

Working agreement

1. Meetings

- In-person meetings will be held twice a week on Wednesday and Thursday.
 - On Wednesdays, the meetings will be held from 11:00 AM to 12:00 PM
 - On Thursdays, the meetings will be held from 5:00 PM to 6:00 PM
 - Meetings will be held at BV on Wednesdays and IC on Thursdays
- Additional and more extended meetings will be held on Discord
 - These meetings will allow members to settle in a more comfortable environment
 - As a general rule of thumb, these meetings will be expected to last longer to ensure that more work is completed
- Failing to attend a meeting will result in a food strike (ie. the person failing to attend will be expected to provide some form of food for the next in-person meeting)

2. Communication

- Additional work and team communication will be done over Discord or Skype.
 - Dedicated channels will be created to address various development processes (ie. a channel for bugs, another for feature development, etc).
- Team members will be notified of any messages and are expected to be in constant communication to reduce lapses in development
 - Members will be expected to respond at the end by the end of the day (at the very least) barring extenuating circumstances

3. Development

- All team members are expected to be working on something relevant at all times
- The distribution of work should be decided by scrum poker so that an even amount of work is allotted to each member of the team
- Team members are expected to comment their code appropriately
 - Over-commenting and under-commenting are both highly discouraged
 - There should be a comment to cover every non-trivial part of the code (ie. there can be one comment to cover multiple lines of code)

4. Version control and commits

- Each commit is required to have a concise message that reflects the changes between the previous and current commits.
- Commit messages will follow this format to make them simpler to read
 - The message should begin with the type of change and be followed by a description of the change without getting too technical
 - For bugs: it is important to state add "[BUG]" at the beginning of the commit message, then explain the problem that was fixed

- For development: begin with “[DEV]”, followed with the changes that were made
- Major commits (such as feature completion) should begin with [DEV] [FEAT] to specify a more specific milestone
- Commits should be done frequently at a reasonable pace at the discretion of the individual, but they should always compile
- Complete code is expected to be submitted at least 4 hours before any due dates or deadlines so that they can be reviewed and tested by other group members before being pushed to one of the main branches

5. Branching strategy

- We will be using the release branching strategy where two main branches will be maintained:
 - Integration testing (or trunk) branch: completed code will be pushed onto this branch to test how it integrates with the rest of the system
 - Complete (release) branch: once code is tested within the system in the integration testing branch, it will be pushed onto this fully production ready branch
- In addition to the two main branches, there will also be various feature branches
 - Feature branches will push into the main trunk branch once the feature is ready for integration testing
- Code will only be pushed to the complete production branch by the scrum master after it has been reviewed and tested in the integration testing branch
 - This will prevent “breaking the main release branch”
- Testing will occur on both branches periodically
 - The majority of testing will occur on trunk branch (that’s what it’s for) to ensure the code pushed to the release branch is fully functional and complete

6. Testing

- JUnit test cases will be developed for general testing (ensure program works as expected)
 - Features are expected to work as expected on feature and integration testing branches
 - Both white and black box testing will be employed for individual modules and features
 - The following types of test cases will be written:
 - Happy path
 - Edge cases
 - Destructive (invalid inputs)
 - Test cases will attempt to cover every situation considered valid
 - eg. different users sharing accounts, changing users at an ATM etc.

7. Contingencies

- If a team member is unable to further contribute to the project (ie. someone drops the course or they get hit by a bus), the professor shall be notified immediately by e-mail or in person