# Personalized Code Completion

Mateusz Wojciechowski

December 2024

# 1 Task 1

There are a couple of reasons why I am interested in becoming a Project Manager Intern. First, I have a deep passion for Machine Learning, AI and software development, as shown by my extensive project portfolio. I consider Federated Learning to be one of the most promising fields in Artificial Intelligence because it supports scalability, real-time updates and more reliable models. It is also energy-efficient, reducing costs and helping to protect the environment.

During my academic journey and personal research, I studied software design, primarily using UML and explored various software development approaches such as Agile and its variants. I found these areas of computer science highly engaging and conversations with professionals in roles like Scrum Master further convinced me that a position at the intersection of technology and leadership could be an excellent match for my mixture of technical expertise, strong interpersonal skills and creativity.

I believe that my background and personal attributes make me particularly suited to work **at the intersection of the sciences and the humanities**, where I can contribute effectively by working on both the technical and human-centered aspects of software projects. Over time, I would be keen to advance into a leadership role on the Federated Compute team, leveraging my capabilities and passion to shape innovative projects that harness the power of AI.

# 2 Task 2

## 2.1 Justifying my approach

I adopted a **hybrid Gantt** approach to project planning, meaning I retain a classical Gantt chart structure while also introducing Agile-inspired sprints. Early stages (defining objectives, architectural planning) often follow a more traditional format, since they involve upfront analysis and decision-making that benefits from a linear flow. By contrast, Stages 3–6 each produce a tangible sub-product through **iterative and incremental development**, allowing for more frequent feedback and course corrections. Meanwhile, the Gantt chart ensures that we maintain a clear visualization of deadlines, dependencies, and overall progress. Through this combination, the team can maintain a high-level roadmap and clear deadlines (Gantt), while also integrating iterative sprints in the phases where rapid iteration offers the greatest value. In addition, it is possible to display the Gantt chart on multiple task depths which makes it **suitable for all kinds of interested parties** such as Stakeholders or Development teams.

When preparing the Work Breakdown Strucutre (WBS), I applied the MECE (Mutually Exclusive, Collectively Exhaustive) principle to ensure that each stage is discrete (no task overlap) and that together they cover the project's full scope. I also designed the plan so that each stage corresponds to a distinct deliverable.

To keep the tasks easy to understand and consistent, I described them using noun-based labels (e.g., "Data Collection Implementation," "Model Integration") rather than verb phrases. This makes it clearer what outcome each task produces. This hybrid approach, iterative yet managed via a Gantt chart, **balances flexibility in responding to changes with a well-defined roadmap** and milestones for the entire project.

## 2.2 Roles of the team members

I also want to briefly describe my understanding of the roles of each team member following which I assigned the tasks.

**R1, R2** - both researchers are experts in the field, particularly skilled in finding ways of solving problems. In addition, they are helpful during project's implementation often working with the support of the Senior Developer (SD)

**SD** - experienced developer, who is able to implement the most challenging tasks

**TL (Team Lead)** - also an experienced developer, who in addition to his/her technical skills possesses managerial skills. TL is involved in critical decision making, is mainly responsible for the project outcome and flow, while also being a representative of the project, meeting with stakeholders and presenting the achieved results. He/She occasionally helps with the implementation

**PM (Project manager)** - In this case, inexperienced project manager responsible for creating and modifying project's backlog. PM is a less technical role, often documents the results, participates in the team's meetings but does not often participate in the project implementation. He/She supervises the project alongside TL and ensures constant progress

## 2.3 Gantt chart

### 2.3.1 Key takeaways

- Each task is assigned an ID, Name, Duration, Owner(s), Priority and Dependencies
- Some tasks have multiple owners
- Priority is one of the following: Minor, Normal, Major, Critical, Show-Stopper as defined in YouTrack
- I made an assumption that **if a parent task is assigned a dependency all sub-tasks of this task also share this dependency**. I think this ensures clarity on the Gantt chart making it easier to comprehend
- Tasks which are parent tasks to other tasks do not have an estimated time, this time is derived from their subtasks

