

# Problem Statement and Goals

## Software Engineering

### Team 4, EcoOptimizers

Nivetha Kuruparan  
Sevhena Walker  
Tanveer Brar  
Mya Hussain  
Ayushi Amin

Table 1: Revision History

Date	Developer(s)	Change
September 18th, 2024	All	Created first draft of document
September 23rd, 2024	All	Finalized document
March 24th, 2025	Ayushi Amin	Fixed all grammer and spelling
March 24th, 2025	Tanveer Brar	Updated output section
March 24th, 2025	Sevhena Walker	Removed mentions of preserving code functionality through testing
March 24th, 2025	Mya Hussain	Removed mentions of database storing
March 24th, 2025	Ayushi Amin	Fixed formatting and grammar/spelling
March 24th, 2025	Sevhena Walker	Update the aspects of the plugin
April 4th, 2025	Ayushi Amin	Removed reinforcement learning from current plan

# 1 Problem Statement

## 1.1 Problem

The Information and Communications Technology (ICT) sector is currently responsible for approximately 2-4% of global CO<sub>2</sub> emissions, a figure projected to rise to 14% by 2040 without intervention (Belkhir and Elmeligi, 2018). To align with broader economic sustainability goals, the ICT industry must reduce its CO<sub>2</sub> emissions by 72% by 2040 (Freitag and Berners-Lee, 2021). Optimizing energy consumption in software systems is a complex task that cannot rely solely on software engineers, who often face strict deadlines and busy schedules. This creates a pressing need for supporting technologies that help automate this process. This project aims to develop a tool that applies automated refactoring techniques to optimize Python code for energy efficiency while preserving its original functionality.

## 1.2 Inputs and Outputs

**Inputs:** Source code that requires refactoring for energy efficiency.

**Outputs:** Refactored code with reduced energy consumption, along with energy consumption metrics.

## 1.3 Stakeholders

### Direct Stakeholders

1. **Software Developers:** They will be the primary users of the refactoring library as they will be the ones to integrate the library into their code for better refactoring.
2. **Dr. Istvan David (Supervisor):** Dr. David has a vested interest in the development of the system. He will play a crucial role in guiding and mentoring our team throughout the project. As the project supervisor, he will be the key advisor, offering feedback on technical aspects, project management, and research methodologies.
3. **Business Sustainability Teams:** These teams are responsible for considering how a company practices affect the environment. They will especially be interested in viewing the metrics provided by the library on how it improves the energy efficiency of software over time, therefore decreasing the burden on hardware and minimizing the company's environmental footprint.

### Indirect Stakeholders

1. **Business Leaders:** They focus on reducing operational costs associated with energy consumption, especially in large-scale or cloud-hosted applications. Use of the library in their products allows them to better achieve those goals.
2. **End Users:** While not directly affected by this refactoring library, end users of technology that use the library will benefit from more responsive and efficient software that consumes less power, especially in mobile, embedded, or battery-dependent applications.
3. **Regulatory Bodies:** They enforce energy consumption and sustainability standards, and ensure that software adheres to environmental regulations and may certify tools that meet efficiency requirements. Their oversight promotes the adoption of energy-efficient software practices.

## 1.4 Environment

### Development Frameworks and Tools:

1. *GitHub* will be used for version control and for CI/CD integration to automate refactoring processes.
2. *Visual Studio Code* will be the IDE used.

## 2 Goals

Our goal is to reduce the energy consumption of Python codebases during execution. We plan to achieve this by developing two core components:

1. **Refactoring Library:**

The library refactors inefficient code patterns to achieve a net reduction in energy consumption of the given codebase. When there are multiple ways to refactor a block of code, it will choose the one with the greatest net reduction in energy consumption.

2. **Plugin:**

The plugin will utilize the refactoring library so that developers can get access to a refactored and energy-efficient version of their Python codebase within an IDE. The plugin has three main aspects:

- **Detecting and Viewing Smells:** The plugin will identify code smells in the Python codebase that contribute to inefficient energy consumption and allow developers to view these smells directly within the IDE.
- **Initiating Refactoring and Viewing Changes:** The plugin will enable developers to initiate refactoring of the identified code smells and view the resulting changes to the codebase.
- **Viewing Energy Savings:** Developers will be able to see the cumulative energy savings achieved in the codebase over time as a result of the refactorings.

