0.1 Design Decision

In the process of developing the prototype, we encountered numerous pivotal moments that required careful and strategic decision-making. These decisions were essential not only to address immediate technical challenges but also to align the project with its overall objectives.

We have tried or learned from previous studies several approaches for each of the challenges. So, in this chapter, we documented both the successful strategies and the ones that failed, along with the reasoning. This comprehensive record aims to provide valuable insights, enabling future researchers to avoid redundant efforts and build upon our findings effectively.

Table 1, Table 2, and Table 3 are made to document how each option (approach) is assessed and why certain decisions are made. The tables are written by following these standard: The table is divided into sections that address specific concerns, list ranking criteria, and provide options. The first column, titled ”Concern (Identifier: Description),” identifies the ethical concern being addressed. The second column, ”Ranking criteria (Identifier: Name),” lists the criteria used to evaluate the options. The third column, ”Options,” details the possible solutions, each identified by a unique name and description, followed by their evaluation against the ranking criteria. The ”Status” row indicates whether the option is accepted or rejected. The accepted options will be applied into the System Architecture (Section. **??**) of this prototype. The ”Rationale of decision” provides the reasoning behind each decision based on the evaluation against the criteria.

0.1.1 Ethic Baseline Model

This subsection compares several manual ethical model that potentially be the baseline ethical model of the prototype (Table 1). The ECCOLA Card (Vakkuri et al. 2021 and Halme et al. 2024) is chosen as the baseline for ethical questions used to build the prototype, due to its peer-reviewed, open-source guidelines specifically optimized for this field. These guidelines have been revised and published multiple times in peer-reviewed papers, ensuring they remain rigorous and credible. They encompass comprehensive ethical considerations that foster transparent and accountable decision-making processes (Table 1).

Table 1: Ethic Baseline Model

Concern (Identifier: Description)		*Con#1: Where do the ethical questions come from?*
Ranking criteria (Identifier: Name)		*Cr#1: Peer-reviewed and proven* *Cr#2: Open source* *Cr#3: Connected to Agile*
Options	***Identifier: Name***	***Con#1-Opt#1: UKRI***
	Description	https://www.ukri.org/publications/ethics-
	Status	*This option is **REJECTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: Unclear whether peer-reviewed and proven* *Cr#2: -* *Cr#3: Not specific to Agile project management*
	Rationale of decision	*The framework is not published in white source*
	Identifier: Name	***Con#1-Opt#2: AMC COC***
	Description	https://www.acm.org/code-of-ethics
	Status	*This option is **REJECTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: -* *Cr#2: -* *Cr#3: Not spesific for agile project management*
	Rationale of decision	*The framework is a bit unclear and did not come with a direct list of questions.*
	Identifier: Name	***Con#1-Opt#3: ECCOLA***
	Description	*Based on ECCOLA Card of Ville Vakkuri et al.*
	Status	*This option is **ACCEPTED**.*
	Relationship(s)	-
	Evaluation	*Cr#1: Peer-reviewed and proven* *Cr#2: Open source* *Cr#3: Optimized for software development management*
	Rationale of decision	*Fulfills all criteria*

0.1.2 Frontend and Backend Placement Configuration for Prototype Deployment

This subsection discusses the selection and justification of the environment where the transformation, front-end and back-end will be placed (Table 2).

The selected option for the end-user placement of the logic/server is a third-party LLM (Language Learning Model) with Python handling both the frontend and backend on the client desktop. This approach is resulting on a lightweight frontend and backend capable of running on a standard client PC while leveraging a third-party server for LLM suggestions.

This option is accepted for the following reasons. First, this option does not require cloud infrastructure, making it cost-effective and manageable. Second, as this option run in same machine aside from the LLM, this option is simplifying integration. Finally, for accessibil-

ity, the option offers an interactive user experience with quick processing times due to local execution.