**Please access the Gantt chart according to the instructions provided in the assignment.**

**In case of any problems, I also provide screenshots of the chart and its table representation below**.

## 2.4 Project plan in table format

Table 1: Stage 1: Project's Definition & Requirements

| ID | Task | Duration | Owner(s) | Priority | Dependencies |
|---|---|---|---|---|---|
| 1.1 | Project's objectives and scope definition | 2d | Me, TL | Show-stopper | - |
| 1.2 | Project's objectives and scope presentation | 1d | Me, TL | Critical | 1.1 |
| 1.3 | Functional & non-functional requirements gathering | - | Me, TL, SD, R1, R2 | Show-stopper | 1.2 |
| 1.3.1 | Business analysis | 2d | Me, R1 | Major | - |
| 1.3.2 | Technical analysis | 2d | SD, R2, TL | Major | - |
| 1.3.3 | Requirements documentation | 1d | Me | Normal | 1.3.1, 1.3.2 |
| 1.4 | Constraints & initial risks identification | 2d | Me, TL, SD, R1, R2 | Critical | 1.3.1, 1.3.2 |
| 1.5 | Privacy measures identification | 1d | Me, TL, SD, R1, R2 | Critical | 1.4 |

Table 2: Stage 2: High-Level Architecture & Data Planning

| ID | Task | Duration | Owner(s) | Priority | Dependencies |
|---|---|---|---|---|---|
| 2.1 | System's architecture design | - | TL, SD, R1, R2 | Critical | Stage 1 finished |
| 2.1.1 | Trigger model - Code Completion model connection design | 2d | TL, SD | Major | - |
| 2.1.2 | FL technology choice | 2d | R1, R2 | Normal | - |
| 2.1.3 | System's workflow design | 2d | TL, Me | Normal | - |
| 2.2 | Data Aspects plan | - | SD, R1, R2 | Critical | Stage 1 finished |
| 2.2.1 | Data to gather choice | - | R1, R2 | Major | - |
| 2.2.1.1 | Model input choice | 1d | R1, SD | Normal | - |
| 2.2.1.2 | Model target choice | 1d | R2, SD | Normal | - |
| 2.2.2 | Data processing plan | 2d | R1, R2 | Normal | 2.2.1 |
| 2.2.3 | Data structure choice | 1d | R1 | Normal | 2.2.1 |
| 2.2.4 | Data storage approach | 1d | R2 | Normal | 2.2.1 |
| 2.3 | Model approaches choice | - | TL, SD | Critical | 2.1 |
| 2.3.1 | Server-side aggregation techniques choice | 3d | SD | Major | - |
| 2.3.2 | Model architectures choice | 3d | TL | Major | - |
| 2.4 | Architecture, privacy measures, overview & refinements | 1d | TL, SD, R1, R2, Me | Major | 2.1, 2.2, 2.3 |

Table 3: Stage 3: Basic Early Prototype Creation (Sprint 1)

| ID | Task | Duration | Owner(s) | Priority | Dependencies |
|---|---|---|---|---|---|
| 3.1 | Data collection implementation (inside IDE) | - | SD, R1 | Critical | Stage 2 finished |
| 3.1.1 | Development data collection environment preparation | 2d | SD, R1 | Major | - |
| 3.1.2 | Data collection scripts implementation | 2d | SD | Normal | 3.1.1 |
| 3.1.3 | Data processing scripts implementation | 2d | R1 | Normal | 3.1.1 |
| 3.1.4 | Data storing process implementation | 2d | SD | Normal | 3.1.1 |
| 3.1.5 | Data pipeline IDE integration | 2d | SD, R1 | Major | 3.1.1, 3.1.2, 3.1.3, 3.1.4 |
| 3.2 | Minimal local model implementation | - | TL, R2, R1 | Critical | Stage 2 finished |
| 3.2.1 | Initial model architecture choice (from the proposed) | 1d | TL, R2, R1 | Major | - |
| 3.2.2 | Model Coding | - | TL, R1, R2 | Major | 3.2.1 |
| 3.2.2.1 | Architecture coding | 3d | R1, R2 | Major | - |
| 3.2.2.2 | Training script coding | 2d | R1, R2 | Normal | 3.2.2.1 |
| 3.3 | Model integration | - | SD, R1, R2 | Critical | - |
| 3.3.1 | Model - Code Completion Integration | 2d | SD, R1 | Major | 3.2.2 |
| 3.3.2 | Model - Data pipeline (3.1) integration | 3d | SD | Normal | 3.3.1, 3.1 |
| 3.4 | Initial Data Collection | 2d | R2 | Major | 3.3 |
| 3.5 | Prototype tests and (user→local model→code completion→user) flow assurance | 3d | SD, R1, R2, TL | Critical | 3.4 |
| 3.6 | Tests documentation preparation | 1d | Me | Major | 3.5 |
| 3.7 | Overview meeting (Sprint review) | 1d | SD, R1, R2, TL, Me | Major | 3.6 |

