Software Development in Practice

How software engineering teams really work?

William Chan
Lead Platform Engineer, 605.tv, Capital One, FreeWheel (Comcast)

Course Objective

- Learn real life software engineering processes outside of the classroom
- Practice real life software engineering practices
- Become software engineers!

Course Breakdown

1. Software Development Methodologies
2. Product Development and Requirements
3. Software Design
4. Software Implementation

Project Breakdown + Technical Requirements

Choosing a project that will help you succeed in this course for grades sake

- Use a relational database or a non-relational database if you can prove a need for it
- A web or desktop application that stores and reads data from a database
- A working demo by the end of the semester that will be presentable to potential employers

Work Environment

- I am the Director of Engineering
- The class will be divided into individual teams
- Teams will be gathering requirements for the product, designing and implementing the application
- 360 Reviews: Peer reviews, getting reviewed by other team members

Typical Project Roles

- Product Manager/Owner - someone who is close to the users and knows the problem that the application solves; the stakeholder in the most general sense
- Software Engineers - responsible for technical design and implementation of the product
- Project Managers/SCRUM Master - person responsible for removing obstacles that block software engineers from delivering the product
- Program Manager - typically in larger organizations to manage teams that break down into multiple products

Presentation Breakdown

1. **Product**: Discuss the overall product mission and what your application intends to solve; may include business idea and notes
2. **Planning**: Milestones for the application from the beginning of the semester to the end of the semester
3. **Requirements:** List the requirements and reasons behind why that constitutes your MVP
4. **Technical Design**: UML diagrams and additional slides that demonstrates the reason to the design
5. **Tests**: What tests will you have in place to ensure the quality of your application
6. **Demo**: Be able to present a partial or preferably a full demo; if its a partial demo, explain why the project could not be completed as is
7. **Retrospective**: knowing what you know after having implemented the project, what would you have changed or think you can do better?
8. Questions: The ability to defend any section of the presentation

Note: there will be assignments that will be the milestones that can be laid out in the presentation

What does a generic software development process look like?

1. Define the product
2. Gather requirements
3. Design the application
4. Implement the application
5. Test the application
6. Release the application
7. Gather user feedback

Software Methodologies

- Waterfall
- Kanban
- SCRUM, SAFe, LeSS and other variants
- Lean

Software Engineering Organizations

- Bureacractic - top down approach where managers make most of the decisions
- Functional - groups of people under functions such as front end developers, back end developers, etc
- Divisional - cross functional teams where there may be front end developers, back end developers, product managers, etc
- Matrix - organized around different products that are delivered
- Flat - everyone reports to the CEO or where have minimal layers of managers

Software Engineering Career Tracks

- Managerial - people manager, generally responsible for the well being of the team and growing people
- Technical - growing people by technical means and ensure technical delivery of a project

Thank you

William Chan
Lead Platform Engineer, 605.tv, Capital One, FreeWheel (Comcast)
http://linkedin.com/in/wchan2 (http://linkedin.com/in/wchan2)