COS 333
Course Conclusion

Robert M. Dondero, Ph.D.
Princeton University

Objectives

- Summarize the course
- Describe the end-of-semester schedule
- Describe project deliverables
- Have project team meetings

Agenda

- **Course summary**
- End-of-semester schedule
- Project deliverables
- Project team meetings

Course Summary

- We have covered:
  - Three-tier programming
    - The Python language, database programming, network programming, concurrent programming with multiple processes & threads, server-side web programming, the JavaScript language, client-side web programming, security issues in web programming, XML programming, server-side options, client-side options (desktop & native Android/iOS apps)

Course Summary

- We have covered (cont.):
  - Software engineering
    - All through the course, but with focus at the end…
    - Requirements analysis, design, implementation, debugging, testing, evaluation, maintenance, process models

Agenda

- Course summary
- **End-of-semester schedule**
- Project deliverables
- Project team meetings

Schedule

- **After this week's meeting with your TA adviser & before your presentation**
  - (Strongly suggested) Declare feature freeze
  - (Strongly suggested) Meet with your TA
    - Rehearse project presentation
    - Discuss final app version
    - Discuss deliverables
  - Polish, debug, test, evaluate, document

Schedule

- **After your presentation & before Dean's Date**
  - Polish, debug, test, evaluate, document

Schedule

- **Dec 13 (Dean's Date) at 5PM**
  - Finalize Google team directory
    - *Project Overview* doc (from early in course)
    - *Timeline* doc (from throughout the course)
    - Presentation slides
    - *Grader's Guide* doc
    - *Product Evaluation* doc
    - *Project Evaluation* doc
    - Source code (compressed file)
  - Finalize GitHub repo
  - Finalize product!

Schedule

- **Dec 14 to (at least) Jan 7**
  - Don’t touch Google team directory, GitHub repo, deployed system
  - Be available to answer grader questions
- **Jan 7**
  - Course letter grades to Registrar
- **Jan 8 (approx)**
  - Project grade reports to you

Agenda

- Course summary
- End-of-semester schedule
- **Project deliverables**
- Project team meetings

Project Deliverables

- Did you…

Deliverables: Presentation

- **Presentation**
  - Cover these topics:
    - Background and motivation?
    - Functionality?
    - Design?
    - Reflection?

Deliverables: Presentation

- **Presentation: Background & Motivation**
  - Provide adequate background?
    - Compose the presentation for the appropriate audience (primarily COS 333 instructors & COS 333 students)?
  - Motivate your system well?
    - Describe existing systems, and compare & contrast your system with them?
    - Consider organizing by scenarios?

Deliverables: Presentation

- **Presentation: Functionality**
  - Describe what your system does?
  - Describe and demonstrate your system's core functionality well?
    - Consider organizing by scenarios?

Deliverables: Presentation

- **Presentation: Design**
  - Describe how your system works?
  - Provide a good description of your system's design at an appropriate level of detail?
    - Provide a graphical system architecture overview?
    - Provide a graphical DB schema?

Deliverables: Presentation

- **Presentation: Reflection**
  - Provide a good reflection on your project experience?
    - Describe what went well during your project, and what you should have done differently?
    - Describe lessons learned?

Deliverables: Presentation

- **Presentation: General**
  - Get the timing right?
    - Use ~15 minutes for the presentation?
    - Allow ~5 minutes for questions?
  - Reasonably balance your presentation among team members?
  - Provide presentation slides in your Google team directory?

Deliverables: Timeline

- **Timeline**
  - Clearly indicate each team member's contribution to your project during each week of the semester?
  - Keep your Timeline current through the Dean's Date?

Deliverables: Grader's Guide

- **Grader's Guide**
  - Describe what your system does and how to get it to do what it does?
  - Did you strive for and achieve:
    - Completeness?
    - Correctness?
    - Clarity?

