

Team Expectations Agreement¹

When working in a team, problems occasionally arise. One source of problems is the differing expectations of team members. For example, one person may think that an email response should come within 2 hours, while another may think that 2 days is acceptable.

This document will serve as a guideline while working on establishing team expectations. In the space below, write down the list of agreed upon guidelines that your team intends to follow. Include guidelines for the following:

- **Methods of communication** (email, phone, messenger, text, ...)
- **Communication response times** (email, phone, messenger, text, ...)
- **Meeting attendance** (when to meet, whether all meetings are mandatory, ...)
- **Running meetings** (when, where, face-to-face vs. online, who takes minutes, ...)
- **Meeting preparation** (whether preparation is needed, what to prepare, ...)
- **Version control** (what to/not to commit, content of log messages, ...)
- **Division of work** (how to divide work, who will decide who does what, ...)
- **Submitting assignments** (when to submit, who will submit, who will review the submission, ...)
- **Contingency planning** (what if a team member drops out, what if a team member consistently misses meetings, what if a team member is academically dishonest, ...) We suggest that in these cases, a team promptly seeks help from the instructor. It is important not to let such situations escalate.

The list above is just meant to get you started. If you had any team problems in the past, think about what went wrong and how expectations can be set to prevent those types of problems.

(Continued on the other side.)

¹Based on *Turning Groups into Effective Teams*, Barbara Oakley et al., 2004.

Team Guidelines (continued)

We accept these guidelines (written in the page 3 of this document) and intend to fulfill them (sign below):

Name (print)	Signature	Date

At the time when you will submit the sprint 0 materials, please submit this document as well. In the event of team disagreements, you may be asked to show this form to your instructor.

Team Guidelines (continued)

Please write down your guidelines in this page.

Methods of Communication

Primary: Discord (team server)
Secondary: Email (for formal/urgent matters)
Emergency: Instagram

Communication Response Times

Discord messages: Within 12 hours on weekdays, 24 hours on weekends
Urgent issues: Within 4 hours
Email: Within 24 hours

Meeting Attendance

Frequency: Twice per week (e.g., Monday & Thursday evenings)
Mandatory: Sprint planning and sprint review meetings
Optional: Three standups a week (async on Discord)
Notice: 12-24 hours advance notice if unable to attend

Running Meetings

Format: Online via Discord voice channel
Duration: 30-60 minutes max
Minutes: Rotate note-taker each meeting, post notes in Discord
Agenda: Posted 24 hours before meeting

Meeting Preparation

Review assigned tasks before meeting
Update Trello board with current progress
Prepare questions or blockers to discuss

Version Control

Commit: Working code only (must compile/run)
Never commit: node_modules/, target/, .env, *.db files
Commit messages: Clear and descriptive (e.g., "Add user login API endpoint")
Branches: Feature branches, merge via pull requests
Main branch: Always deployable

Division of Work

Tasks assigned during sprint planning meetings
Workload distributed equally based on skill level
Volunteer for tasks first, then discuss assignment if needed
Reassign tasks if someone is blocked or overwhelmed

Submitting Assignments

Deadline: 24 hours before official due date (internal buffer)
Review: At least one teammate reviews before submission
Final check: All team members approve via Discord before push

Contingency Planning

Unresponsive member: Contact via all channels, escalate to instructor after 48 hours
Missed meetings (2+): Team discussion, then involve instructor
Drops course: Immediately notify instructor, redistribute work
Academic dishonesty: Report to instructor immediately
Skill gaps: Pair programming, AI tools assistance, no blame culture

Code Quality Standards

Use AI tools (Claude, Copilot) but understand the generated code
Comment complex logic
Follow Java/React best practices