyright Law, and you will be required to certify that all work is your own.