Lab – Mock Sprint

# Summary

This lab will simulate sprint planning and execution, from start to finish, for a single sprint. The purpose is to make you more familiar with the process, in a hands-on fashion.

You are not expected to do any real coding for this lab, but you are expected to do everything else. We will skip the daily standup for this exercise since there's no real work being done.

Please read the entire lab before starting any work.

# Team Size

4 people

# Deliverables

1. Binaries built and released from VSTS
2. A sprint retrospective, shared with the class

# Requirements

## Form a team

1. Form a team based on the lab team size
   1. Try to work with different people than you have before
2. Create a team name
3. Send me your team name and all team member aliases using Skype
   1. I will create a team and area path for your team in VSTS
   2. The Area Path for your work items should be Cohort#\Labs\<LabName>\<YourTeamName>
   3. Cohort8\Labs\MockSprint\jeclANDhyde

## Perform sprint planning

Perform sprint planning, assuming a two-week sprint

1. Create a large hypothetical user story that has something do with software engineering.
   1. For example, "As a user, I can use an air traffic control service for my drone to ensure I don't crash into other drones"
   2. As a user, I can use this vending machine to order food/drinks depending on the current weather.
2. Think about what user stories are required for this to work. Use VSTS to create a work item for each user story
3. Break down the user stories into smaller tasks. Use VSTS to create work items for each task, using child tasks as appropriate.
   1. **You should be discussing business requirement tasks, not implementation details.**
   2. The Area should be Cohort#\Labs\<LabName>\<YourTeamName>
   3. The Iteration should initially be Cohort#\Backlog.
   4. This Area/Iteration combination is your product backlog.
   5. Fill out rough time estimates for each task as a team.
4. Write a query that gets a tree view of the user stories in your team’s area, along with all their child tasks, sorted by priority ascending.
   1. Save this query as a shared query so anyone on your team can run it.
5. Once the query works, prioritize the user stories, and tasks within the user stories.
   1. Remember, Agile is about delivering something that runs (even if it’s incomplete), then iterating on it.
6. Decide on your minimum viable product (MVP), and change the iteration for those tasks to Cohort#\Iteration1.
   1. This is your sprint backlog.
7. Distribute the tasks amongst your team.
8. Each engineer should create child tasks of their business tasks which include implementation details and technical requirements
   1. The child tasks should each have a time estimate. The time estimates of child tasks should not be more than the parent task. If they are, discuss it with your team.

## Create team branch

1. Browse to <https://microsoftleap.visualstudio.com/_projects>. Click on the project for your cohort.
2. Click on Code -> Branches in the top ribbon
3. Select the repository for this lab
4. One person should create a new remote branch from master that matches your team name. This will be your team branch

## Create build automation

1. Click on Build and Release -> Builds in the top ribbon
2. One person should create a build definition for your team branch. Name the build <LabName> - <yourTeamName>
   1. You can just clone the build called “<LabName> - master” and make the appropriate changes (name, code source, etc…)
   2. Kick off the build manually to make sure it works before moving forward.

## Create release automation

1. Click on Build and Release -> Releases
2. One person should create a release definition for your team branch. Name the release <LabName> - <yourTeamName>
   1. You can just clone the release called “<LabName> - master” and make the appropriate changes (name, artifact source, environment variables, etc…)
   2. This release should automatically be created when a build completes, and the Dev environment should automatically deploy when the release is created.
   3. Kick off the build manually which will trigger the release when it completes. Make sure the Dev deployment completes successfully.
3. To see the released files, open a command prompt and run the following command
4. net use Z: \\leap.file.core.windows.net\test /u:AZURE\leap Y9m0DQpWqoAwxUWRFhsNxwieR4D9p/gUEX6sWN/E77rRw/INBkgcTlIwZDXCbVbR/WNyc1Y0ytUZXgVGfF02+g==
5. Browse to your released files
6. Open windows explorer and browse to the Z drive
7. Double click on the name of your release definition.
8. Double click on the name of your environment (Dev, in this case).
9. Double click on the release number (Probably Release-1, in this case).
10. Your build artifacts should be in this folder

## Create personal branches

1. In VSTS, each team member should create their own remote personal branch from the team branch. Name it something like <yourAlias>-dev
2. Each team member should clone the remote repo, which will get you a local master branch.
3. Each team member should create a new local branch from their personal remote branch. Give it the same name as your personal remote branch.

## Execute the sprint

1. Each team member should commit work into their local branch and push it to their personal remote branch.
   1. In this case, the work is just adding dummy files or making small changes to existing files. Nothing real, just mock work.
2. When ready, each team member should create a pull request from their personal remote branch into the team branch.
3. The team should perform code reviews to make sure the team branch stays healthy.
   1. The reviewers should not be the entire cohort but should instead be your team in VSTS.
   2. In this case, the code reviews are again not real, but it’s good to practice the process.

## Perform a sprint retrospective

1. Perform a sprint retrospective with your team. Each member should briefly discuss what went well, what didn’t go so well, and what can be done to make things better next time.
2. When all teams are finished, pick a team member to share the retrospective with the class. Five minutes max.