TESLATECH TECHNICAL TEAM WORKFLOW GUIDELINES

This documentation outlines the workflow, guidelines, and best practices for a technical team to function efficiently and collaboratively while maintaining discipline and ensuring high-quality output. Each point outlined herein is to be adhered to with professionalism and discipline by all members of the technical team. For clarity and ease of implementation, the points are categorized into distinct sections.

1. ClickUp

a. Prioritization:

i. Product Stability is always High Priority. If needed, should work on off-days as well.

ii. Priority of Tasks from CEO/Management should be propagated through Project Manager.

iii. Priorities of tasks related to upcoming releases to be communicated and discussed with Project Manager along with related departmental dependencies.

iv. Team leads determine task priorities after discussions using ClickUp's priority tags to ensure critical tasks are addressed first. Regularly review priorities to align them with project goals.

v. Minimize mid-task interruptions unless the new task is critical. Leads evaluate urgency and potential impact before adjusting priorities, ensuring smooth task execution.

b. Task Deadlines:

i. Specify clear deadlines for each task when it is created or during breakdown. Deadlines should be realistic and account for task priority, complexity, dependencies, and resource availability to avoid unnecessary delays.
NOTE: Work hours for leads should be capped at 80% of total daily work hours.

ii. Add max 1 working day for buffer, as necessary.

iii. Take consideration for pre-approved leaves

iv. Discuss and clarify Inter-team and department dependencies.

c. Feature Ownership and assigning of the task:

i. Assign one member as the Feature Owner to oversee the task. The Feature Owner is responsible for coordinating with other team members, tracking progress, and ensuring timely completion. This ensures accountability and clarity in task execution.

ii. In case of absence of the feature owner:
   1. Next available team member to be Person of Contact
   2. Else, Lead to be Person of Contact

iii. In case of deadline miss, feature owner to report respective leads/overseer on the reasoning for it and document it in the Task itself.

iv. Assign a Watcher to tasks requiring dependency oversight without direct contribution. This prevents unnecessary clutter in Kanban columns while keeping relevant stakeholders informed.

v. Mention impact analysis based on the documentation and work done.

d. Time Tracking (Mandatory):

i. Use ClickUp's time tracking features to monitor individual's time spent on tasks. This data helps identify inefficiencies and provides insights for future estimations and planning along with the details to complete the current pipelines.

ii. Track Time for Peer/Final Review and Breakdowns in the actual Task itself (main task, if available) with relevant comments.

e. Backlog Review and Retrospectives:

i. Revisit Backlog (Tech Kanban) monthly.

ii. Document Retrospective in respective ClickUp tasks. Retro to be done by Feature Owner after 2 weeks of release date and add comments in the respective task(s).
   1. If no problem, go to Complete
   2. If there are any changes to be made, adding a subtask or creating a main task linking to the task with the issue (dependent on the task).

f. Planning Release and Development Cycles:

i. Define start and end dates for each cycle to ensure clarity. Calendar generated.

ii. Release cycle to be strictly followed, 3 weeks cycle.

iii. Planning to be done for at least 2 future versions. Later on to be increased as per the stability of the workflow and tasks.

iv. Feature Freeze will be done 4 days prior to Closed Beta date. Closed Beta to be considered as Demo for all Teslatech members.

v. Only the tasks completed prior to code-freeze-day will be included in the upcoming release, else will be shifted to the next release.
NOTE: Leads must take responsibilities in order to provide clarification on why the task was not completed as per the plan.

2. Communication

a. 4 Pillars Communication:

i. Discuss and clarify the objective to all the pillars during the meeting held.

ii. Make sure that all the pillars are on the same page and need to address the upcoming release bottlenecks and issues, if any during the development phase.

b. Interdepartmental Communication:

i. Encourage regular updates and check-ins with other departments on shared tasks. Scheduled sync-ups ensure alignment and address interdependencies promptly (every Friday, All Leads Tech-Sync), minimizing delays. Department leads to prioritizing the dependencies so that the other department's tasks are not put on hold.

ii. Discuss with all departments before task assessment.

iii. Daily Standup to be held for all departments. (9:30 AM)

  1. DSM missing members should be provided in written form on the Discord channel.

iv. While transferring tasks to other departments,

  1. Add a relevant comment or generate tasks as well as communicate through Discord.
  2. Remind regarding the timeline for the completion agreed on the task assessment.
  3. Clarify the delay of dependencies

v. Pre-approved leaves to be communicated on the approved day and discuss the impact along with providing the person of contact in their absence.

vi. At least one member of each department should be present on the working day.

c. Protocol:

i. Message or Task Acknowledgment:

  1. Respond to messages promptly to confirm receipt and understanding.
  2. If no response is received within a reasonable time (e.g., 30 minutes during working hours in case of urgency), escalate by calling (phone) the concerned person, if necessary.

ii. Communicate after the completion of the task for the peer review and final review even after assigning the comment in the ClickUp so that the tasks propagate in a timely and orderly fashion.

d. Meetings:

i. Prepare agendas beforehand to keep discussions focused and efficient.

ii. Decide on a Moderator and Minute keeper before the start of the meeting.
iii. The meeting minutes should be brief and concise.
iv. Cap meetings at 30-45 minutes (max) to ensure productivity. In case of extension, 5 mins breaks are encouraged to avoid zone-outs.
v. Use designated meeting rooms or virtual platforms for discussions to avoid disturbing other team members.
vi. Summarize the conclusion of the meeting, before ending. When a conclusion is needed for the meeting agenda, the conclusion should also be clearly communicated. Also, set an agenda for the next meeting, if needed.
vii. Track time for each meet.

