# **Special Topics** (CSE 490-R)

## Training minutes – Weeks 5-6

|  |  |
| --- | --- |
| **Date of meeting:**  **Instructor:**  **Student:** | May 29, 2022  Bro. Clements  Guillermo Oliva |

* **Prompts from assignment:**
  1. What advantages as a developer does TDD provide?
     1. We only write code that is needed.
     2. Modular design
     3. Easier to maintain
     4. Easier to refactor
     5. High test coverage
     6. Test document the code
     7. Less debugging

**Source**: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-test-driven-development-tdd/>

* 1. What steps do I do to implement TDD?
     1. Read, understand, and process the feature or bug request.
     2. Translate the requirement by writing a unit test. If you have hot reloading set up, the unit test will run and fail as no code is implemented yet.
     3. Write and implement the code that fulfills the requirement. Run all tests and they should pass, if not repeat this step.
     4. Clean up your code by refactoring.
     5. Rinse, lather and repeat.



* 1. What tools or software can I use to do unit testing?
     1. nvm (Node Version Manager) for Node.js and NPM: NVM allows you to run the Node.js version you want and change it without affecting the system node.
     2. npm libraries for development:
     3. Jest for unit tests
     4. ESLint for linting
     5. Prettier for formatting
     6. Husky and lint-staged for pre-commit Git hooks

**Source:** <https://developer.ibm.com/articles/5-steps-of-test-driven-development/>

* **Notes taken from lecture**
  1. The design element is crucial any software development project.
  2. Many tools are available for design purposes:
     1. UML (the most popular)
     2. Class Diagram
     3. Sequence Diagrams
     4. Component Diagrams
     5. Entity Relations
  3. Design Patterns are also useful:
     1. Behavior
     2. Structural
     3. Creation
  4. Other tools include:
     1. Process flow
     2. CRC
     3. Layering
     4. Data Structures
     5. Pseudocode
     6. Database Definition
     7. Workflows
     8. Petri Charts
     9. Formal Language
     10. Viewpoints
  5. Creation of WBS:

A picture containing text

Description automatically generated

* 1. Samples of WBS:

**A picture containing table

Description automatically generated**

* 1. Functional SW Characteristics:

**Diagram

Description automatically generated**

* 1. Software quality:

**A picture containing timeline

Description automatically generated**

* 1. Quality attributes and models (there is more on this topic from the Software Engineering II class):

**Diagram

Description automatically generated with medium confidence**