

Sections of a Working Agreement

A working agreement is a document, or a set of documents that describe how we work together as a team and what our expectations and principles are.

The working agreement created by the team at the beginning of the project, and is stored in the repository so that it is readily available for everyone working on the project.

The following are examples of sections and points that can be part of a working agreement but each team should compose their own, and adjust times, communication channels, branch naming policies etc. to fit their team needs.

General

- We work as one team towards a common goal and clear scope
- We make sure everyone's voice is heard, listened to
- We show all team members equal respect
- We work as a team to have common expectations for technical delivery that are documented in a [Team Manifesto](#).
- We make sure to spread our expertise and skills in the team, so no single person is relied on for one skill
- All times below are listed in CET

Communication

- We communicate all information relevant to the team through the Project Teams channel

- We add all [technical spikes](#), [trade studies](#), and other technical documentation to the project repository through [async design reviews](#) in [PRs](#)

Work-life Balance

- Our office hours, when we can expect to collaborate via Microsoft Teams, phone or face-to-face are Monday to Friday 10AM - 5PM
- We are not expected to answer emails past 6PM, on weekends or when we are on holidays or vacation.
- We work in different time zones and respect this, especially when setting up recurring meetings.
- We record meetings when possible, so that team members who could not attend live can [listen](#) later.

Quality and not Quantity

- We agree on a [Definition of Done](#) for our user story's and sprints and live by it.
- We follow engineering best practices like the [Engineering Fundamentals Engineering Playbook](#)

Scrum Rhythm

Activity	When	Duration	Who	Accountable	Goal
Project Standup	Tue-Fri 9AM	15 min	Everyone	Process Lead	What has been accomplished, next steps, blockers
Sprint Demo	Monday 9AM	1 hour	Everyone	Dev Lead	Present work done and sign off on user story completion

Activity	When	Duration	Who	Accountable	Goal
Sprint Retro	Monday 10AM	1 hour	Everyone	Process Lead	Dev Teams shares learnings and what can be improved
Sprint Planning	Monday 11AM	1 hour	Everyone	PO	Size and plan user stories for the sprint
Task Creation	After Sprint Planning	-	Dev Team	Dev Lead	Create tasks to clarify and determine velocity
Backlog refinement	Wednesday 2PM	1 hour	Dev Lead, PO	PO	Prepare for next sprint and ensure that stories are ready for next sprint.

Process Lead

The Process Lead is responsible for leading any scrum or agile practices to enable the project to move forward.

- Facilitate standup meetings and hold team accountable for attendance and participation.
- Keep the meeting moving as described in the [Project Standup](#) page.
- Make sure all action items are documented and ensure each has an owner and a due date and tracks the open issues.
- Notes as needed after planning / stand-ups.
- Make sure that items are moved to the parking lot and ensure follow-up afterwards.
- Maintain a location showing team's work and status and removing impediments that are blocking the team.
- Hold the team accountable for results in a supportive fashion.

- Make sure that project and program documentation are up-to-date.
- Guarantee the tracking/following up on action items from retrospectives (iteration and release planning) and from daily standup meetings.
- Facilitate the sprint retrospective.
- Coach Product Owner and the team in the process, as needed.

Backlog Management

- We work together on a [Definition of Ready](#) and all user stories assigned to a sprint need to follow this
- We communicate what we are working on through the board
- We assign ourselves a task when we are ready to work on it (not before) and move it to active
- We capture any work we do related to the project in a user story/task
- We close our tasks/user stories only when they are done (as described in the [Definition of Done](#))
- We work with the PM if we want to add a new user story to the sprint
- If we add new tasks to the board, we make sure it matches the acceptance criteria of the user story (to avoid scope creep). If it doesn't match the acceptance criteria we should discuss with the PM to see if we need a new user story for the task or if we should adjust the acceptance criteria.

Code Management

- We follow the git flow branch naming convention for branches and identify the task number e.g. `feature/123-add-working-agreement`
- We merge all code into main branches through PRs

- All PRs are reviewed by one person from and one from Microsoft (for knowledge transfer and to ensure code and security standards are met)
 - We always review existing PRs before starting work on a new task
 - We look through open PRs at the end of stand-up to make sure all PRs have reviewers.
 - We treat documentation as code and apply the same [standards to Markdown](#) as code
-

Last update: August 26, 2024