Table 4: Stage 4: Federated Learning Integration (Sprint 2)

| ID | Task | Duration | Owner(s) | Priority | Dependencies |
|---|---|---|---|---|---|
| 4.1 | Model evaluation approach | - | R1, R2 | Major | Stage 3 finished |
| 4.1.1 | Metrics research & choice | 1d | R1, R2 | Major | - |
| 4.1.2 | Model performance scoring system design | 2d | R1, R2 | Normal | 4.1.1 |
| 4.2 | FL architecture implementation | - | SD, R1, R2, TL | Critical | Stage 3 finished |
| 4.2.1 | Central model implementation (according to the initial architecture) | 3d | SD | Major | - |
| 4.2.2 | Parameter sending logic implementation | 2d | SD, R1 | Major | 4.2.1 |
| 4.2.3 | Server-side aggregation approaches implementation | 3d | R2, TL | Major | 4.2.1 |
| 4.2.4 | Central model update logic implementation | 2d | SD, R1 | Major | 4.2.1 |
| 4.2.5 | Central (server) model - local models integration | 3d | TL, R1, R2 | Critical | 4.2.1, 4.2.2, 4.2.3, 4.2.4 |
| 4.3 | Integration testing (global model→local models→global model) | 4d | SD, R1, R2, TL | Critical | 4.1, 4.2 |
| 4.4 | Testing documentation preparation | 1d | Me | Minor | 4.3 |
| 4.5 | Training (multiple iterations) | - | R1, R2, SD | Major | 4.3 |
| 4.5.1 | Data Collection | 2d | R1, R2, SD | Major | - |
| 4.5.2 | FL training | 2d | R1, R2, SD | Normal | 4.5.1 |
| 4.6 | Performance evaluation & documentation | - | SD, R1, R2, Me | Major | 4.5 |
| 4.6.1 | Metrics results overview | 2d | R1, R2 | Normal | - |
| 4.6.2 | Initial aggregation approaches performance comparison | 2d | SD, Me | Normal | - |
| 4.7 | Overview meeting (Sprint review) | 1d | SD, R1, R2, Me, TL | Major | 4.6 |

Table 5: Stage 5: Remaining Models Implementation, Integration & Testing (Sprint 3)

| ID | Task | Duration | Owner(s) | Priority | Dependencies |
|---|---|---|---|---|---|
| 5.1 | Remaining model variants implementation | 5d | SD, R1 | Critical | Stage 4 finished |
| 5.2 | Pipeline - model integration | 4d | SD, R2 | Major | Partially in parallel with 5.1 (can start at some stage of 5.1) |
| 5.3 | Testing environment update (for new models) | 2d | SD | Major | Partially in parallel with 5.2 (can start at some stage of 5.2) |
| 5.4 | Integration testing (as in 4.4) | 3d | SD, R1, R2, TL | Critical | 5.3 |
| 5.5 | Testing documentation preparation | 1d | Me | Minor | 5.4 |
| 5.6 | Training (as in 4.6) | 4d | SD, R1 | Major | 5.4 |
| 5.7 | Results documentation | 2d | R2, Me | Normal | Partially in parallel with 5.6 (can start at some stage of 5.6) |
| 5.8 | Initial comparisons & observations | - | SD, R1, R2, TL | Major | 5.6 |
| 5.8.1 | Initial comparisons between models documentation | 1d | SD, R1 | Normal | - |
| 5.8.2 | Initial aggregation & model performance observations documentation | 1d | TL, R2 | Normal | - |
| 5.9 | Overview meeting (sprint review) | 1d | SD, TL, R1, R2, Me | Major | 5.8 |
| 5.10 | Potential Modification of the Model Pool | 1d | SD, TL, R1, R2, Me | Major | 5.9 |

