MET CS 893 - Special Topics in Open Source Development

Ronald Czik
Rev. 4/12/2008
v0.4

Introduction

This graduate level course builds on previous design and programming courses and introduces students to the technological, social, and pragmatic aspects of developing open source software through direct involvement in an open source project. Students will learn to use the tools, techniques, and strategies of open source developers. This is a project-based laboratory course. Students will be directly involved with and integrated into an open source project. They will be expected to be active participants in the project and contribute to the project in various ways. Projects in this course will use the Android platform developed by Google. Android is a software stack for mobile devices.

Course Goals

Students will become familiar with the open source movement, its philosophy and history, the open source process and its development methods, tools, communication mechanisms and licensing issues. During the semester students will:

- Choose between the various open source licenses and learn the implications for users, developers, and the software community in general
- Use the communication modes particular to the open source world through participation in such things as mailing lists, IRC, wikis, etc.
- Become familiar with and become adapt using the tools of open source development, for example: distributed revision control; documentation tools; automated build and test systems; debuggers; source code utilities; tracking systems; on-line resources, etc.
- Write software that integrates and interacts with the open project’s code. For example: add-ons; bug fixes; new features; etc.
- Learn and understand Agile development methodology and use it to develop open source software within the project
- Work collaboratively with fellow students and other members of the project’s community

Prerequisites

- CS 673 (Software Engineering) or its equivalent
- CS 565 (Advanced Java Programming) or its equivalent

Intellectual Property

Given that this course is focused on open source development, and that students will be working on developing open source code, all student work will become open source. The particular license used will be determined by the project to which the implementation contributes.

Grading

- 60% - Project Deliverables (e.g., code, documents), marked in terms of quality(70%), and quantity (30%). There will be 3 deliverables through the semester, v0.1, v0.2, and the final v0.3 at the end of the semester.
- 20% - Project Participation through Wiki pages and updates. Students will be graded on project and personal page's quality, depth of explanation, frequency of update, etc.
- 20% - Contributions to other projects. Since open source development is a collaborative endeavor, students are expected to contribute in areas outside of your portion of the project. Students will be graded on the quantity and quality of their contributions to other groups.

Course Syllabus

- Lecture 1 – Course introduction
    - Intro to open source. What is open source?
    - License Issues (MPL, GPL, LGPL, etc.)
    - Contrasting and comparing open source vs. traditional development methodologies
- Lecture 2 – Project Introduction
    - General purpose computer vs. handheld computer applications
        - HW/SW constraints
        - UI
        - User needs
        - Network
    - What is Android?
        - Project goals and introduction
        - Review technologies used (Java, XML)
    - Discuss and assign projects
- Lecture 3 – Collaboration, Community and Communication
    - Using IRC and integration with the project community
    - Wikis, Wikipedia, course pages, adding pages
    - Communication and etiquette

- Lecture 4 – Managing and Build Project Source
    - Distributed source code control (CVS, SVN)
    - Make
    - Building the source
- Lecture 5 – Debugging and Development Tools and Tests
    - Defining bugs
    - Filing bugs with the project
    - Debugging techniques
    - Using Eclipse and Android SDK
    - First application (Hello Android)
- Lecture 6 – Exploring the Android API
    - Perspective and architecture overview
    - Design philosophy
    - Anatomy of an Android Application
    - Application life cycle
- Lecture 7 – UI
    - Implementing a User Interface
    - Building blocks
- Lecture 8 – Data Storage and Retrieval
    - SQLite
    - Integrating with an application
- Lecture 9 – Security
    - Security model
    - Security considerations
    - Implementing
- Lecture 10 – Building Custom Components
    - Extending existing components
    - Building new components from scratch
- Lecture 11 – Optional and External APIs
    - Location based
    - Media
    - 3D graphics
    - Integrating Google APIs and External Services
- Lecture 12 – Project Presentations
    - Students present and demo their projects
    - Class discussion on lessons learned

Texts and Course Resources

- "*Cathedral and Bazaar*" by Eric Raymond
- "*Code Reading: The Open Source Perspective*" by Diomidis Spinellis
- Various handouts by the instructor – TBD
- Project and technology related handouts – TBD
    - GNU documents
    - Open Source Initiative
    - GPL
    - Android documentation
    - Other project related notes and hand outs
- If possible, Android related staff either virtually on-line or if local in the class room to guest lecture
- Course Wiki and IRC
- Source control server