Even though, the setup process requires some Python knowledge to run the user interface, but given that the target users are part of a software project management team with a high level of technical expertise, this criterion is of lower importance.

However, for data security, there is a potential risk of breaching non-disclosure agreements due to data transmission to the LLM server even though we have study the data privacy from the LLM provider (OpenAI n.d.(c) OpenAI 2023 OpenAI n.d.(b) OpenAI n.d.(c)). Despite this, since all components except the LLM operate locally, future enhancements could easily transition to a fully local setup, thereby minimizing technical debt of privacy protection.

Although the option does not fully satisfy all the criterias, the associated risks are deemed manageable. The potential breach of data security can be mitigated in future iterations, and the minor Python knowledge required is considered acceptable given the technical proficiency of the users.

Table 2: Frontend and Backend Placement Configuration for Prototype Deployment

Concern (Identifier: Description)		*Con#2: Which configuration of frontend and backend components should be used for the deployment of the prototype?*
Ranking criteria (Identifier: Name)		*Cr#1: Resource availability* *Cr#2: Data security* *Cr#3: Ease of connection with other components* *Cr#4: Ease of setup* *Cr#5: Accessibility*
Options	***Identifier: Name***	***Con#2-Opt#1: Python backend in server | Web frontend***
	Description	*Setting up a server with Python transformation and backend, and a web frontend. Clients can provide the API of their JIRA to the web.*
	Status	*This option is* ***REJECTED****.*
	Relationship(s)	
	Evaluation	*Cr#1: Cloud* *Cr#2: Might not align with NDA* *Cr#3: As the processing is on our side, it is easy to be accessed by other components* *Cr#4: Requires setting up the server again* *Cr#5: Easier for clients to access as they don't need to install any extra software*
	Rationale of decision	*This option does not fulfil the Cr#2, Cr#3, and Cr#4.*
Options	***Identifier: Name***	***Con#2-Opt#2: Python desktop app on client***
	Description	*Provide the open-source code to be run directly on the client's laptop. This option is feasible as the client consists of software development teams, so they should be able to run a Python script.*
	Status	*This option is* ***REJECTED****.*
	Relationship(s)	
	Evaluation	*Cr#1:Cloud* *Cr#2:Better Data Security* *Cr#3:Need internet to connect with other components* *Cr#4:No need server. But need to explain to user how to do it* *Cr#5:Harder accessibility, as we expect client to run the apps by themselves*
	Rationale of decision	*Con #1-Opt #2 has better data security. However, the available training data is insufficient to retrain a fast LLM that is light enough to run on a typical client PC.*

Options	Identifier: Name	***Con#2-Opt#3: Transformation, backend, and frontend all compacted in a Python desktop app on our server***
	Description	*Clients will provide us access to their JIRA API or send a CSV file containing their user stories, and we will send back the report.*
	Status	*This option is **REJECTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: Cloud* *Cr#2: Might breach NDA, but the data is processed in another party's LLM* *Cr#3: Easier to connect with other components as it runs in our system* *Cr#4: Easier to set up* *Cr#5: Harder to access and less interactive*
	Rationale of decision	*This option does not fulfill Cr#2, Cr#3, Cr#4, and Cr#5*
Options	Identifier: Name	***Con#2-Opt#4: Transformation with third-party LLM, frontend and backend developed in Python and run on the client desktop***
	Description	*Create a lightweight frontend and backend that can run on a typical client PC and perform LLM suggestions on a third-party server.*
	Status	*This option is **ACCEPTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: No need for cloud* *Cr#2: Might breach NDA due to passing data to LLM. However, since all components besides LLM run locally, it will be easier to move fully local in future enhancements.* *Cr#3: Easier to connect with other components as all besides LLM run on the same machine* *Cr#4: Requires a bit of Python knowledge to run the UI* *Cr#5: Interactive with UX, and the process will be fast as it runs locally*
	Rationale of decision	*This option does not fulfill Cr#2 and Cr#4. However, for Cr#2, it will be the smallest technical debt. For Cr#4, only basic Python knowledge is needed to run the application, and since the client for this prototype is a software project management team with higher technical knowledge, this criterion is not of high importance.*

0.1.3 Transformation algorithm

This subsection compares the transformation algorithms that we have tested and evaluates their suitability for achieving the prototype's goal (Table 3).

