Project Details

March 15, 2016

The goal of the final course project is to give you a flexible opportunity to demonstrate the web application software engineering techniques you have learned in this course. By now you should have formed a final project team, proposed your final project idea, and received feedback on your project proposal.

We have created a GitHub repository for your project team. It is named "team<n>" (where n is your team number) and you should be able to see it when you log into GitHub. Put project related documents and code in this repository. (If, for some reason, you need to use a different GitHub repository for your project, please discuss this with the instructor after class.)

You will continue work on your final project by developing a detailed specification of your project and then creating three progressively complete prototypes of your project (two for the Sprint presentations and one for the final project demonstration to the course staff). You will develop your prototypes (including your final product) using an iterative development process loosely based on Scrum, a popular framework for agile software development.

Important dates

- **March 18** Project specification due (see the related section, below)
- **March 22 & 24** Sprint 1 presentation
  Deliverables due: source code, presentation documents, backlogs
- **April 5 & 7** Sprint 2 presentation
  Deliverables due: source code, presentation documents, backlogs
- **week of April 18** Final demo

Scrum

For this project, we want to you to use an agile iterative development process that resembles Scrum, but we require less requirements analysis and fewer process--related artifacts than you would produce during a typical execution of Scrum. Each iteration in Scrum is called a *sprint*. The overall set of features to be developed for your product is called the *product backlog*. The set of features to be developed for each single sprint is called the *sprint backlog*.

The goal of an iterative development process is to define firm, precise goals for each short sprint, but allow you to evaluate your progress and adapt your goals as needed at well-defined time points at the conclusion of each sprint. The firmness of the goals during each sprint allows your team to work rapidly toward project goals during the sprint, without your individual team members needing to coordinate

each implementation detail with each other during the sprint. The flexibility of project goals between the sprints allows your team to evaluate its progress and (re-)define an appropriate set of goals to be achieved for the overall project and in the next sprint.

Based on this approach, you should avoid redefining your project goals during a sprint but may change your project goals at the conclusion of each sprint, based on the progress you just completed and the unforeseen problems you encountered.

For each sprint your team must define a single person to be the *product owner*. The product owner is accountable for coordinating the team's efforts and interpreting and prioritizing the set of project goals for each sprint. For this project we require you to rotate the role of product owner among your team members for each sprint, until all your team members have been the product owner for at least one sprint.

Project specification (due March 18th at 11:59pm)

The goal of the specification is to clarify and precisely define the functionality of your final project, as well as define the features you will complete during your first sprint. In the Scrum terminology, then, your project specification should describe your product backlog and first sprint backlog in great detail.

Your project specification should include:

- The *product backlog*: a complete list of all functionality (i.e., the actions) of your project, and an English description of each action. We strongly suggest that you organize these features into groups/modules based on related functionality.
- The first *sprint backlog*: a complete list of the functionality you will complete during your first sprint, and how that work is allocated among your team members.
- The name and Andrew ID of the *product owner* for the first sprint.
- A complete implementation of the data models used by your application. This may be written in as Django models, SQL, or some equivalent style implementation if you use another framework.
- A complete set of drawn wireframes or HTML mock-ups for your application, for all non--trivial views within the application.

From your English description of your actions and your wireframes/mock-ups, it should be clear how a user can navigate your site and what features are available from each page. A typical specification is 2--3 pages plus whatever code is necessary for your models and wireframes/mock-ups.

Organize files within your team repository however you want to organize the files for your overall project, but place any specification--specific documents (such as your product backlog and first sprint backlog) in the root directory of your team's repository.

Sprint presentations (March 22nd & 24th and April 5th & 7th)

For each sprint conclusion you should prepare a short (<10 minutes) presentation of your progress. We will assign each team a presentation location and time. In this presentation you should explicitly demonstrate your site, showcasing the features you have completed so far. The prototype you demonstrate should be a complete, high-quality and cohesive implementation of some subset of your overall product features.

Your presentation should additionally include:

- a short background of what your project is and who is working on it,
- your original goals for the just-completed sprint,
- what you have completed during the sprint,
- a discussion of your progress and the problems you encountered, and
- a brief description of your goals for the next (upcoming) sprint.

You should make sure the following is committed before your presentations:

- source code for your project
- any presentation materials (slideshow, etc.)
- the sprint backlog for the next sprint to come, including the name and Andrew ID of the product owner for that sprint

Final demo (week of April 18th)

Your final project demonstration is not a presentation; it's a demonstration of your project's features, an evaluation by the course staff of your features, and a code review of your final product. You should not prepare a presentation for your final demo; you will merely guide us as we examine the key features of your site and evaluate your implementation. A typical demonstration and code review will be about 30 minutes.

For your project demo, you must deploy your project on a server accessible by the course staff from their own laptops. Your deployment might be local to your own computer --using Apache or Nginx or a server other than Django's development server --or might be a cloud service such as Google App Engine, AWS, or Heroku. In any case, you must ensure that your site is accessible to remote clients (i.e., not just your laptop or server) before the project demo so that course staff do not encounter problems accessing your site during the demo.

If you are unable to demonstrate your course project at the scheduled time, you may discuss with the professor the possibility of demonstrating your project the following week with a reduction in your project grade.

Evaluation

Even though this is a team project, you will receive individual grades for the project. Historically, for most teams all members of the team receive the same project grade unless there are clear differences in the team members' contributions to the project.

Your final project grade will be based primarily on your final product but also on the artifacts you produce during your development process, including your proposal, specification, and non--final sprints.

To help our evaluation, all team members must use Git as a version control system and actively contribute to the team's GitHub repository. Each commit message should describe your contribution to the project as well as (if necessary) the context for the work you just committed. If your team uses techniques such as pair programming, your commit messages must explicitly describe the active contributors and you must rotate the active member of the pair so that all of your team members contribute fairly (and visibly) to the project. If you are using a repository other than the team's GitHub repository that we have set up for you, you must discuss this with the instructor.

Your project grade will be based on a broad range of criteria, including at least:

- Functional correctness and functional design, including correct validation of all user input, positively demonstrating an understanding of concurrency, management of web application state, usability, and avoiding basic security flaws such as XSS, CSRF, and SQL injection attacks.
- Your development process, including your use of the Git version control system, substantial active contributions from all team members.
- Use and positive demonstration of understanding web framework tools, including models and ORM interactions, forms and model forms, templates and the use of view--based template languages, sessions, and possibly cookies.
- Software design principles, including modularity and information hiding, separation of concerns into appropriate files and modules where appropriate, and demonstrating an understanding of MVC frameworks.
- Overall complexity of your project, including the overall size and sophistication of your project idea, diversity of implementation tools and technologies, diversity of external libraries and data sources, and deployment-related issues.`