The most important doc

Deliverables: Grader's Guide

- **Grader's Guide: Completeness**
  - Begin your document with an overview of your system?
  - Organize your document as use cases?
  - Provide a table of contents listing your use cases?
  - Compose your first use case (or some introductory text) to describe how to access or install your system?
  - Include use cases to cover all (or most) of your system functionality?
  - Provide a section at the end that describes additional system functionality?

Deliverables: Grader's Guide

- **Grader's Guide: Correctness**
  - Populate your system's database to contain the data required by the use cases?
  - Compose your use cases such that they can be executed sequentially **by each grader**?
  - Compose your use cases such that they can be executed sequentially **among graders**?
    - Instruct the graders to contact you for a database reset (if necessary)?

Deliverables: Grader's Guide

- **Grader's Guide: Clarity**
  - Compose your use cases such that the graders can understand them?
  - Compose your use cases such that they're concrete and specific?
  - Use screen images effectively?

Deliverables: Product Eval

- **Product Eval**
  - Include these sections:
    - Testing?
    - Eval by users?
    - Eval by experts?

Deliverables: Product Eval

- **Product Eval: Testing**
  - Answer the question "How well does the system work?"
  - Describe how you tested your system (see next slide)?
  - Describe the results of your testing, that is, which parts of your system work well and which do not?
  - List all known bugs?

Deliverables: Product Eval

- **Product Eval: Testing** (cont.)
  - Describe your:
    - Internal testing?
    - White box external testing?
      - Statement testing?
      - Boundary testing?
    - Black box external testing?
      - Use case testing?
      - Stress testing?
    - Test automation?

Deliverables: Product Eval

- **Product Eval: Eval by Users**
  - Answer the question "How well does your system meet the needs of its users?"

Deliverables: Product Eval

- **Product Eval: Eval by Users** (cont.)
  - Conduct interviews with typical users?
    - Compose a task list (maybe abstracted from use cases) and provide it in an appendix?
    - Present your task list to typical users?
    - Observe your users as they used your system to perform the tasks?
    - Encourage your users to talk aloud while performing the tasks?
    - Take thorough notes?

Deliverables: Product Eval

- **Product Eval: Eval by Users** (cont.)
  - Provide a summary of the interview notes?
  - Provide your detailed notes from each user interview in an appendix?
    - Tell us how many user interviews you performed?
  - (Optionally) Conduct surveys or distribute questionnaires?

Deliverables: Product Eval

- **Product Eval: Eval by Experts**
  - **You assume the role of expert!**
    - Maybe with help from Princeton's *User Experience Office*

Deliverables: Product Eval

- **Product Eval: Eval by Experts** (cont.)
  - Perform a thorough heuristic evaluation of your system?
    - List the 10 Nielsen evaluation categories, with positive and negative comments in each category?
  - Compose your heuristic evaluation to be specific to your system?
  - (Optionally) Perform cognitive walkthoughs of any parts of your system?

Deliverables: Project Eval

- **Project Eval**
  - Include these sections:
    - Project experience?
    - Technical issues?
    - Acknowledgements?

Deliverables: Project Eval

- **Project Eval: Project Experience**
  - Reflect upon your project experience?
    - Describe some positive and negative aspects of your project experience?
    - Describe what you learned from your project experience?

Deliverables: Project Eval

- **Project Eval: Technical Issues**
  - Describe some technical problems that you encountered and how you solved them?

Deliverables: Project Eval

- **Project Eval: Acknowledgements**
  - List (and briefly describe, as necessary) the pre-defined software that you used?
    - PostgreSQL, Flask, Jinja2, jQuery, React, Bootstrap, Render, Heroku, …
  - Cite the major sources of information that you used?
    - Particularly helpful web pages, tutorials, books, ChatGPT or other LLMs, …

Deliverables: Source Code

- **Source code**
  - Extract your code (and only your code) from your GitHub repo?
  - Place your code in a src directory (or a src.zip file) in your project directory?

Agenda

- Course summary
- End-of-semester schedule
- Project deliverables
- **Project team meetings**