The first algorithms we considered were ANN Poplazarova et al. 2020. ANN should be more accurate as it has been validated by a big company, based on the paper "Ethical decision-making in biopharmaceutical research and development applying values using the TRIP/TIPP model". The model is accessible via API and has been previously proven. But, it is only work for certain context.

Secondly, sentiment analysis, it can be done locally. But, it has no proof of accuracy and has not been previously proven. It does not have pretrained models and the technique is quite old as well.

Next, we tried several Large-Language Model (LLM). The script we used for comparison are all available on the Github(cinapr 2024).

The first LLM is GPT4ALL. It utilizes several HuggingFace pre-trained models as the base. The models were chosen based on their performance recommendations on the Hugging Face website. The inclusion criteria included models that were in English, lightweight, and capable of running on the device. Therefore, four models were selected:

```
mistral-7b-openorca.gguf2.Q4_0.gguf (424.12 seconds)
Nous-Hermes-2-Mistral-7B-DPO.Q4_0.gguf (544.64 seconds)
orca-mini-3b-gguf2-q4_0.gguf (152.23 seconds)
replit-code-v1_5-3b-newbpe-q4_0.gguf (0.96 seconds)
```

Due to the limited availability of device resources, instead of fine-tuning and comparing them, we first asked a simple question to compare how quickly each model could respond. When asked, *"What is London?"*, the `orca-mini-3b-gguf2-q4_0.gguf` model responded in 152.23 seconds, while `Nous-Hermes-2-Mistral-7B-DPO.Q4_0.gguf` and `mistral-7b-openorca.gguf2.Q4_0.gguf` responded in 544.64 seconds and 424.12 seconds, respectively. Unfortunately, the `replit-code-v1_5-3b-newbpe-q4_0.gguf` model was unloadable due to the wrong number of tensors.

Even though this approach was considered the best for data privacy concerns since it runs locally, it was deemed unsuitable due to the excessive time taken to answer basic questions. The fastest response took more than two minutes, which is impractical for our prototype that needs to loop through numerous user stories.

Next, we compared GPT2. GPT2 is run on a local server. It took 1405.1124 seconds to be trained, and the classification compared to the ECCOLA sample dataset returned a 3.61% similarity. On average, the LLM took 0.195 seconds to classify each user story sample, showing better speed performance than GPT4ALL. Unfortunately, its reasoning is still weak, and future fine-tuning is not feasible as the basic fine-tuning already consumes more than 80% of the CPU and RAM of the available machine (Figure 1).

