sprint #1one

Prof Bill - Apr 2016

Project Odyssey. Sprint #1. Let's code already. (™)

- ★ Starts: **Fri Apr 22** (end week 4)
- ★ Ends: **Mon May 9** (start week 7)

That's 17 days, 3 weekends.

Coding

We are going to code individually on this sprint. We'll split into our teams and each team will build similar areas of the User Experience (UX).

While coders will work individually, you are **strongly encouraged** to collaborate with your fellow team members, your counterpart working on similar stuff in the other team, hey, any warm body in 495. Also, if you snarf a code snippet from google or StackOverflow, that's cool. Please note the snippet's origin (URL) with a comment in your code. I **only prohibit** copy-pasting each other's code.

Here's my rationale for this approach:

1. We have an extraordinary group of students: capable and positive (and amazing)
2. The downside of this approach is redundancy. The upside: we'll be able to compare different implementations looking for the best code and the purest application of Rails idioms.
3. Development is fairly compartmentalized at this point. The Home guy won't really be impacted much by the work of the Schedule guy, for example. We'll tackle authorization and integration issues in the next sprint. Coders will focus only on their screens for now.
4. There are still many unresolved issues, some chaotic times await us. So, let's run, make some mistakes, and correct them in the next sprint.
5. It will be a good to code up some Ruby on Rails on your own.
6. It will be a good to collaborate in your teams to overcome tech problems that you may be having individually.
7. In Sprint #2, we'll do some pair programming as we integrate code, incorporate authorization throughout, and adjust to any UX changes.

Deliverables

Your deliverables for Sprint #1 are:

- ➢ Your **coding assignment** (all UX screens plus the notes above) to be completed and pushed up to your team's Github repo.
- ➢ Your code must be excellent. It must be **commented**, mostly block comments for long or complex sections of code.
- ➢ You must have Rails **test cases**, ala our tutorial. (not too crazy though)
- ➢ I will complete a **code review** at the end of Sprint #1 with each team member.

Finally, each team will demo their system at the end of the sprint, in class on May 9.

Some Details

Some details in a nerdy bullet list:

- ➢ Please use the **UX V1.0 document** in the #ux channel as your guide. If you have questions about a screen or its behavior, please ask any member of the Bootstrap Cell (Nick D, Emily H) or a product owner (Prof Bill, Jennifer D).

- ➢ There are **5 dev areas**, following the major areas in the UX: Login, Home/Pickup, Schedule, Report, Admin. Each team has **5 coders**, one per area. We'll negotiate team member responsibilities at the beginning of the sprint.

- ➢ The Home, Schedule and Report screens are to be developed **without any authorization**. Don't worry about users logging in or tracking them for now. If you need to, then assume the user level is "normal" and not "entry" or "admin".

- ➢ The **Login and Admin** screens are "special". They will implement screens **and** share authorization code as sort of a proof of concept.
    - The Login area has only one trivial screen (the first one). So, the Login developer will implement an authorization protocol for users logging in for a Haymaker session.
    - The Admin coder is responsible for UX screens **and** for working with the Login developer to incorporate his/her authorization scheme into the Admin screens.
    - One team will implement the "roll your own" authorization scheme shown in the tutorial. The other will implement a gem like Devise or CanCan.

- ➢ We'll all rely on some centralized and shared code. And we're on it. Data model => Arch Cell. Common CSS => Bootstrap Cell. Github => Git Cell. I don't think any of these areas will be completely ready on day one. So, stay positive. Communicate. Help out.

- ➢ Use the #sprint1 channel to share!

Scrum stuff

Recall the 3 roles in the Scrum approach:

- ➔ Product owner (Bill, Jennifer D)
- ➔ Scrum master (our team leaders: Rachael R, Nick O)
- ➔ Team members (all y'all)

Our product backlog items (dev areas for the sprint) are defined in UX V1.0 plus these notes.

During Sprint #1:

- ❑ Teams will meet at the beginning of each class to discuss the issues of the day. How's it going? Issues? Progress? Things we need to discuss with Prof Bill. Etc.
- ❑ The team leader will ask each coder his/her **spotlight status**:

  Green = stability and good control over project

  Yellow = caution and suggests steps for regaining control

  Red = a crisis that requires attention

  Or, more colloquially: green good, yellow worried, red floating at the bottom of the pool

- ❑ We'll work together to get a team member reporting a yellow or red status back on track!
- ❑ Important - It is not the team leader's job to "be the boss of you". (Hey, that's my job!) I expect that each team will power through their issues by working together. If we have some issue or conflict that needs to be resolved, I'll step in and help.

Dev assignments

Here are the Sprint #1 assignments.

UX/Dev Area	Team #1 DramPhDZ	Team #2 Code Weas
Login	Eric Z	Chase S
Home	Mark P	Kyle L
Schedule	Bilal A	Nick ▫
Report	Rachael R	Gerardo P
Admin	Bill M	Charlie M

Very exciting!
thanks… yow, bill

PS - The spotlight status idea is borrowed from:

- I originally heard about the idea from Alan Mulally, the CEO of Ford: Alan Mulally's Management Secret: Peer Accountability. BTW, Mulally's book, American Icon, is outstanding: www.goodreads.com/book/show/13132620-american-icon

- My wording is from a health care site dealing with asthma: Stoplight (Red-Yellow-Green) Tools for Patients with Asthma or Diabetes