

Team 30 (Nerdy-Thirty) Rules

1. Team Purpose and Objectives

Our team aims to develop a successful software project while learning and working under common SWE practices. Creating this project brings us an opportunity to not only learn efficient collaboration skills, but to also provide ourselves an idea of an average SWE team experience.

Team Objectives:

Delivering an efficient and high-functioning software product under a certain deadline.

Applying SWE principles like unit testing, CI/CD Pipelines, documentation, etc.

Building a safe and respectful environment for collaboration.

Supporting each other in their learning and SWE goals.

2. Values, Mission, and Common Goals

Core Values:

Respect: We treat all members with dignity and try to take into consideration their personal situations.

Support: We provide aid to each other in order to encourage growth and help with difficult situations.

Kindness / Psychological Safety: Making mistakes is okay. Ensuring that no one is targeted or singled out during their mistakes is crucial, avoiding blame and focusing on the issue at hand is a core aspect.

Teamwork / Cooperation: We want to solve problems collectively, while providing each other support and effective collaboration skills.

Common Goals:

Learning how to efficient collaborate and work in a team-based SWE environment.

Delivering a working, and efficient product within an academic deadline.

Practicing clear and frequent communication between each other.

Reflecting regularly and adjusting our team dynamics as needed.

3. Strengths and Weaknesses

Team Strengths:

Having a variety of members originating from a diversity of backgrounds, providing their own hard and soft skills into the equation, increasing our problem-solving capabilities.

Varying levels of expertise can allow us to provide peer teaching and pairing development opportunities.

Team Weaknesses:

Scheduling conflicts due to having other academic/personal obligations outside of SWE.
Tight deadlines may limit the scope and quality of our final product.

Mitigation Strategies:

Using tools like When2Meet or LettuceMeet to schedule weekly meetings.
Defining and staying to general meeting hours where most team members are able to attend.
Setting clear weekly sprint goals and documenting for those unable to attend.
Setting regular retrospectives to evaluate progress or an obstacles in our project.

4. Roles, Responsibilities, and Accountability

General Accountability:

Everyone will take part in sprint planning and reviews.
Everyone will share the responsibility for contributing to documentation, commenting on PRs, and keeping the issue trackers up to date.

Specific Roles:

George (Secretary Lead): Pushing finalized meeting notes and information to the project repo. **Cody (Team Mediator):** Establishing resolutions of any interpersonal conflicts or disagreements. **Kevin and Ali (CI/CD Leads):** Setting up and maintaining CI/CD pipelines. Other are allowed to observe and contribute to the pipelines.

All Members: Acting as "secretaries" during meetings to ensure all details and notes have been described in our meeting notes before finalization.

Driver/Observer Pairing: Based on our team's approach and the current project requirements. We will be establishing a Pairing development, with each task or sub-task being assigned a *Driver* and an *Observer* for mentorship and efficiency.

Roles may rotate based on the availability, interest, and/or emergencies.

5. Needs and Expectations

Team Communication:

Platforms: Discord, Slack, In-person.

Weekly stand-up summaries: Each member describes their accomplishments during the week, which is followed by setting up our goals and tasks for the following sprint.

Collaboration Expectations:

Contacting or letting the team know if you are unable to meet a deadline due to personal conflict.

Submitting pull requests with detailed commit messages and request reviews.

Arriving on time and participating in the meetings.

Time Commitment Considerations:

Weekly meetings are necessary, if you are unable to attend for a reason, let the team know.

Conflict Resolution Process:

1. Address the issue, not the individual - Raise concerns without targeting a team member or putting the blame on someone.
2. Mediation - If a direct conversation is not enough, involve the team mediator.
3. Team Review - For recurring issues, the team should hold a retrospective discussion to come to a fair solution.

6. Working Methods, Rules, and Progress Tracking

Development Approach:

Using Github Projects/Issues to organize work.

Diving work into small, trackable tasks assigned to 2-3 members per issue.

Using agile-style sprints with weekly results.

Fair Contribution and Accountability:

Working is tracked with Github commits, issue completions, meeting participations, as well as documentation notes detailing the work done.

Having a rotating accountability check-in (like a peer review) can help verify equal and efficient contribution.

Measuring Success:

Sprint goals should be met on time.

Working demo milestones should be accomplished.

Code must meet quality standard and follow the guidelines.

If you have read the rules and agree to the following terms, sign below: Julia Wu