

# Reflection Report on ImgBeamer

Joachim de Fourestier

April 10, 2023

## 1 Changes in Response to Feedback

### 1.1 SRS

This course really drove me to better explain and share my thoughts. I realize now that there are never too many figures. A good visual explanation really does say more than words could express. In response to some confusion, I create many figures that are now included in the SRS with many visual examples with a concise explanation for each of them. The wording for the problem and its solution was made a clearer. I had many minor issues with typos, inconsistencies with acronyms, and some missing and incomplete definitions. Following the "mantra" of "design for change", I realize the importance of making requirements and assumptions as atomic as possible: not only is it easier to reference, but it is easier to update.

### 1.2 Design and Design Documentation

By separating the responsibilities of the software into so-called modules, the code was updated to match and is now much clearer than it was before. As a result, it was easier for me to understand and see what other changes were needed to improve maintainability. Needed changes (or identified flaws) that were not completed (or fixed) were at least noted in the code as "to-do" comments. Otherwise, some global variables were not even used and have been removed.

### 1.3 VnV Plan and Report

Direct feedback helped make a clearer testing plan that I then followed myself. Had I not taken the time to address the changes to the VnV plan, I likely would not have found the several inconsistencies and small issues as noted in the VnV report. I cannot imagine how much worse it could have been otherwise, and if someone else had done the testing without these changes or the knowledge I have. The tests should be numerous and each input and output needs to be justified. It is important to include tests even if we *know* the "answer", since more often than not we are can be wrong and miss things... which I did ;-)

## 2 Design Iteration

The first version was very rudimentary. It was only after multiple interactions with the "expert consultants" that the needs became clearer for all parties involved. This really goes to show what Dr. Parnas meant why a "rational design" process needs to be "faked". Some requirements and even assumptions do not become clear till much later on. The original design was going to be a simple input-output program with two numbers: a pixel size and a relative spot size. This has now evolved into an interactive visualization program that allows for different spot shapes and display each part of the sampling process.

## 3 Design Decisions

The original software was likely going to be in python due to the large repository of helpful libraries and preloaded modules. However, the idea of a web application came to mind, because of the built-in cross-platform design and its incredible ease to sharing information / webpages. It is *by design*... During implementation, instead of just displaying the resulting image, it became clear that it was very beneficial to show each step of the sampling and image formation process.

## 4 Reflection on Project Management

One of the most difficult tasks for me, is to estimate how much effort and time some things take. As a software developer, it is always so easy to fall into the "trap" of feature creep. A scope needs to be well-defined, realistic, and justified: "done one thing and do it right". Over-committing and not having a realistic scope can ultimately lead to the failure of the whole project. In some sense, I guess this is about "modular" project planning or risk management, where smaller manageable tasks mean smaller manageable possible failures.

### 4.1 What Went Well?

By splitting up the design into separate documents with a clear purpose for each, it became clearer what was priority and what was not. This helped define the project scope. As for testing, I was relatively surprised what I could find. Even though most of the issues were minor, their sum effect lead to subtle but noticeable issues that I likely would have otherwise dismissed as simple "rounding issues" or "cosmic rays"...

### 4.2 What Went Wrong?

Naturally, I was overly ambitious with the original design with the time allotted. Much of the testing had to be limited and some features I wanted to implement had to be put aside, such as a coordinate system or noise simulation.

### 4.3 What Would you Do Differently Next Time?

I realize that it is sometimes better to go ahead with small but wrong design decisions and move forward than to hesitate. Not everything is knowable from the start. I need to accept that we will to go back and change things. Next time, I should gradually grow of a list of potential items ordered by priority, but **estimate** the time (with extra leeway) it takes for each. Then, take the pragmatic approach of limiting or truncating the list to the time allotted.