## 3 Stretch Goals

1. **GitHub Integration for DevOps Pipelines:**

- **Automated Refactoring:** We aim to build a feature that can be integrated into GitHub's DevOps pipelines, which would automatically refactor Python code present in the pipeline to a more energy-efficient version.
- **Test Compatibility:** This feature will work in tandem with the user's test suites to ensure that the refactored code maintains its original behaviour. This is done to ensure smooth adoption within CI/CD workflows.

2. **Reinforcement Learning Model:**

The goal is to build a model that evolves its refactoring recommendations over time to maximize energy efficiency while aligning with developer preferences. The model achieves this by:

- Incrementally identifying energy optimization techniques from past refactorings that result in larger net reductions in energy consumption.
- Prioritizing energy-efficient refactorings that align with the developer's personal preferences, such as a preference for fewer lines of code or specific coding styles. The model will account for the acceptance or rejection of refactorings, tailoring future suggestions to suit individual developer styles.

## 4 Challenge Level and Extras

The expected challenge level of our project is general. This is due to the relatively straightforward technical knowledge required for its completion. The project primarily involves applying known software optimization and refactoring techniques, which are well-documented and accessible. Additionally, the

required programming knowledge is in Python which is known by all of the team members and was taught in the undergraduate courses. Although the project does involve substantial development and research components, it is anticipated the overall scope and depth of the work is to be manageable within the given timeframe.

To further enhance the project and address any potential gaps in the challenge level, we propose two additional activities: User Documentation and Usability Testing. These extras will allow us to provide support for future users of the tool and ensure that the tool meets user expectations and accessibility standards.

Approval of the challenge level and extras will be discussed with the instructor, and adjustments may be made as needed throughout the term.

## Appendix — Reflection

### Mya Hussain Reflection

1. *What went well while writing this deliverable?*

Our team was able to create a clear and concise plan as to what we were trying to achieve and investigate the details as to how we could potentially achieve it. We were able to brainstorm multiple possible routes to take depending on how successful we are while developing. We were also able to reach out to our supervising professor and receive many research papers that helped bring us up to speed as to what this project would require, so that we could set out realistic goals for the project.

2. *What pain points did you experience during this deliverable, and how did you resolve them?*

A lot of time was taken for the research papers we requested from our supervisor to actually hit our inboxes. In the meantime, we were on our own to brainstorm solutions for the problem. This was both a positive and a negative as on the positive side we got to do some self-study and think of some new ideas from scratch. While on the downside, we didn't know if our brainstormed ideas were methodologies our supervisor would want us to use. Learning with mentorship is great because we were able to gain confidence that what we were doing was correct due to our supervisor's experience. Being able to explore the project on our own was cool because we got to investigate a lot of different technologies surrounding the space regardless of how efficient the solution may be. So, in the end, our pain points were just learning experiences, before we were thrown a life vest.

3. *How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?*

My team and I started the project off by getting very familiar with each other's strengths and weaknesses. We shared resumes, transcripts and highlighted all the former experiences we had before selecting a project that most aligned with our combined skillsets. Once we had a project, setting goals was the easy part as in our goals we ensured that we would have fallbacks if the project was overly ambitious allowing us to jump with a safety net. For example, we initially discussed the potential for developing a neural network that chooses the best refactoring for a given code input. Recognizing that we have low ML experience between us we opted to implement an algorithmic approach for our POC and attempt an ML implementation in the second semester if the first one works properly. We also ensured that there would be middlemen libraries we could pull from if we were unsuccessful in writing all the code ourselves. The clearest example of this is there being smell detection code readily available in many online free to use libraries. Planning out these fallbacks between ourselves made sure everyone was comfortable with the project and that we could tweak the difficulty as time went by being that we are trying something new.

### Ayushi Amin Reflection

1. *What went well while writing this deliverable?*

Writing the deliverable went quite well overall. The project itself was well defined and we were able to meet with the industry supervisor which helped clarify most of the details. Our team worked well together which helped each member to clarify any concerns particularly about the database we decided to use. Our team had some good discussions regarding making decisions on what languages and tools to use in the project such as our decision to use MySQL for our database.

2. *What pain points did you experience during this deliverable, and how did you resolve them?*

Not all the information was clarified or given by the supervisor at an early stage so writing this document took some time which caused us to delay working on this for a couple days. The team and I decided to consistently check in with the supervisor to encourage him to provide the required documents and information we needed as soon as possible. We used email as well as discord to directly reach out to the supervisor in which we finally ended up getting the information two days before the deadline and then crammed to finish the milestone documents.

