Development Plan
Software Engineering

Team #23, Project Proxi
Savinay Chhabra
Amanbeer Singh Minhas
Gourob Podder
Ajay Singh Grewal

Table 1: Revision History

Date	Developer(s)	Change
September 21, 2025	Savinay Chhabra	Created Draft Development Plan; Sections 1, 2, 3, 4, 5, 7
September 22, 2025	Savinay Chhabra	Sections 6, 8
September 22, 2025	Ajay Singh Grewal	Section 9, 10, 11
September 22, 2025	Amanbeer Singh Minhas	Team Charter and Reflection

This report outlines the development plan for the Proxi AI Assistant. The document details the following sections of the Development Plan:

- Confidentiality Information
- IP Protection
- Copyright License
- Team Meeting Plans
- Team Communication Plans
- Team Member Roles
- Workflow Plan
- Project Decomposition and Scheduling
- Proof of Concept Demo Plan
- Expected Technology
- Coding Standard

1   Confidential Information?

There is no confidential information to protect.

2   IP to Protect

There is no IP to protect.

3   Copyright License

This project adopts the MIT License. LICENSE file can be found here.

4   Team Meeting Plan

The Team will meet every Wednesday from 4:00PM - 4:45PM in person. Campus library meeting rooms will be booked weekly, depending on availability.

- A rotating chair will lead each meeting and prepare the meetings agenda in advance.
- A rotating notetaker will record meeting minutes, action items and decisions made in the meeting.

- Feature requests and Decisions will be documented using Github Issues

- Meetings with graduate students speacializing in Human-Computer Interfaces will occur biweekly. These will either be virtual or in-person depending on availability.

5 Team Communication Plan

Urgent issues or clarifications will be communicated through Discord. Discord will also be used for any virtual meetings between the team. Meetings with industry advisors and graduate students will take place over MS Teams if held virtually. Github issues will the the source of truth for techincal information and progress tracking.

6 Team Member Roles

Table 2: Team Member Roles

Role	Responsibilities	Member(s
Meeting Chair	Sets meeting agenda and leads weekly meetings	Rotates weekly
Note Taker	Records meeting minutes and decisions	Rotates weekly
Team Liaison	Primary point of contact for project supervision and industry advisors	Savinay Chhabra
Subject Matter Expert	Develops deep knowledge of the projects domain	Gourob Podder
Test Lead	Responsible for planning, organizing and overseeing testing process.	Amanbeer Singh Minhas
Reviewer	Responsible for evaluating Pull Requests to ensure that they adher to team standards.	Ajay Singh Grewal

7 Workflow Plan

- Each Feature/Bug/Defect will have a Github Issue created for it. This issue will included detailed information along with current progress, assignee and appropriate labels. Github Projects will be used to track all incoming and in-progress issues.

- Each Issue will a branch for any associated development work to be done.

- Once the author is satisfied with their solution, they can put up a Pull Request to merge their changes into the main branch. Each Pull request must link to one or more issues and have a detailed description of the changes made and the user impact of those changes.

- After the Pull Request has been raised and all required pre-submit checks have succeded, the author can add the "Review Needed" label to their PR.

- Each PR must have at least 2 reviewers. After the reviewers have approved, the PR can be merged and the associated issues can be closed.

8 Project Decomposition and Scheduling

The team will use the Kanban Board on Github Projects to track incoming and in-progress issues. The Board for the project can be found here.

Table 3: Project Scheduling

Deliverable	Deadline
Problem Statement, POC Plan, Development Plan	September 22, 2025
Req. Doc. and Hazard Analysis Revision 0	October 6, 2025
V&V Plan Revision 0	October 27, 2025
Design Document Revision -1	November 10, 2025
Design Document Revision 0	January 19, 2026
Revision 0 Demonstration	February 13, 2026
V&V Report and Extras Revision 0	March 09, 2026
Final Demonstration (Revision 1)	March 29, 2026
EXPO Demonstration	TBD
Final Documentation (Revision 1)	April 06, 2026

9 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

The main risks revolve around security, adaptability, and user trust, since the AI will have access and control to one's computer through model context protocols, functioning as the central master control program.

Granting the AI full control of the computer introduces major security risks and vulnerabilities. Since the MCP will have deep access to a user's computer, including files, settings, and applications, any vulnerabilities could lead to catastrophic consequences. Malicious actors could exploit the system to steal sensitive data, install harmful software, or commit unintended actions which could become destructive. This risk must be reduced by having safeguards the AI must adhere to, a thoroughly tested environment, user permission before completing risky actions, and the ability to rollback changes. Hence, a robust error handling and recovery system is needed so incorrect commands do not disrupt any workflows or overall user experience.

Another big risk is the AI being capable enough to adapt to diverse range of computer environments. Every computer is different, with certain operating systems, setups, workflows, and personal preferences. If the AI cannot accurately understand its environment, it can lead to incorrect commands, actions, and overall poor experience. The AI must be capable of learning its environment in a dynamic manner and customize its actions accordingly.

