Question 1 Answer:

Each of the team members installed Crucible onto their own laptops. We downloaded the Crucible files, and then followed the instructions provided by Crucible site (https://confluence.atlassian.com/crucible/installing-crucible-on-linux-and-mac-298977373.html), where we verified Java installation, specified where to store Crucible data, and set up the required environment variables. We then launched a Crucible instance and accessed the web portal in a local browser. In the web portal, we created a user account, linked our repository, and started making comments on code. We could not link Jira into Crucible because we do not have Jira admin privileges.

During the process, multiple team members encountered the following issues:

1. JDK installation was not compatible with Crucible, and we had to install a previous release of JDK (1.8.x) which led to more problems such as OSX complaining about a newer version already being present. This of course wasted some time.
2. The instruction provided by Crucible was not very clear on known errors and did not provide the means to resolve some of the errors. One of our members did the setup, tried to launch a crucible instance, and the fisheye log file was nowhere to be found. We were not able to find the possible cause of this so we had to redo the whole setup procedure on his laptop again.
3. When adding the repository to Crucible instance, after entering the Github repo link, it shows lines of code as 0. We suspected that Crucible failed to pull down the code base because of a poor network connection.
4. We have seen one incident where the indexing step of the code base in Crucible takes forever. We suspected that it was also caused by inconsistent network connection.
5. In another instance, one team member was able to get the project loaded into Crucible, such that he could see all of the branches, but was unable to interact with it. In this case, the repository status was marked as ‘Error’, and was labeled as having 0 lines of code.

Crucible is ideally supposed to be hosted on a dev server so that each team member can to access the same instance and work collaboratively. However, due to the class format, we had to individually install Crucible locally and run multiple instances in order to conduct the code review. Some members were unable to set up a running Crucible instance, so they alternatively

used Github to conduct and submit their reviews. For the purposes of this assignment, the project manager then aggregated reviews and screenshots from Github and Crucible. It obviously was not ideal that both Github and Crucible were needed to conduct the review, but we incorporated Github to allow for more team members to participate. Comparing Github to Crucible, we did find an advantage of Crucible over Github. In Github, review comments tend to be injected into pull requests at the commit level, whereas in Crucible, comments can be added to the code at any time, making it simpler to browse branches as a whole. However, reviewing Crucible from the perspective of the development process, because Github already allows for review of pull requests—and because we are already using it for version control—we found that adding Crucible into the tool suite created an additional complexity that wasn't worth the reward.