### Sevhena Walker Reflection

1. *What went well while writing this deliverable?*

Right off the bat, I want to say that this deliverable was characterized by a great sense of *discovery*. I learnt more about my team members and what their goals are. I learnt more about the project we had chosen and our supervisor's reasoning behind proposing it. I learnt new information on topics that were novel to me. This deliverable has filled me with a sense of excitement and a budding confidence that I hope will only continue to grow.

There is much to learn about the project my team has chosen. Being able to go through the process of breaking it down and identifying what technologies will be used, and what features our system will have makes our project more concrete. Compared to the start of the project when things seem more vague and daunting, there remains a sense that we will achieve what we have set out to do.

2. *What pain points did you experience during this deliverable, and how did you resolve them?*

Of course, not everything is just sunshine and roses. Having to deconstruct a project with concepts that you are only vaguely familiar with can be an overwhelming task. While we were able to seek guidance from our supervisor, the onus was still on us to slowly unravel this new mass of information and piece it together in a coherent and structured manner. Our supervisor has been out of country and in a different time zone which has led to a gap in discourse. There were times when our team was left scratching our heads trying to figure out what the best way or, at the very least, what would be a *suitable* way to plan out our project.

### Nivetha Kuruparan Reflection

1. *What went well while writing this deliverable?*

Writing this deliverable went smoothly and was an important learning experience for the team. Meeting with our project supervisor early on helped us clarify the project's scope and objectives, making it easier to define the problem and outline the technical aspects. We had many discussions about which tools and technologies to use, and these conversations helped solidify key decisions, like selecting Python for the development language. By collaborating on this deliverable, it gave us a better understanding of our team's strengths and priorities and sets a positive tone for the project moving forward.

2. *What pain points did you experience during this deliverable, and how did you resolve them?*

During this deliverable, we faced a few challenges related to efficiently planning a large software project. One of the main pain points was learning how to set realistic goals that aligned with our team's capabilities and timeline. Initially, there was a temptation to create our own refactoring library from scratch in Rust (where no team member has experience in coding), which felt ambitious and overwhelming as we considered the scope of the project.

To resolve this, we decided to utilize the resources and guidance provided by our supervisor. By using existing tools and libraries, we can focus on ensuring that the refactoring processes

work effectively without getting bogged down in the complexities of building everything from the ground up. This experience reinforced the importance of planning when tackling large projects.

### **Tanveer Brar Reflection**

1. *What went well while writing this deliverable?*

This deliverable has set a great starting point for us as the prompts instigated multiple discussions among each other. This document will serve as a great point to begin discussing the architecture for our project.

2. *What pain points did you experience during this deliverable, and how did you resolve them?*

After having had two meetings with our Supervisor and multiple discussions with team members, I had the impression that we have a clear idea of what our goals are. However, when writing down the Goals and Stretch Goals, it was clear that our idea had gaps of ambiguity. While the development plan clarified multiple areas of initial project setup and other logistics for the team, I am glad that this document helped us identify certain aspects of our project that were unclear and needed additional discussion. After listening to everyone's idea about the ambiguous components and doing additional research, I was able to clearly document the goals.

### **Group Answers Reflection**

1. *How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?*

We carefully adjusted our project scope by focusing on the most important goals and prioritizing features that mattered most, while also setting clear milestones to keep us on track. The feedback we got from our supervisor helped us refine our goals, making sure we were ambitious but also realistic about what we could achieve.

## **References**

- Lotfi Belkhir and Ahmed Elmeligi. Assessing ict global emissions footprint: Trends to 2040 and recommendations. *Journal of Cleaner Production*, 2018. URL [https://www.researchgate.net/publication/322205565\\_Assessing\\_ICT\\_global\\_emissions\\_footprint\\_Trends\\_to\\_2040\\_recommendations#:~:text=in%20industrial%20development.-,...,%25%20by%202040%20%5B11%5D%20](https://www.researchgate.net/publication/322205565_Assessing_ICT_global_emissions_footprint_Trends_to_2040_recommendations#:~:text=in%20industrial%20development.-,...,%25%20by%202040%20%5B11%5D%20).
- Charlotte Freitag and Mike Berners-Lee. The real climate and transformative impact of ict: A critique of estimates, trends, and regulations. *Patterns Volume 2, Issue 9*, 2, 2021. doi: 10.1016/j.patter.2021.100340. URL <https://www.sciencedirect.com/science/article/pii/S2666389921001884#:~:text=If%20the%20ICT%20sector%20should,these%20sectors%20will%20have%20to>.