3. Team Discipline

a. Mini Breaks Policy:

i. Standard breaks are 5-10 minutes.
ii. Extended breaks require prior approval and an explanation to the lead to ensure accountability.
iii. If the break is extended more than the mentioned time frame, the person should explain the reasoning.

b. Professional Conduct

i. Foster a focused environment by avoiding unnecessary chatter and distractions. Respect teammates' focus time and schedule discussions for collaboration.
ii. Avoid loud discussions in shared spaces to preserve a productive work environment.
iii. Handle disagreements constructively by listening to all perspectives and working toward a consensus.
iv. Maintain a respectful tone during all communications.
v. Lengthy discussions should be conducted in designated meeting areas.

4. Leadership and Reporting

a. Quarterly Reports

i. Tech leads prepare detailed quarterly reports summarizing progress, estimations, and team goals achieved.
ii. Reports to highlight achievements, challenges, and action plans for the upcoming quarter as well.
iii. Feature owners to write reports after each feature completion.
    1. Report should include collaborative efforts of team members.

b. Leadership Development

- i. Provide mentorship opportunities for team members to develop leadership skills.
- ii. Encourage them to take ownership of tasks and participate in decision-making processes.

5. Planning and Documentation

a. Effective Planning

- i. Pre-schedule feature and bug fix phases to create a balanced development cycle.
- ii. Allocate buffer time for unforeseen issues or delays, ensuring project timelines are met.
- iii. Include time for Code Reviews and Bugfix in Planning
- iv. Extended estimated time to be included only in Planning, not in the actual task itself so that members are encouraged to complete the task in the mentioned time frame.

b. Comprehensive Documentation

- i. Flow chart and tech related documentation
- ii. Maintain detailed and updated task and feature documentation.
- iii. Use clear descriptions, goals, and acceptance criteria to minimize ambiguity and facilitate efficient execution.
- iv. In the case of changes in User Story/ Description when a task has started, the updated description is to be added on the description on the last line, without deleting or changing content of the original Description. If any requirements from the original description are not needed anymore, Strikethrough to be done.
- v. Documentation of Feature Code Flow having at least a Flowchart after feature completion.
  1. Workflow
  2. What it does
  3. Challenges Faced

c. Standardize Git Commit Messages

- i. Study this later: https://www.conventionalcommits.org/en/v1.0.0/
- ii. Brief session to be held on this

6. Quality Assurance

a. Task Coordination

- i. QA communicates with feature owners (if unknown, Leads) before creating tasks to prevent duplication/verification of the bug. This streamlines testing efforts and avoids redundancy.

ii. Logs for Bugs and Steps to Reproduce are mandatory where applicable.

iii. Environment (Branch/Production) and Version of where the bug encountered is mandatory.

iv. If a bug is encountered, do not move ahead and start fresh (For QA) to avoid chain-reaction.

b. Integration Testing

i. Conduct thorough integration testing to verify compatibility and functionality across different modules and systems. This ensures smooth operation and reduces post-release issues.

ii. Unit Test Screenshots to be included before sending for Review by developers.

iii. Integration Testing to be done before being sent for Review by developers.

iv. Regression testing to be done strictly for core gameplay, must not be skipped before release by the QA team.

v. Server Side testing and platform specific testing to be done by QA testers using tools, when needed re-run the unit test done by the developers with tweaked input/output.

vi. Local server should be used for the testing, then only to be merged into the staging server for testing.
TBD: Repo access to be granted to interdepartmental or private to only department. Should only be able to access and not manipulate the repo.

7. Evaluation and Benchmarking

a. Performance Evaluation

i. Regularly evaluate task performance by tracking metrics such as completion time, code quality, and adherence to deadlines.

ii. Use this data to identify areas for improvement and implement actionable solutions.

    1. Consider Reports from employees after each feature completion

b. Benchmarking

i. Set clear performance benchmarks for productivity and task quality.

ii. Periodically compare results to identify gaps and implement process improvements.

iii. Measures/Properties for Benchmarking?

    1. TBD: Needs R&D, measures should be ready within Q3
    2. Monthly 1-on-1.

8. Problem Resolution

a. Priority Visualization

i. Use ClickUp’s dashboard, custom views, and priority tags to enhance priority visualization and task tracking, ensuring the team stays focused on the most critical tasks.

1. Board/List View

9. Summary of Expectations

a. Maintain a structured workflow and prioritize effective communication and time management.

b. Encourage discipline, leadership, and focus within the team.

c. Strive for high-quality deliverables through thorough planning, testing, and collaborative efforts.

d. Find out measures to make the environment more fun to work on. Attending seminars, coaching lectures, team building activities, etc.

Tech System Workflow Visualization : link here