LinuxFest Northwest 2016

Improving Computer Science Education with FOSS-style Projects

Roberto C. Sánchez

About the Presenter

- I am a software engineer and consultant
- I have been programming since 1990
- I have been active in FOSS since 2002
- I teach Computer Science/Engineering at Wright State University (Dayton, OH)
- Find me here:
  http://www.linkedin.com/in/robertocsanchez/
  https://github.com/rsanchez-wsu

About the Course

- Title:
  - CEG3120 - Introduction to the Design of Information Technology Systems
- Catalog Description:
  - Introduction to the design of information systems comprising modern technologies such as SQL database programming, networks, and distributed computing with CORBA, electronic and hypertext (HTML) documents, and multimedia.

My Objective

- When I started teaching I wanted to give my students what I had missed in school
- I wanted to craft a project experience that:
  - Covered academics required in the course
  - Introduced students to a FOSS-style project
  - Gave students a taste of “large” project work
  - Taught more than “what is in the book”
- I wanted to leave them with something valuable, that they could recall later

Outline

- Main Points:
  - Introducing new techniques/technologies
  - Meeting the course academic requirements
  - Structuring around a project concept
  - Further broadening the students' horizons
- Aspects:
  - Opportunities
  - Challenges

Presentation Format

- Each point/topic will follow this format:
- *Some enlightening/insightful thing*
  - *[C] A challenge posed by the above thing*
  - *[O] An opportunity presented by the above*

New Techniques/Technologies

- Source Code Management (Git)
  - [C] Few students (~4/class) have used SCM
  - [C] Some students muddle their way through
  - [O] Most eventually realize the value of SCM
- Project Management (GitHub issues, wiki)
  - [C] Managing issues/docs takes time from dev
  - [O] Non-programming tasks are important
  - [O] Teams collaborate better when they can see what tasks others are working on

New Techniques/Technologies

- Continuous Integration (Jenkins)
  - [C] It takes time for students to adopt the mindset that “broken build == bad thing(TM)”
  - [O] CI teaches that a working build is a feature
- Static Analysis (Checkstyle, FindBugs)
  - [C] Students get frustrated by cryptic errors
  - [C] There is an upfront productivity loss
  - [O] Students learn what buggy code looks like
  - [O] Integration of code modules is easier

New Techniques/Technologies

- Collaboration (within teams, among teams)
  - [C] Unequal distribution of effort within teams
  - [O] Students learn inter-team coordination and how parts of a project depend on each other
- Communication (Slack)
  - [C] There is a tendency for meeting in-person
  - [O] Students learn asynchronous, distributed, and remote collaboration/interaction

Academic Requirements

- Java topics to cover: Swing, networking, RMI, XML, JavaBeans, JDBC
  - [C] Difficult to find one project idea for all items
  - [O] Students work on multifaceted code
- Still need homework, quizzes, exams, etc.
  - [C] Hard to tie to together assignments/project
  - [C] Grading takes effort (I have no TA and use only short answer/essay questions)
  - [O] Focus is on “learn by doing” not on “study”

Project Concept

- Must be accessible to varying skill levels
  - [C] Despite pre-req's, many students arrive with weak programming skills
  - [O] There are lots of non-programming tasks
  - [O] Team members can help each other
- Must scale to a 15 week semester
  - [C] Each team needs viable weekly tasks
  - [C] “Real” projects experience hiccups
  - [O] Students learn to work within constraints

Project Concept

- Students are empowered to decide/design
  - [C] Students not accustomed to self-direction
  - [C] The “norm” is very detailed specifications
  - [O] Teams learn to become the project drivers
- Instructor still needs to mentor/evaluate
  - [C] Reviewing all commits/issues takes time
  - [C] Students need guidance on good coding
  - [O] Instructor gets to practice PM skills
  - [O] Students can help with peer review

Broadening Horizons

- I use readings from *Joel on Software* as homework assignments
  - [C] There are so many candidates for readings
  - [O] Students gain some outside perspectives
- I try to get students thinking about more than just coding a weekly assignment
  - [C] Everyone is busy (work, family, classes)
  - [O] Students can make a better decision whether “programming” is really for them

Parting Thoughts

- Teaching in this manner is challenging
  - I get to be instructor, project manager, mentor, customer, and so on (time consuming)
  - Grading is Not Fun(TM) but is required
- Teaching in this manner is rewarding
  - Students tell me the experience is unique, has helped them improve as programmers, has helped in job interviews, has helped them decide they love/hate programming, etc.
  - It makes me a better developer

Questions?

Evaluate this session

Improving Computer Science Education with FOSS-style Projects

Session videos will be posted to the LinuxFest Northwest YouTube channel.

Thank you!