

# TeamWork as Software Engineers

## Season 2025-II

### TeamWork Guidelines

Eng. Carlos Andrés Sierra, M.Sc.

Lecturer

Computer Engineering Program

School of Engineering

Universidad Nacional de Colombia

The following guidelines are designed to help you manage your team project efficiently and professionally. They incorporate industry best practices and aim to enhance your technical skills while fostering effective teamwork.

#### Recommendations

##### 1. Goal Setting and Progress Tracking:

- Define clear, measurable goals for each sprint and review progress at the end.
- Use metrics (e.g., velocity, burndown charts) to track team performance and adapt plans.

##### 2. Repository Setup:

- Create a dedicated `git repository` (or *organization*) for your project.
- On GitHub, include a **README.md** with a project summary, a proper **.gitignore** (tailored to the programming language used), and a suitable open-source **LICENSE** (e.g., GNU/GPL).

##### 3. Project Structure:

- Organize your project with a dedicated **src** folder for **code** and a **docs** folder for **documentation**.
- Ensure that your **README** provides a clear overview of your **business model**, **user stories**, and **repository structure**.

---

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Lecturer at Universidad Nacional de Colombia.

Any comment or concern regarding this workshop can be sent to Carlos A. Sierra at: *casier-rav@unal.edu.co*.

#### 4. Commit Practices:

- Make **atomic commits** with *clear* and *descriptive messages*.
- **Commit** only a few files per *commit* to simplify *reviews* and *potential reversals*.

#### 5. Sprint and Task Management:

- Plan your work in **two-week sprints** with *clearly defined, bite-sized tasks*.
- Use a **task board** (e.g., Github Projects or Trello) to **track progress** and **adjust tasks** as needed.
- Conduct regular **stand-up meetings** (daily or bi-weekly) to discuss *progress*, *blockers*, and *next steps*.
- Hold **sprint retrospectives** to reflect on what went well and what can be improved.

#### 6. Branching Strategy:

- Create a separate **branch** for each team member; for example, using the notation `developer/your_name`.
- *At the end* of each **sprint**, **merge** your work into the **main branch** via **Pull Requests**, using peer reviews to ensure quality.

#### 7. Agile and Documentation Best Practices:

- Even if you are new to **agile methodologies**, strive to implement them by defining **weekly tasks** and adhering to **sprint plans**.
- Ensure that your **code** includes *detailed documentation* (**methods documentation**, **inline comments**, among others) for *clarity* and *maintainability*.

#### 8. Team Dynamics:

- Maintain *professional* and *respectful communication* at all times.
- Embrace **feedback** as an opportunity to improve —both individually and as a team— and always focus on achieving the **final goal**.
- Use *clear*, *concise*, and *respectful* language in all **communications**.
- **Document** decisions and discussions in *shared channels* (e.g., Slack, project board).
- Schedule **regular check-ins** to ensure alignment and resolve conflicts early.

#### 9. Feedback and Continuous Improvement:

- Encourage **open, constructive feedback** within the team.
- Regularly *review processes* and outcomes to identify improvement opportunities.
- Celebrate successes and *learn from failures together*.

**10. Roles and Responsibilities:**

- Clearly define each team member's **role** (e.g., Scrum Master, Product Owner, Developer, Tester).
- *Rotate roles* periodically to build versatility and understanding.
- Ensure accountability by assigning *ownership* for specific deliverables.

**11. Deliverables and Demos:**

- Ensure **all deliverables** are *well-documented, tested*, and meet the agreed *requirements* before submission.
- Prepare *short demos* or presentations at the **end of each sprint** to showcase progress. It is a preparation for the final delivery.

If you have any questions regarding these recommendations, feel free to ask during class, at the designated student support times, or via the course's Slack channel.