**Working Agreement**

**Goal**

This living document represents the principles and expected behavior of everyone involved in the project. It is not meant to be exhaustive nor complete. The team should be accountable to these standards and revisit, review, and revise as needed. The agreement is signed off by everyone.

**Code of Conduct**

We pledge to follow the [Microsoft Open Source Code of Conduct: Commercial Software Engineering](file:///C:\app\code_of_conduct.md) [TechnicalEngagements](file:///C:\app\code_of_conduct.md) which strives to ensure that every member of our team and the world-class organizations we engage with feels welcome and safe. In that spirit, we ask you to agree to and uphold our conduct standards as part of this technical engagement and to hold us to these same standards.

**Communication**

* We communicate all information relevant to the team through the project communicationTeams channel.
* We add all research results, design documents and other technical documentation to the project repository through 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 - 3PM (PST) when possible.
* Sometimes emails or IM messages happen when thoughts come to mind and may come outside of working hours. We are not expected to answer emails or IMs past 5PM, 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 stories and sprints and live by it.
* We follow engineering best practices like the [Code](https://github.com/microsoft/code-with-engineering-playbook) [With EngineeringPlaybook](https://github.com/microsoft/code-with-engineering-playbook)
* We try to do demos every sprint when appropriate, we don't force a demo to fit just because it is the end of the sprint.
* Appropriate time to prepare demos is given and the expectation of a demo on a moments notice will be avoided.

**Scrum rhythm (This will be completed after ADS)**

We will follow guidance from the Code With Engineering Playbook on [Agile coreexpectations](https://github.com/microsoft/code-with-engineering-playbook/blob/main/docs/agile-development/core-expectations/README.md), which provides guidance on the purpose and structure of the ceremonies noted below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **When** | **Duration** | **Who** | **Accountable** | **Goal** |
| Project Stand-up &  Parking Lot | Mon-Fri 10:05  Pacific | 40 min | Everyone | Process Lead | Stand-up: What has been accomplished, next steps, blockers (15 minutes)  Parking Lot: Optional extended discussion (25 minutes) |
| Sprint Review | Tuesdays  9:30AM | 120  minutes | Everyone | Tech Lead | Present work done and sign off on user story completion. |
| Sprint Retrospective | Tuesdays  10:15AM | 90  minutes | Everyone | Process Lead | DevTeams shares learnings and what can be improved. |
| Sprint Planning | Wednesdays  9:30AM | 90  minutes | Everyone | Product  Owner | Size and plan user stories for the sprint. |
| Backlog refinement | Mondays | 90 | Dev Lead, | Product | Prepare for next sprint and ensure that |

**Ceremony Lead**

The process of leading the daily stand-up will rotate around the team on a sprint-by-sprint basis, with the crew taking turns to lead the ceremony in alphabetical order. The rotation includes both the Microsoft team and partner. This helps to keep everyone engaged in what is happening in the entire project and reinforces the skill of managing the board and making sure tasks are moving at an appropriate speed (velocity).

* Facilitate stand-up meetings and hold team accountable for attendance and participation.
* Keep the meeting moving as described in the Stand-up section of the [Agile core expectations](https://github.com/microsoft/code-with-engineering-playbook/blob/main/docs/agile-development/core-expectations/README.md) page.

**Process Lead**

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

* 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 stand-up meetings.
* Facilitate the sprint retrospective.
* Coach Product Owner and the team in the process, as needed.

**Dev Lead**

* Serves as team lead of the development team.
* Works with TPM on backlog maintenance.
* In charge of upskilling and tasking and time estimating.
* Responsible for coaching team and engineering quality.
* Focuses on specific technical challenges needed in project, including deeper technical explorations.
* Responsible for architecture and leading technical discussions.
* Works with TPM on feedback, technical gaps, and sharing.
* Works with TPM to bring in Subject Matter Experts (SMEs) when appropriate.

