**Sprint Goals**

* Develop an understanding of the product that the client wishes to make
  + By the end of Sprint 1, we will have a fleshed out list of functional and nonfunctional requirements based on the client’s initial backlog information, flagging functionality that the client seems unsure of. The related requirements will be grouped together into potential epics, and each requirement will be graded with a priority by the entire team so that a roadmap can be established.
  + By the Sprint 1 meeting with Product Manager (the TA), we will have formed questions to ask to fill in gaps in our understanding where we find the backlog provided by the client to be unclear.
  + By the end of Sprint 1, we will have formed a first draft use case diagram for the product that will be reviewed by Product Manager and revised accordingly.
* Develop an understanding of Prattle
  + By the end of Sprint 1, we will have a first draft UML Class diagram of the provided Prattle code so that it is easy to see how the existing classes and functions interact with each other and where we can inject our own logic to turn it into our Slack product.
  + By the end of Sprint 1, we will have tested the provided Prattle code with Junit tests to reach the required SonarQube coverage and merge the code into master branch of our GitHub repository
* Create our first Backlog in Jira
  + By the end of Sprint 1, we will have created our first Backlog in Jira. Specifically, we aim to write the user stories which capture all the features of instant messaging between individual users and groups so that we will be set up for Sprint 2 accordingly. This will necessitate discussion and design for the instant messaging architecture that will be the centerpiece of our product’s functionality.
* Build status notification on Slack
  + By the end of Sprint 1, we will be getting notifications on slack regarding the build failures or success when the code is pushed to GitHub.
* Automatic branch creation on GitHub
  + By the end of Sprint 1, if we create a ticket on Jira and assign it to someone, automatically a branch will be created on GitHub with the Jira ticket ID/feature name.
* Smart Commits
  + By the end of Sprint 1, if someone commits to the branch with Jira ticket ID in commit message, the ticket will automatically be closed in Jira.