Furthermore, since the AI has full control of one's computer, transparency is essential. If users feel uncertain of how the AI is doing things on their computer, it can risk user trust, leading to hesitance in using the tool. Creating a clear and intuitive user interface, where the technology is properly explained, will help mitigate this risk.

To address these risks, the PoC will focus on showing that the MCP and AI can safely and effectively interact with a computer, learn its environment, and execute meaningful tasks while keeping user trust.

10   Expected Technology

- Programming Language: Python for the AI agent, tool plugins, and overall backend structure. TypeScript and React for desktop UI

- Agent Core Libraries: pydantic, asyncio, httpx

- System Automation: For Windows we will use pywin32, UIAutomation-Core; for MacOS we will use pyobjc, some AppleScript/JXA bridge; For Browser we will use Playwright

- Storage/Database: SQL

- Pre-trained Models: OpenAI GPT-4o-mini API for natural language understanding and generation

- Local LLM For Future (if needed)

- Linter Tools: pylint, Prettier, ESLint

- CI/CD: Git, Github

11   Coding Standard

The team will strictly adhere to and follow the following coding standards to ensure clarity, maintainablity and readability:

- Python code will follow PEP 8 guidelines, using pylint for linting.

- TypeScript code will follow the Microsoft Coding guideline, using Prettier for formatting and ESLint for linting.

- Git and GitHub will be used for version control, with feature branching and pull requests for code reviews.

- Unit tests, integration tests, and exploratory tests will be performed to ensure code works before merging.

Appendix — Reflection

1. **Why is it important to create a development plan prior to starting the project?**
Having a development plan was crucial for our team because it gave us a shared understanding of our goals and responsibilities. I (Aman) realized early on that without a plan, we could easily duplicate work or miss deadlines, like in a previous project where we jumped straight into coding and ended up wasting time fixing miscommunication. Ajay mentioned that in his co-op, projects without clear roles caused constant confusion, so he emphasized setting up structured responsibilities and rotating roles and meeting lead. Savinay shared that documenting meeting agendas and communication methods helped him stay accountable he had seen other projects waste time on unstructured meetings.Gourob, drawing from his experience in AI-focused work, pointed out that a well-documented technical workflow prevents misunderstandings about modules like speech recognition or NLP if we intend to use those to achieve our goals.Overall, the plan helped us coordinate efficiently, understand dependencies, and feel confident moving forward.

2. **In your opinion, what are the advantages and disadvantages of using CI/CD?**
Our team had a lot of discussion about CI/CD and its implications. Savinay thought the biggest advantage is catching integration issues early; he recalled a project where manual merges led to significant delays, which could have been avoided with automated pipelines. I (Aman) agreed, but also noted that over-reliance on automated tests can be frustrating when debugging, especially if tests fail unexpectedly as I experienced in a small personal project. Gourob, who has worked with AI pipelines before, emphasized that CI/CD ensures multiple interdependent modules, like speech recognition and text-to-speech, remain stable even as different developers make frequent changes. Ajay raised the practical concern that setting up CI/CD initially is time consuming and misconfigurations can be discouraging, but he concluded that the long-term benefits, such as reliability and reduced errors, outweigh these challenges. Collectively, we agreed that CI/CD will help us maintain a high-quality codebase, streamline collaboration, and catch potential problems early before they escalate.

3. **What disagreements did your group have in this deliverable, if any, and how did you resolve them?**
Our team had a few meaningful disagreements while preparing this deliverable, mainly around the level of detail for our workflow and how to structure GitHub issues. Gourob argued for very detailed issue tracking and branching rules because in his previous co-op, unclear workflows had caused confusion and delayed delivery. I (Aman) thought a simpler approach would be faster and less bureaucratic, based on a personal experience where over planning slowed down a small project unnecessarily. Ajay

was concerned that if we over complicated the workflow, weekly meetings would become too long and tedious. Savinay suggested a compromise: we would document the main workflow in the development plan but leave flexibility within GitHub issues for the finer technical details.The resolution involved more than just compromise it was about understanding each other's reasoning. We each shared past experiences, explained why we preferred our approach, and considered potential consequences of both over-planning and under planning. In the end, we created a workflow that balanced structure and flexibility. This disagreement actually strengthened our team dynamic because it forced us to communicate openly, respect different perspectives, and make decisions based on practical experience rather than assumptions.

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

The long term goal for our team is to eventually make Proxi as our software business. In the short term goal is to create a product that can be showcased at the Capstone EXPO that can demonstrate not only technical functionality but also creativity, innovation, and accessibility. This project can be highlighted in future interviews, showcasing our collaboration and problem-solving skills and hopefully land us jobs. Another short term goal is that we are striving to receive an A+ in the course by delivering a high-quality, well documented, and fully integrated project.This dual focus on excellence and potential entrepreneurship motivates the team to deliver the project at the highest standard.

Attendance

Expectations

Expectations are clearly communicated to all team members regarding attendance and participation in meetings. So all team members are on the same page, that we want at least 90 percent attendance rate for all scheduled meetings. Each team member is expected to be punctual and fully engaged during meetings that will run for 45 minutes each week as discussed. As a full time student we know it might not be possible to always meet in person, so virtual attendance is acceptable if a member cannot make it in person. We do agreed that in case we are meeting virtually, we will have our cameras on to ensure active participation. If a member cannot attend a meeting, they are expected to notify the team at least 24 hours in advance either through Discord or email. If a member misses more than one meeting without a valid excuse, the team will discuss the situation and may involve the TA if necessary.

