Team ATD
Mona Assarandarban, Amirreza Barin, Jessica Greenling, Nicholas Nelson, Victor Szczepanski

Process

Our main process is centered around employing the Scrum methodology. We set our Sprint period to two weeks, with each deadline coinciding with each Demo date. At the beginning of every Sprint, as a team we will estimate the time for the tasks in our backlog and select the tasks we believe can be completed during that Sprint. As such, our proposed timeline may vary from the actual timeline as we become more familiar with the difficulties inherent in the project. We will have weekly Scrum Meetings so that each member can explain what s/he has done, and discuss what difficulties any members are facing.

To facilitate using Scrum, the team will use Trello as a Scrum board. We will track our backlog, current Sprint log, and time spent using Trello. Additionally, we will use the Scrum for Trello plugin to facilitate tracking estimated time and consumed time; we can then leverage the Burndown chart to track project velocity.

Some of our tasks can be completed in parallel. For these tasks, we will split into two sub-teams and work on them independently. However, some tasks act as bottlenecks that prevent us from working on other tasks until they are completed. In these cases, the teams will regroup and complete the bottleneck task. Then, as soon as the bottleneck task is complete, the teams will split up again. The entire team will be responsible for all project level documents.

We have not decided on a specific testing framework, but we are investigating using PyUnit, Behave, or the provided unittest library. We will not adhere strictly to Test-Driven Development practices, but during development we will integrate unit tests. That is, we will not wait until the system is complete to write unit tests, but will integrate them in each task.

Finally, to ensure we have a common development environment, we will use an Azure cloud server to develop and test our code on. We will use our team SVN repository for source control and coordination, but all testing will be carried out on the Azure server. This does not negatively impact our software's deployment environments, however, since the Azure server will run CentOS with no special dependencies aside from Docker.

Timeline

We will organize our timeline into 2 week sprints, with the deadline coinciding with each Demo date.

Sprint 1: 2/19/2015 - 3/17/2015

- Create Architecture diagram
- Create Class diagram
- Create User Stories
- Create Domain Model

Learn about Docker - How to install, how to run, how to manage containers, how to select distros.
Set up standardized environment for team development using Azure cloud server.
Set up Trello board to use as Scrum board.
Create Docker Invoke - a module that can take a Docker file, create a container, and run the instructions in the Docker file.

Sprint 2: 3/17/2015 - 3/31/2015

Create Testing Description
Define repo layout - a standard location in a repository where the unit tests can be found.
Create DF gen - a script that will generate a Docker file and calls Docker Invoke to run the new Dockerfile
Begin Container Manager - a module to manage the running docker container

Sprint 3: 3/31/2015 - 4/14/2015

Finish Container Manager
Investigate transferring docker output to caller (Output Manager / Container Manager)
Begin DF output manager - a module to manage transferring the output from docker into the calling environment
Begin DF output parser - a module to parse and interpret the output from DF output manager

Sprint 4: 4/14/2015 - 4/28/2015

Finish DF output manager
Create Notifier - a module to notify the responsible parties of test results (User and Repo Developers).

Sprint 5: 4/28/2015 - 5/2/2015

Finalize documentation

During Sprint 1, we will work on all tasks as a single team. All team members will be responsible for the creation of documents and the Docker Invoke module. All project documentation will be created by all team members. Code documentation will be the responsibility of the relevant developers.

Starting with Sprint 2, we will divide our team into two sub teams: A and B.
Team A will be responsible for DF gen, while Team B will begin development of the Container Manager. Since Container Manager is a bottleneck, once Sprint 3 begins, Team A will assist Team B with finishing Container Manager.

In Sprint 3, after Container Manager is complete, we will again split. Team A will be responsible for DF output manager, and Team B will be responsible for DF output parser.

In Sprint 4, each team will finish any outstanding work and regroup to finish Notifier. That is, every team member is responsible for Notifier.

In Sprint 5, every member will review final documentation.