**5.30 Personal Class Notes**

Track everything you do for this class on **Toggl**

**Travis-ci**: Test driven development. Every time you push your code to the central location, you can check to see if it is working. The tests have to pass on the Travis-ci and you have to show it to clients. You need to have a code coverage of at least 95%.

**Travis-ci**: is a continuous integration tool, so every time you put your code on GitHub, it tests it

**Sonarcloud** allows us to find any security vulnerabilities

**Code Smells** – code works but it might not be an easy read

**TDD** – test driven development

* You write tests for code
* Make sure that test fails
* Write just enough code so that it passes
* Incremental steps and write a new test
* Create unit tests for your code

**Unit testin**g - Does this unit of code pass the test

**Coverage** – does this line run. Percentage of code lines that works. We will be focusing on line coverage. But there are two types: line coverage and branch coverage

**Sonar cloud** just checks for bad practices in coding style

**Git commands in visual studio**

* View > command palate > git clone
* After you commit changes, push them into your local branch

**Remote** – where a repository lives. Remotes will contain repositories. Github is a remote

Clone the account and change the main remote to the central repository your group made

From feature to develop branch to master

**Scrum master** delegates tasks for each member for the sprint backlog