Acceptable Excuse

- Illness or medical emergencies

- Family emergencies

- Unavoidable class/lab conflicts

- Scheduled exams

- Transportation issues (with advance notice)

- Technical problems (e.g., laptop malfunction)

Unacceptable Excuse

- Forgetting about meetings or deadlines

- Personal errands during scheduled team time

- Last-minute cancellations without valid reason

- Preventable conflicts that weren't communicated in advance

All absences require timely communication to the team. The team expects honesty and responsibility from all members. When valid excuses arise, the team will work together to redistribute tasks and maintain project momentum.

In Case of Emergency

In case of an emergency and a team member cannot attend a scheduled meeting or meet a deadline, they should notify the team as soon as possible. Ideally notice should be at least 6 to 8 hours before the meeting or deadline, explaining the situation and providing an estimated time to catch up. The team will temporarily adjust responsibilities to keep the project on track. A member may miss no more than one meeting or one major deadline due to an emergency without extra discussion; missing more than that requires a team plan to catch up and may involve adjusting responsibilities or notifying the TA. Once the emergency is resolved, the member must promptly catch up on all missed work, review meeting notes, and resume contributions to upcoming tasks and deadlines, ensuring the team maintains momentum and uses scheduled time effectively.

Accountability and Teamwork

Quality

It is expected that all members ome to meetings fully prepared may it be having reviewed their assigned tasks or notes, or any relevant materials ahead of time. Deliverables brought to the team whether code, documentation, test cases, or

research should meet a high standard of clarity, completeness, and correctness. Before any work is submitted or integrated into the project, it will be reviewed by at least one other team member to catch errors, improve clarity, and ensure consistency. Work should be well-organized and easy for other team members to understand. By maintaining high-quality deliverables, thorough preparation, and a peer review process, the team can avoid technical debt, ensure smooth collaboration, and create a professional product that not only meets academic goals like earning an A+ but also has the potential to be refined and sold as a market-ready solution in the future.

Attitude

Everyone from our team is expected to be respectful, cooperative, friendly. All the team members should be able to freely communicate their ideas. We want meetings to be friendly and positive and it is expected to actively participate in discussions, listen to different perspectives, and give helpful feedback when reviewing each other's work and avoid negative critical tone. Our team do believe that a positive attitude is important because it not only boosts individual performance but also helps the whole team collaborate more effectively and stay motivated. We know that in a team conflicts are bound to come up and if they do, we will first talk it through as a team. This make sures that everyone has a chance to explain their side and present proofs of why they think one solution is better than another if its a conflict based on different solutions proposed.

Stay on Track

To keep the team on track, we'll rely on GitHub projects and issues to organize tasks, track deadlines, and see everyone's progress in real time. We'll review completed work, upcoming deadlines, and any blockers at every weekly 45-minute meeting so nothing falls through the cracks. Everyone is expected to attend most meetings (90 percent benchmark), complete their assigned tasks on time,and have their work peer-reviewed before submission. Team members who consistently do a great job will be recognized during meetings, and small rewards like leading a discussion or picking a fun team activity can be given. If someone falls behind or misses targets, the team will first check in with them to see if they need support and may redistribute tasks if needed. Repeated issues might involve minor consequences, like bringing snacks or coffee to the next meeting for the highest contributor on the project, or in serious cases, asking the TA for guidance. Due to this clear tracking, regular check-ins during our meets, and open communication, we make sure everyone contributes fairly. It will also help the team to stay coordinated, and the project keeps moving forward smoothly toward both our A+ goal and the long-term vision of creating a market-ready product.

Team Building

We'll be working together for a full year, so our team wants to make every meeting and work session something we actually look forward to. At the start of each meeting, we might do a quick check in where everyone shares a fun story, a small win from the week, or even a random interesting fact just enough to get everyone laughing or thinking. When we hit milestones, example finishing a big deliverable or meeting a tight deadline, we will celebrate with team dinners or casual outings. This will help us to enjoy the moment and connect outside of coding. We also plan to have spontaneous brainstorming sessions, mini challenges, or even 10-minute games to keep our energy up and creativity flowing. Spending a whole year together gives us a rare chance to really learn each other's strengths, keep motivation high, and grow Proxi beyond a school project, this is something we hope could eventually turn into a real business. By keeping things lively, flexible, and supportive, we make sure working on this project is fun, productive, and memorable for the whole team.

Decision Making

The majority of the decisions made by our team have be and will always be discussed collectively so that everyone can contribute their thoughts. We quickly vote to determine the best course of action if we are unable to agree either through discord or by show of hands in person. When differences arise, we make sure that everyone listens to one another and shares their point of view. We seek the TA for assistance or take a brief break if we are still unable to make up our minds. Everyone will feel heard in this way, and we can continue to work efficiently and remain inspired on Proxi.