**Product Owner / Technical Program Manager**

* Responsible for turning customer and partner requirements into detailed stories with definitions of done.
* Responsible for documenting risks appropriately, surfacing risks to the right stakeholders at the right time, and managing mitigations.
* Responsible for scoping and defining projects as mapped to customer/partner business goals.
* Actively manages the customer Product Owner and executive/business sponsor, and technical stakeholders.
* Jointly responsible with Tech Lead for Game Plan Documentation and Review.
* Responsible for backlog organization, grooming, and prioritization with customer/partner.
* Facilitates customer/partner workshops, leads discussions about problems and business goals.

**Backlog management**

* 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 the deliverable has been merged to the code repository
* We work with the TPM if we want to add a new user story to the sprint

**Estimation**

* We will estimate Stories with [Fibonacci Sequence](https://medium.com/ducalis-blog/story-points-how-fibonacci-sequence-works-for-agile-estimation-4c7e0081f4d8) story points to estimate our work, and will make our estimates as a team during Sprint Planning. This will help us gauge how much work we commit to within a sprint.

|  |  |
| --- | --- |
| **Story Points** | **Estimated Effort** |
| 1pts | half a day of effort or less |
| 2pts | full day of effort |
| 3pts | day and a half of effort |
| 5pts | 2 - 3 Days of effort |
| 8pts | 4 Days of Effort (~1 week of a sprint) |
| 13pts | 6 Days of effort (~1.5 weeks of a sprint) |
| 21pts | 10+ Days of effort (a Full end to end Sprint, or more) |

* Tasks and Bugs require hour estimates, as this will track our velocity. Task and bug owners are responsible for maintaining the Original Estimate, Remaining, and Completed fields in their Task record.
* It is everyone's responsibility to update their Tasks prior to the stand-up meeting

**Code management (Will update after ADS - Partner dependent)**

* We prefer a PR per Task, rather than per User Story. Small PRs are preferred so the team can review them quickly and provide feedback early on in design, encouraging iterative development.
* Marking up your PR with informative comments is recommended to provide reviewers an understanding of your code & thought process as well as to direct them to the critical areas in a PR. you should consider code comments that live on in the codebase, vs ephemeral notes that help the reviewer understand what your check-in is doing. These comments are encouraged, not required.
* Consider adding images of UI in the comments when appropriate.
* We address and resolve all PR comments. Comments suggesting improvements or refactoring may be considered out of scope by the PR creator, and are not expected to be implemented in the active PR. Instead, a new story should be created from the comments and pulled into a sprint accordingly. This decision rests with the PR author.
* Avoid extended conversations or debate in comments - if you need to discuss in more detail, talk.
* We follow the git flow branch naming convention for branches and identify the task number, e.g., alias/123- add working-agreement.
* We merge all code into main branches through PRs.
* All PRs are reviewed by a minimum of two people.
  + The most recent pusher is prohibited from approving their own changes.
* All PRs will use Squash merge.
* We link our Pull Request to the work item that the request is addressing.
* We always review existing PRs before starting work on a new task.
* For PR comments other than required fixes, reviewers will add a tag to the beginning of the comment:
  + **[typo]** For spelling and grammar errors.
  + **[nit]** For small fixes that are not necessary.
  + **[fmok]** "For my own knowledge." For the reviewer's own learning rather than feedback for the PR.
* We look through open PRs at the end of stand-up to make sure all PRs have reviewers and are not blocked.
* We treat documentation as code and apply the same [standards to Markdown](https://github.com/microsoft/code-with-engineering-playbook/blob/main/docs/code-reviews/recipes/markdown.md) as code.
* While Pull Requests will be required as an official form of review for any work done, informal ad-hoc code reviews or design reviews are encouraged.
* When we need to, we use the **Mark as draft** option to set the PR to draft status to indicate it's not ready for review. You can select **Publish** to remove draft status and mark the PR as ready for review.