Reflection and Traceability Report on FBP CT Image Reconstruction

Qianlin Chen

April 18, 2025

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

Feedback from teammates and reviewers highlighted several areas for improvement, which were promptly addressed:

Issue 4

One reviewer pointed out that Section 3.1 lacked a visual system context diagram. This feedback was extremely helpful, and I addressed it by adding a diagram showing the relationship between users, external systems, and the CT reconstruction application. This addition made the architecture clearer to both technical and non-technical readers.

Issue 5

Another comment emphasized that assumptions listed in Section 4.2.1 were inconsistently referenced in the Traceability Matrix. In response, I carefully cross-linked each assumption (A1, A2) with all relevant sections, including Theoretical Models (TM), Data Definitions (DD), and Instance Models (IM). This improved the internal consistency and traceability of the document.

Issue 6

The peer suggested that the table in Section 4.2.7 lacked software-specific constraint details, such as frequency limits for filters. Based on this, I extended the description to clearly specify expected input ranges and system handling for out-of-bound values.

Issue 7

The maintainability NFR in Section 5.2 mentioned a threshold of "10% of the original development time," but lacked a formal definition of "FRACTION." I added this constant to the Values of Auxiliary Constants section and clarified it in the NFR description.

Issue 8

A reviewer noted inconsistent naming conventions in Section 4.2.2, where "Radon Transform" was occasionally written as "Randon Transform." These typos were corrected and naming across the document was standardized to avoid confusion.

1.2 VnV Plan and Report

Add more details and fix the error based on the final implementations.

1.3 Design and Design Documentation

1.3.1 MIS

From the peer review:

Issue 11

A reviewer recommended replacing variable names like theta with their corresponding mathematical symbols. This suggestion aligned with the intent of MIS to be a mathematical document rather than a programming specification. I updated all such instances to use proper math symbols, improving clarity and formatting consistency.

Issues 12 and 14

Fix the minor syntax error.

Issue 13

A peer flagged that a function F appeared in a formula without being defined anywhere in the document. I resolved this by either removing the undefined function or adding an explanation for its role in the context of the filter operation. This helped reduce ambiguity and improved traceability.

Issue 15

One of the local functions was named inconsistently as padded_radon vs. padding_radon, and a return value center was unused in the subsequent documentation. After review, I corrected the function name and removed the redundant return to streamline the semantics and reduce confusion.

1.3.2 MG

Issue 16

It was noted that some modules, particularly M5 and M6, were missing explicit type declarations (e.g., Abstract Data Type, Abstract Object, etc.). After reviewing their roles, I added the appropriate classifications to both modules. M5 was marked as a Service Module, and M6 as an Abstract Data Type, reflecting their usage patterns and the level of abstraction they represent.

Issue 17

The Filters Module (M3) was originally labeled as an Abstract Data Type. A peer reviewer correctly pointed out that it should be classified as an Abstract Object instead, since it encapsulates behavior but maintains no internal state. This was an insightful distinction, and updating the classification made the design documentation more precise and aligned with the standard MG template.

2 Challenge Level and Extras

2.1 Challenge Level

General level.

2.2 Extras

video guide

3 Design Iteration (LO11 (PrototypeIterate))

The initial design focused solely on implementing the Filtered Back Projection (FBP) algorithm via a command-line interface. However, early feedback from users indicated that visual interaction and easier parameter tuning were highly desired, especially for debugging and educational purposes. This led to the integration of a Tkinter-based GUI.

Another round of usability feedback during internal demo sessions highlighted confusion about input file formats and lack of result comparison. In response, I redesigned the GUI to clearly label required formats (e.g., .png, .h5) and added a dual display feature for original and reconstructed images side by side. We also included error pop-ups to guide users if invalid files were uploaded. Each iteration incorporated user feedback to better align the interface and features with end-user needs.

4 Design Decisions (LO12)

Several core design decisions were driven by technical constraints and user expectations:

- Limitations: I chose not to include advanced CT algorithms (like iterative reconstruction) due to project scope and performance limits on average machines.

- Assumptions: I assumed input images would be relatively clean, and that users would be familiar with basic imaging formats.

- Constraints: Portability was important, so I used Python and standard libraries (e.g., NumPy, Matplotlib) to avoid OS-specific dependencies.

Choosing ttkbootstrap for styling allowed us to maintain a modern interface without writing complex CSS or JavaScript. Separating logic into service modules also helped with future extensibility, which was important for users who may want to swap filters or preprocessing steps.

5 Economic Considerations (LO23)

While the CT reconstruction app is primarily academic, it holds potential for educational and training use in medical physics or image processing courses. If commercialized, the product could be positioned as a lightweight, open-source toolkit with optional premium features like advanced reconstruction methods or GPU acceleration.

Since the product is already released on GitHub, I would focus on community building and open-source visibility via documentation, example datasets, and outreach to imaging research groups.

6 Reflection on Project Management (LO24)

As a solo project, I followed my initial plan closely. I maintained a consistent workflow using GitHub for version control and issue tracking. Feedback was received from the instructor and a peer reviewer, which I scheduled into my timeline after each deliverable.

6.1 How Does Your Project Management Compare to Your Development Plan

Working alone gave me flexibility in task prioritization. I was able to move quickly, and GitHub helped me document all progress systematically. Reviewer feedback was clear and easy to integrate.

6.2 What Went Well?

Time estimation for testing and writing formal documents like MIS was off. I spent more time than expected fixing LaTeX formatting and clarifying notation.

6.3 What Went Wrong?

I'd set earlier deadlines for documentation drafts and automate more of the formatting checks. I'd also request feedback earlier instead of waiting for the final round.

6.4 What Would you Do Differently Next Time?

I'd set earlier deadlines for documentation drafts and automate more of the formatting checks. I'd also request feedback earlier instead of waiting for the final round.