CS 360 Open Source Project Presentation   **Monday, Nov 21, 2016**   20 points

Communication is a very important skill to any developer, open source or otherwise. **Strain 222**

Your group will have **7-8 minutes** to present to the class the open source project you will be working on for the rest of the semester. This needs to be a **professional and practiced** presentation that walks the class through the project you chose, why you chose that project, how development will proceed, and what your final project should look like.

You are encouraged to build slides in Google Docs and share them with will4614@pacificu.edu before the presentation.

You are encouraged to log on to the TV machine (just use your Linux password) and view your slides *to make sure they are readable from the back of the room* and practice your speaking! If you don't know how to get the TV machine to work, talk to any CS professor. Do not use the whiteboard.

Each team member must speak during this presentation. The presentation should be roughly split equally among the team members.

►You will be expected to ask good questions to other groups after their presentation.

If you are creating your own project:
You need to address the following topics (not coincidentally mostly the same topics that are in your README.md file)

   Brief description of the project
        Why is this useful?
        What other open source projects out there solve the same or similar problem?
        Can you mock up a simple screen shot of the final product?

   ► How will each team member contribute?

   Who are your users?

   How will your teammates communicate with each other?
        How will outside contributors communicate with you?

   What libraries/external software does this project rely on

   How do users report bugs (Will you use GitHub Issues or something else?)

   How do users contribute code
        How many positive reviews does a pull request require?
        Can a user merge their own pull request?
        What is your statement of coding standards?
        How will testing be handled? Will you require unit tests?

   What license did you chose and why? How do you want the community to be able to use/build off of
        this project in the future?

If you are contributing to an exiting project:

Brief description of the project
  Why is this useful?
  What other open source projects out there solve the same or similar problem?

What do you plan on doing for this project over the next month?
  Where is this project hosted?
    What version control software is used?
  How does the project communicate? Email, IRC, etc?
  How do users report bugs?
  What is the project's development process?
    How do users contribute code (some of the following questions may or may not be relevant)
      How many positive reviews does a pull request require?
      Can a user merge their own pull request?
      What is the project's statement of coding standards?
      How will testing be handled? Are you required to write/run unit tests?

  Is there documentation for a new contributor to get started?
      How good is that documentation?
      Are you going to help update that documentation?
  What challenges do you see in working with this project that worry you?
  How do you build and test the project?

► How will each team member contribute?

What libraries/external software does this project rely on?

What languages/technologies do you need to learn/understand to work on this project?

What license is the project released under?