Example of working agreement template created in a virtual whiteboard

1. Frequency, time, and location
    a. Sprint planning: Wednesday @1:30pm, https://meet.google.com/bxu-tgos-ueb/ or offline: AnnexLibray
    b. Daily Scrum: Every day, Monday-Friday, https://meet.google.com/bxu-tgos-ueb
    c. Sprint Review: Wednesday @1:00pm, https://meet.google.com/bxu-tgos-ueb
    d. Sprint Retrospective: Friday @8:00pm, https://meet.google.com/bxu-tgos-ueb
2. Communication Channels: slack or anything specified by the requirement in the sprint
   Slack channel:
   https://join.slack.com/share/enQtNzg0NjE1MjU5ODM1My01NDFjOTY3ODQ4Yjk3NTIxMzBmYTYwNjkxMmRjYWFhOGNkYmUwODJhYzJjNzYwODAxMzgyZTY2YmQ1NzQ0NThj

3. Tools
    a. Jira: Project management and tracking
       Jira Board link:
       Sprint one completed Stories Report:
       https://tamu-team-office-tracker.atlassian.net/jira/software/projects/SCRUM/boards/1/reports/burndown?source=sidebar
       Sprint Burnup Report Link:
       https://tamu-team-office-tracker.atlassian.net/jira/software/projects/SCRUM/boards/1/reports/burnup

    b. Github: code version control URL:
       https://github.com/et-tran50/CSCE_606_Office_Hours_Tracker/tree/main

4. **Definition of Done (the criteria that must be met for a task or feature to be considered "done.")**

   The criteria that must be met for a task or feature to be considered "done" typically include the following:

   1. The code is fully developed and adheres to coding standards.
   2. All unit tests and automated tests have passed.
   3. The feature has been reviewed and approved by peers.
   4. The functionality has been tested and confirmed to work as intended.
   5. Documentation is updated, if necessary.
   6. The code is merged into the main branch, and all build pipelines have passed.
   7. The feature is deployable to the production environment.

5. **How the team approaches work:**

   Our team adopts a collaborative and iterative approach. We break tasks into manageable units, prioritize based on project needs, and assign ownership accordingly. Clear communication and regular check-ins are central to our process, and we focus on collective problem-solving. Peer reviews are part of our routine to ensure quality and promote continuous learning. We work in sprints, staying flexible and adaptable while focusing on delivering value with each iteration.

6. Where documentation and artifacts will be stored (Determine the location of important documents like user stories, sprint plans, and technical documentation.)
   a. Shared Google drive: For documentation still requires heavy editing https://drive.google.com/drive/u/3/folders/1hMrZdfL0pOiByzCx4NsOfpCCc5e9Kj3B

   b. GitHub repository: For stable documentation

7. Additional Agreement
   a. **How should we handle situations when a team member is unable to complete their task on time?**

      When a team member is unable to complete a task on time, we should respond with understanding and open communication. Encouraging them to share their challenges fosters a supportive environment for discussing obstacles.
      Once we grasp the situation, we can brainstorm solutions together, such as redistributing tasks, offering help, or adjusting deadlines.
      Maintaining transparency about our progress and difficulties is essential. Regular check-ins during stand-ups or team meetings will help ensure everyone feels supported and on track. By fostering a culture of collaboration, we can help each

other meet our goals and keep the project moving forward.

b. **Should we push code directly to the main branch on GitHub, or use separate branches with peer reviews?**

   We should use separate branches with peer reviews rather than pushing code directly to the main branch. This ensures that any changes are thoroughly reviewed and tested before being merged, reducing the risk of bugs or issues affecting the main codebase. It also promotes collaboration and helps maintain code quality across the team.

c. **Should we deploy to Heroku after each push to GitHub, or designate someone for Heroku deployments?**

   It's best to designate someone for Heroku deployments rather than deploying after each push to GitHub. This allows for more controlled and stable deployments, ensuring that only tested and reviewed code goes live. It also avoids the risk of deploying incomplete features or potential bugs and ensures consistency in the deployment process