Table 6: Stage 6: Extended Training & Testing (Sprint 4 and the Following Sprints)

| ID | Task | Duration | Owner(s) | Priority | Dependencies |
|---|---|---|---|---|---|
| 6.1 | More data gathering | 5d | SD, R1, R2 | Critical | Stage 5 finished |
| 6.2 | Extensive training & testing sessions execution | 10d | SD, R1, R2 | Critical | 6.1 |
| 6.3 | Performance documentation | 8d | R1, Me | Normal | Partially in parallel with 6.2 (can start at some stage of 6.2) |
| 6.4 | Approaches evaluation | 8d | R1, R2, SD, TL, Me | Normal | Partially in parallel with 6.2 (can start at some stage of 6.2) |
| 6.5 | Model architecture adjustments and hyperparameter tuning | 5d | SD, TL, R1, R2 | Major | Partially in parallel with 6.2 (can start at some stage of 6.2) |
| 6.6 | Overview meeting - Planning the next iteration OR Choosing the best performing model for production | 2d | SD, TL, R1, R2, Me | Critical | 6.2, 6.3, 6.4, 6.5 |

## 2.5 Screenshots of the Gantt chart

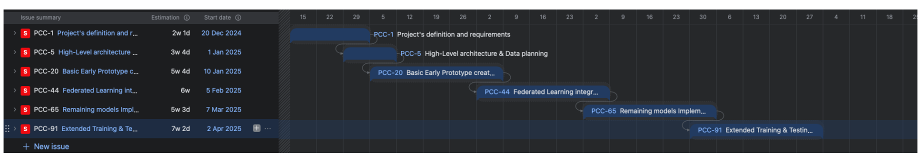


Figure 1: High-level view

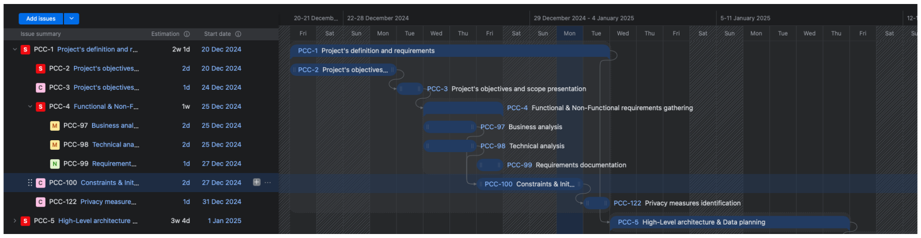


Figure 2: Stage 1

Figure 3: Stage 2

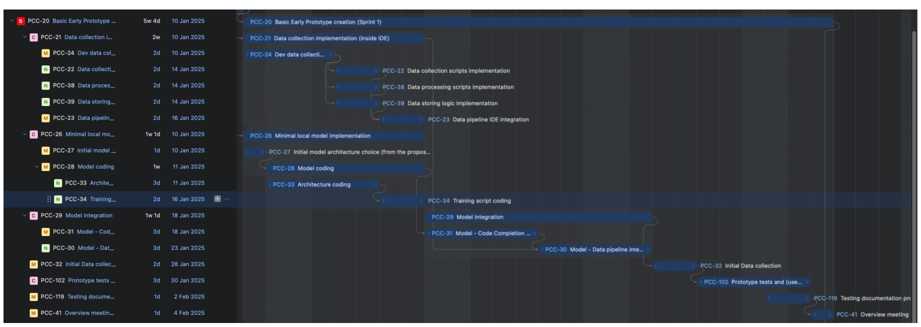

Figure 4: Stage 3

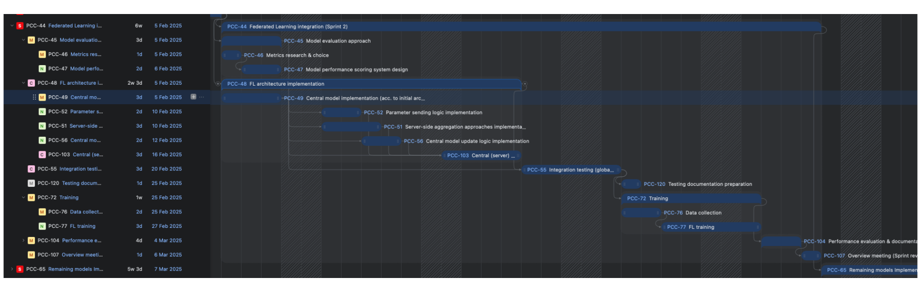

Figure 5: Stage 4

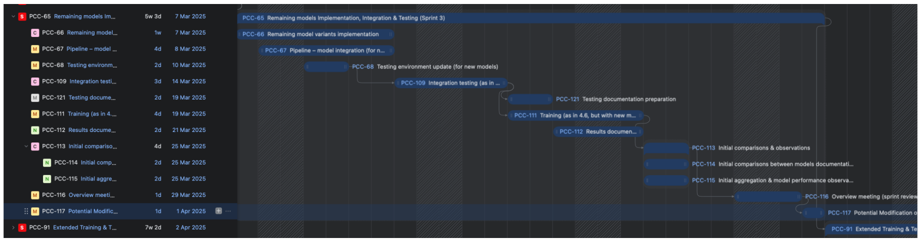

Figure 6: Stage 5

Figure 7: Stage 6

# 3 Task 3

## 3.1 Reflection on the situation

The main issue in this situation was the lack of critical safety information communicated to the legal team. The initial focus on data collection during the discussions between teams resulted in insufficient presentation of the planned safety principles. Because these principles were not properly presented or consulted with the legal team, they lacked relevant knowledge and comprehensive understanding. Consequently, they did not have the chance to suggest refinements and express privacy concerns at the very start of the project, when the changes would have had less impact on the overall project timeline.

These problems could have been easily prevented by actively involving the members of the legal team throughout the project. They should have participated at least after completing project's milestones and in the overview meetings when the work performed in the preceding stage or sprint was being summarized. This would enable the legal team to share observations and concerns, enabling the development team to update their work backlog to accommodate any legal guidelines. This modified approach would likely have prevented the safety concerns that caused delays, while also resulting in a safer product and boosting stakeholder confidence during development.