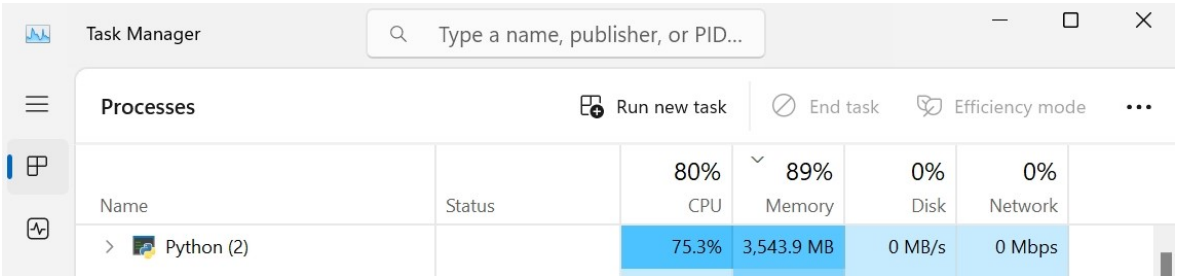

Figure 1: Task Manager Screenshot Showing GPT2 Fine-Tuning Consumption

The next algorithm we compared, after fine-tuning, was BART. The base model for BART is not located on the client machine. It took 2059.9524 seconds to be trained, and the classification compared to the ECCOLA sample dataset returned a 7.22% similarity. It showed better speed performance than GPT4ALL. Unfortunately, its reasoning remains weak, and future fine-tuning is not possible since the basic fine-tuning consumes more than 90% of the CPU and RAM of the available machine (Figure 2). Moreover, compared to GPT2, BART still falls short as the base model is hosted on an external server.

The fourth LLM algorithm we compared was GPT-3.5, which runs on a server owned by OpenAI. It took 963.47 seconds to be trained, and the classification compared to the ECCOLA sample dataset returned 26.32%. Although it lacks in terms of privacy, as it runs on an OpenAI server, we have reviewed the privacy policies provided by OpenAI (OpenAI n.d.(a)OpenAI 2023OpenAI n.d.(b)OpenAI n.d.(c)). Since the LLM process does not run on a local machine, this algorithm offers the best performance in terms of speed and min-

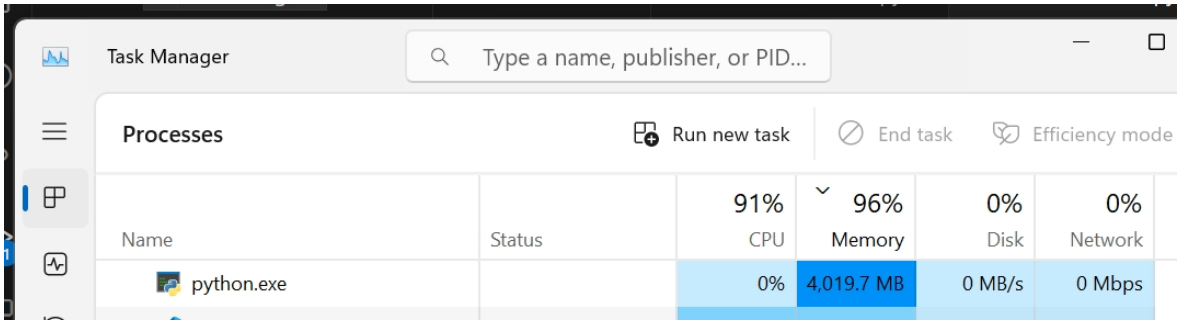

Figure 2: Task manager screenshot that showing BART fine-tuning consumption

imal CPU and RAM consumption. Therefore, better reasoning can be expected from this algorithm.

From the perspective of machine availability, speed, and project timeline, we chose GPT-3.5, which runs on an OpenAI server. However, we recommend that future researchers with better machine or server availability consider deeper fine-tuning of GPT2 to improve its reasoning skills. GPT2, which runs locally, would be better for user story data privacy.

