Milestone One

Primary Objectives:

1. Form your development team
2. Learn to work as a team
3. Create team documents with a common format
4. Propose 3 possible projects

Overall Requirements:

- [ ] Team name
- [ ] Team motto
- [ ] Team logo
- [ ] Team letterhead
- [ ] Team business cards
- [ ] Résumé for each team member
- [ ] Workbook skeleton
- [ ] Team schedule, including meeting time with advisor
- [ ] Slack team
- [ ] Git repository for team project
- [ ] Three team project ideas

Grade sheets (ods)

Introduction

This milestone will get your team organized and begin the process of working together as a group. Don't underestimate the difficulty of doing simple things **as a group**. Communicate, communicate, communicate!

In step 6 you will create a Git repository for your team. This is where all the documents created during each Milestone should be placed (neatly and in appropriate folders). A note on file formats: you should default to plain text files, such as Markdown or HTML, for everything. The resumes (e.g. Word) and logos, etc. (e.g. png) can and should be in the appropriate formats, but if you're only writing text, put it in a Plain Old Text file. Source repositories, such as Git, are for source code, which means a text file. Of course, html and Markdown are plain text and also give you formatting options if you so desire.

The 3+Branch, with Pull Requests Git Workflow that we'll be using all term isn't introduced until Milestone 2. For now you can either learn the workflow ahead of time, or do the less-than-desired method of manually transferring files to the repo owner, who should add/commit/push them. **Under no circumstances should all team members have admin or write access to the team repository.** That's a great way to screw up your repo. Last term we all had our own repositories and could control the entire push/pull cycles. Imagine 4 people randomly adding, committing, pushing and pulling!

Tasks:

1. Select a team name. This is how your team will be identified for the next 22 weeks, so give it some thought. Also create a team motto, something short like “Success through engineering”. You will need to create a logo and a letterhead for your team. Also prepare business cards for each member.

2. This section of the milestone will include a resume for each member of the team. **All of the resumes should look the same, contain the same major headings, and have the same format including font size and type, and order of sections.**

   We'll leave it up to your group to decide on the best format, but these must be good ones that you'd be proud to submit with a job application today. There are many, many good resources online for what should go on your resume and what it should look like. One page is best but you can prepare a second one if you wish that has more pages.

3. Acquire and assemble a 1.5" three ring binder with dividers for each work product. Include:

   - Scrum Reference Card: place a copy in the front of your notebook
   - Have Tab separators for each Milestone and Sprint (8 dividers)

- Each Milestone section will contain (in this order)
  - Milestone grade sheet (**on top and completely filled out before your meeting**)
  - A printout of the milestone (i.e. this page -- it avoids confusion during meetings.)
  - Work products **in the order they appear in the milestone**. Include all items that can be printed out (e.g. logo, resumes, business cards, ... for this milestone).

4. Create a schedule for your team that lists **five hours of meetings during the week**. Although you will not use all of them, it is good to know they are available during “crunch time”. There are several specific meetings described in the SCRUM tutorials **that are required**! Define some of your schedule times for the five defined SCRUM meetings. Standup meetings occur at the beginning of each class, and the scheduled meeting with your team instructor will function as the sprint review meeting. Schedule time for the remaining meetings into your team’s weekly schedule. See the SCRUM reference card for a quick overview of the meetings.

   As you'll recall, face-to-face meetings are regarded as the best for agile teams so strive for that. If your team members' schedules do not permit this then block out time for online meetings. **A lack of communication, primarily due to team members not being available for Coordination, Collaboration and Conclusion style meetings, is very high on the list of reasons for poor team performance and low grades.**

   As just mentioned, you will need to schedule a fixed time to meet with your project advisor during the week. As in the real world, you will need to find a time that is convenient for them. Not one that works for your schedule. Your best bets are to look at times that match already scheduled office hours or are next to existing courses.

5. Create a Slack Team for your team. Name it appropriately using your team name. There is no need to invite Scot to your Slack Team. Dr. Morgan may want to be invited; please ask her at your first meeting if she is your advisor. Only with her permission may you change this to a Discord server.

6. Create a GitHub repository for your team. This will be the repository for **everything** for your team for the next 22 weeks! Name it appropriately with your team name. The owner of this repository will have additional duties and management oversight of your team's documents and code, including Azure integration. It is probably a good idea for this person to be the same person you might envision as your Team Lead, or the person with the most Git experience. Add a contributors file and a README that describe your team and this project. Have all team members (and your advisor) **fork** the repository and then initialize a local copy (i.e. clone their own fork). We will be going over this process extensively in class.

7. Lastly, come up with at least 3 candidate project ideas for your team project. For each one, write a one sentence (or very short paragraph) description. Be prepared to discuss these in more depth at the meeting with your advisor. Here are some high-level requirements for your team projects:

   - Must be based primarily on the architecture we learned last term: ASP.NET MVC 5 + SQL Server, both deployed to Azure. You are encouraged to add other resources to this as needed. Some examples: other Azure or AWS services (i.e. file storage), email (SendGrid), SMS (Twilio).
   - Must be something interesting and useful; it should not be boring or routine.
   - Must use an external API for at least one major feature. By external API I mean something like: Lyft, Twitter, Instagram, Google, YouTube, US Government. And I don't mean something as simple as dropping a Google Map element on your page and calling it good. I really hope one or more groups uses one of these (please, please?): IBM Watson, Microsoft Azure Cognitive Services, Google Cloud AI, Amazon Machine Learning on AWS.
   - At least one major feature must involve an “algorithmic component” developed by your group.
   - It's OK to take on a higher risk project. Personally I think it's better to try to do something more interesting and potentially more difficult and accomplish less, than to have a really polished project that's boring and easy to do. That being said, you need to manage the risk. If your project depends very heavily on X, then you better put a whole bunch of time into learning and figuring out X during the inception phase. If depending on X forms part of your architecture, you'd better prove it early as the book says.