CS 130: Software Engineering

Lab 1C: Week 1 Discussion

Overview

Week 1: Team Formation, Part A Requirement, Project Management, ArgoUML

Week 2: Design Pattern

Week 3: Part A Presentation, Part A is due on Thursday 11:59 PM.

Week 4: Part B Requirement, Midterm Review

Week 5: Software Testing

Week 6: Part B Presentation

Week 7: No lab (Veterans Day)

Week 8: Code Review

Week 9: No lab (Thanksgiving Holiday)

Week 10: Final Review

Group Project: Rules and FAQ

- 35% of your final grade
- All of the team members **must** be **enrolled in the same discussion section** since your TA will be in charge of grading your group project
- You **must attend** the **lab section** that you are enrolled in
- 5~6 team members per team. **No exceptions**
- You will also be graded based on **your teammate's feedback / evaluation**

Team Formation

- If you already have a team but are looking for more members
  - Tell everyone about your idea and needs!
- If you are looking for a team
  - Tell everyone about your skills and experience!

Part A Requirements

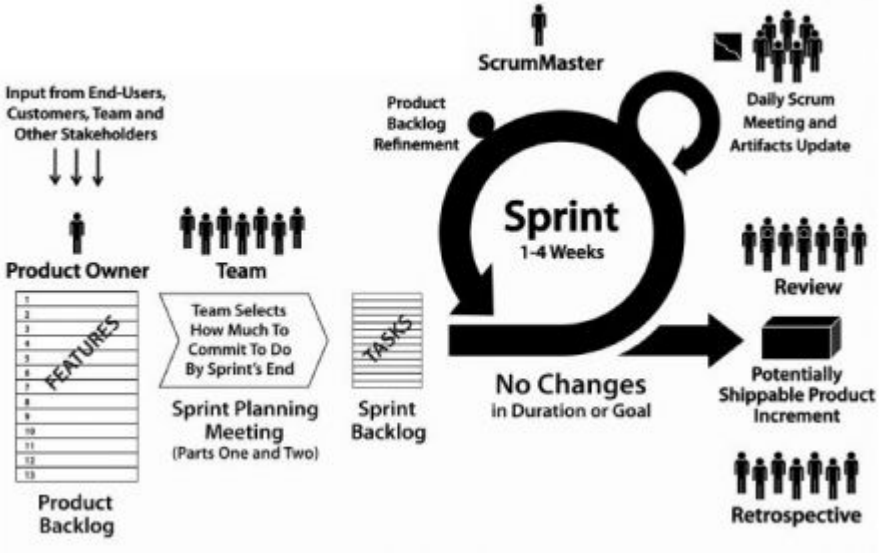
- **Due October 13th, 2016, Thursday, 11:59 PM. Upload everything on CCLE**
- Form a team
- Set up a project repository on **Github** and add me as a watcher
- Mock-up designs (ex. screenshots)
- Setup **Trello** with tasks and add me as a watcher
- Write out a **proposal report**
    - Sample Report
- Prepare a 7-minute **presentation** followed by a 3-minute Q&A session
- **Detailed Requirements** and **additional sample report** will be uploaded on **CCLE** by tonight

Project Management Tools

- Trello: Free Project Management Tools
  - Bi-weekly stand-up
- Slack: Free Team Messaging Tool

Scrum in Industry

- Scrum is a lightweight agile development process.
- The organization is split into small, cross-functional and self-organizing teams, e.g., product owner, scrum master, and scrum team.
- The time is split into short fixed-length iterations (sprints).



Scrum in Industry (Cont.)

- The project is described as a list of features: the **Product Backlog**.
- Each feature is described in terms of **User Stories**.
- **Sprint Planning Meeting** to pick stories from the product backlog, estimate the work (hrs) for each story, and make a **Sprint Backlog**.
- **Daily Scrum Meeting** to discuss **What did you do yesterday? What will you do today? Any obstacles?**
- Product owner owns the backlog, e.g., talking with customers/stakeholders
- ScrumMaster is the coordinator of the development process, e.g., hosting scrum meeting, exchanging information between product owner and the team
- Scrum team: developers, quality engineers, etc.

Bi-weekly stand-up

- Teams are required to do a bi-weekly stand-up with me for participation grade
- Quick update / walkthrough on who’s working on what, how much time you spent etc.
- What did you work on last week? What will you work on this week? Any obstacles?

UML Modeling Tools

- Gliffy Online
- ArgoUML

* ArgoUML Tutorial
http://users.ece.utexas.edu/~miryung/teaching/EE461L-Spring2012/labs/uml.htm

Websites for Free Icons and Background

- Free icons
  - http://fontawesome.io/
  - https://design.google.com/icons/
- Free high-quality photos
  - https://unsplash.com/
  - https://pixabay.com/

Lecture Recap

- Use-case Diagram
- State Diagram
- Class Diagram
- Sequence Diagram

Class Diagram

- Difference between dependency, association, aggregation, and composition are ambiguous and up to your own interpretation
- On quizzes, midterm, and final, we will **NOT** trick you. Pick the **best** answer. **Do not** put **dependency** for every single question. **Justify** your answer.
- Dependency: **uses** relationship (ex. argument, local variable, etc.)
- Association: **has** relationship (ex. member variable)
- Aggregation: **owns** relationship (ex. arraylist of an object as a member variable)
- Composition: **is made up of** relationship (ex. private inner class)

Quiz Next Wednesday! (10/5)

- You should know how to solve these problems
  - Given a use-case diagram, briefly write what the program does
  - Given a state diagram, briefly write what the program does
  - Given a **class diagram**, write the code interpreting the given class diagram
  - Given a **sequence diagram**, write the code interpreting the given sequence diagram
  - **Parnas' Information Hiding Principle** (Next Monday's Lecture)
- Review **Think Pair Share** and **Review Questions**!