Table 3: Transformation Algorithm

Concern (Identifier: Description)		*Con#3: Which transformation algorithm should be chosen for generating the suggestion?*
Ranking criteria (Identifier: Name)		*Cr#1: Speed of response* *Cr#2: Data privacy* *Cr#3: Accuracy of classification* *Cr#4: Resource usage* *Cr#5: Previously proven* *Cr#6: Model is openly distributed*
Options	***Identifier: Name***	***Con#3-Opt#1: GPT-4ALL with HuggingFace Pre-trained Models***
	Description	*Using GPT-4ALL with several pre-trained models from HuggingFace based on performance recommendations. It is an open-source alternative to OpenAI GPT.*
	Status	*This option is* ***REJECTED****.*
	Relationship(s)	
	Evaluation	*Cr#1: Response times varied from 152.23 to 544.64 seconds, which is too slow for practical use.* *Cr#2: Ensures data privacy as processing is local.* *Cr#3: Accuracy not thoroughly evaluated due to response time limitations.* *Cr#4: Requires significant resources to run on local machines.* *Cr#5: Not previously proven in similar contexts.* *Cr#6: Models are openly distributed.*
	Rationale of decision	*This option, while excellent for data privacy and openness, is unsuitable due to slow response times and high resource usage, making it impractical for the prototype's needs.*
Options	***Identifier: Name***	***Con#3-Opt#2: GPT2***
	Description	*Testing GPT-2 for performance and classification accuracy.*
	Status	*This option is* ***REJECTED****.*
	Relationship(s)	
	Evaluation	*Cr#1: Faster response time compared to GPT-4ALL models.* *Cr#2: Limited by local machine capabilities, impacting fine-tuning.* *Cr#3: -* *Cr#4: High resource consumption for fine-tuning.* *Cr#5: Proven in various NLP tasks but not specifically for this use case.* *Cr#6: Openly distributed.*
	Rationale of decision	*Despite better speed and openness, the high resource consumption prevents it from being further fine-tuned.*
Options	***Identifier: Name***	***Con#3-Opt#3: BART***
	Description	*Testing BART model for performance and classification accuracy.*
	Status	*This option is* ***REJECTED****.*
	Relationship(s)	
	Evaluation	*Cr#1: Slow training time but better response time.* *Cr#2: Depends on an external server, affecting data privacy.* *Cr#3: Higher accuracy compared to GPT-2.* *Cr#4: High resource usage.* *Cr#5: Proven in various contexts.* *Cr#6: Openly distributed, with APIs that run on HuggingFace server and downloadable models that run locally.*
	Rationale of decision	*Better accuracy and proven effectiveness but still impractical due to server dependence and high resource usage for local fine-tuning.*

Options	Identifier: Name	***Con#3-Opt#4: GPT-3.5 (API)***
	Description	*Testing the GPT-3.5 model for performance and classification accuracy. Based on the paper: "A Survey on Large Language Model-Based Autonomous Agents."*
	Status	*This option is **ACCEPTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: Fast response time suitable for the prototype.* *Cr#2: Data privacy concerns due to server processing.* *Cr#3: Expected better reasoning and accuracy.* *Cr#4: Low local resource consumption.* *Cr#5: Proven effective in various applications.* *Cr#6: Not openly distributed.*
	Rationale of decision	*Despite data privacy concerns, the fast response time, low resource usage, and proven effectiveness make GPT-3.5 the best option for this prototype.*
Options	Identifier: Name	***Con#3-Opt#5: ANN***
	Description	*Based on the paper: "Ethical Decision-Making in Biopharmaceutical Research and Development Applying Values Using the TRIP TIPP Model."*
	Status	*This option is **REJECTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: Expected to be accurate but no direct speed measurements.* *Cr#2: Requires external uploading, impacting data privacy.* *Cr#3: Proven in biopharmaceutical research contexts.* *Cr#4: Requires API access.* *Cr#5: Proven effective in specific contexts.* *Cr#6: Not openly distributed.*
	Rationale of decision	*While it shows promise in accuracy, it has been proven only in certain contexts.*
Options	Identifier: Name	***Con#3-Opt#6: Sentiment Analysis***
	Description	*Sentiment analysis using traditional techniques.*
	Status	*This option is **ACCEPTED**.*
	Relationship(s)	
	Evaluation	*Cr#1: No proof of speed and accuracy.* *Cr#2: Ensures data privacy as it can be done locally.* *Cr#3: Not proven in relevant contexts.* *Cr#4: Low resource requirements.* *Cr#5: Not previously proven in relevant contexts.* *Cr#6: No pre-trained models found.*
	Rationale of decision	*While data privacy and resource usage are advantages, the lack of proven accuracy and the outdated technique make it less desirable.*