## 3.2 Proposed process improvements

To ensure that the legal team is thoroughly informed about all privacy details and risk mitigation strategies we could introduce a few key elements into our project's pipeline.

### 3.2.1 Initial risk identification and design involvement

The legal team should be included in the early stages of the project where various kinds of potential risks are identified and the system's design is planned. By combining development and safety guideline expertise from both teams, we can make more suitable design choices and reduce the need for late-stage safety refinements, leading to more efficient development.

### 3.2.2 Dedicated meetings held by the TL

Regardless of how well-thought the initial design is, it would almost certainly require modifications. Those modifications must be presented and explained to the legal team, if there is any suspicion they might cause security or privacy vulnerabilities. Since the need for modification can not be planned in advance it is vital to take into account that occasional not previously scheduled meetings would have to take place. The TL (possibly along with the PM) should schedule these meetings whenever necessary, then inform the development team(s) about the outcome, ensuring the project aligns with updated security guidelines.

### 3.2.3 Security and compliance documentation maintenance

One of the primary tools for effective communication with the legal team is maintaining detailed security and compliance documentation. Many security vulnerabilities can be overlooked in the process of software development and possibly cause similar obstacles and delays as presented in the problem. Proper documentation ensures that each meeting results in resolving any flaws in the system and helps keep both technical and legal team members on the same page.

### 3.2.4 Milestone and Stage/Sprint completion meetings engagement

As mentioned earlier, in addition to dynamically scheduled TL-Legal team meetings we should plan some of them at the beginning of the project in the phases where we are certain they will be required. These meetings would allow the team to present stage's or sprint's outcome along with the potentially modified security and compliance documentation. Formally finishing a stage of development if the legal team does not express any concerns, or in the opposite case modifying the existing approach such that the developed part of the product is fully compatible with applicable legal guidelines.