Optometry Lab Project

icuc

Emily, Eric, Liz, **Victoria**, Marcella, & Vlad

Coached by Steven Wu

Customer: Professor Austin Roorda

Agenda

- Project Overview
- Requirements
- Design Overview
- Technical Challenges and Solutions
- Agile Practices Checklist
- Tools Usage Checklist
- Customer Interaction
- Live Demo

Project Overview - A Greenfield App

Who

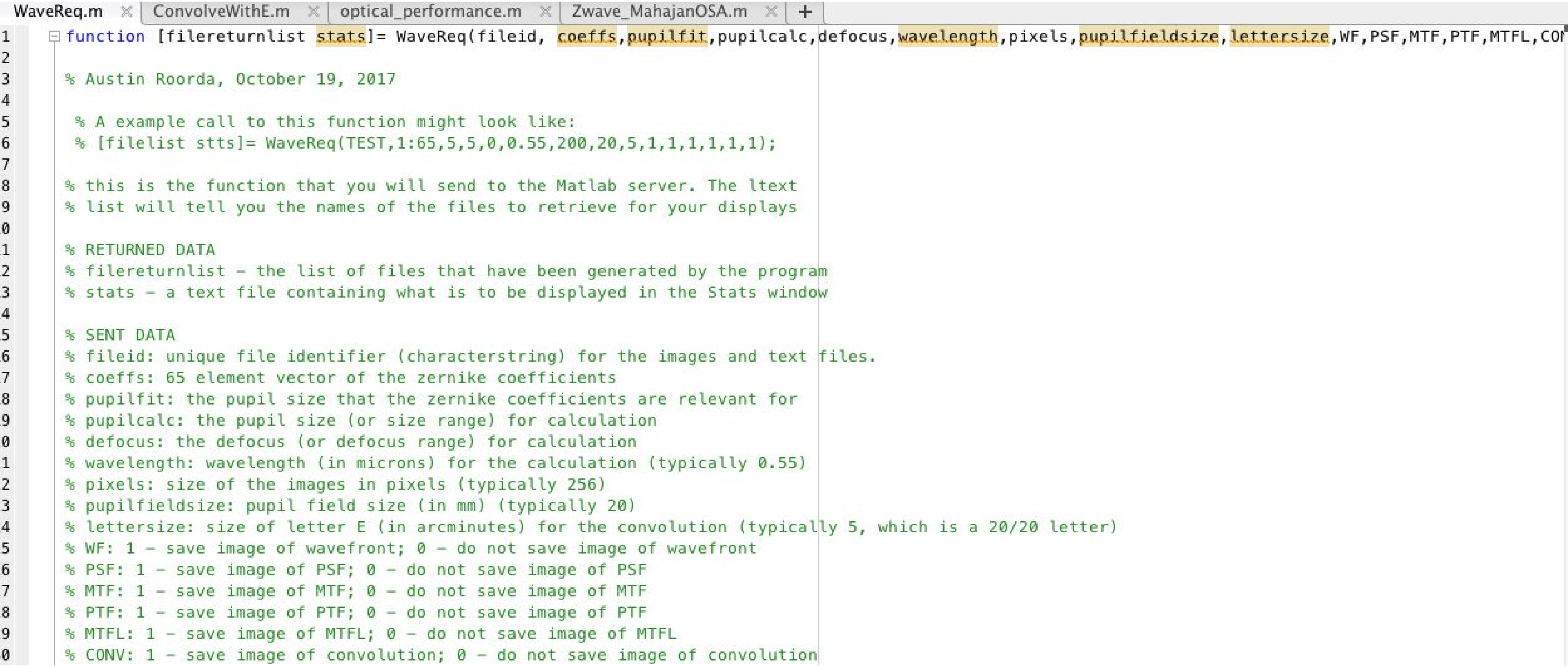
Our customer is UC Berkeley Optometry Professor Austin Roorda

What

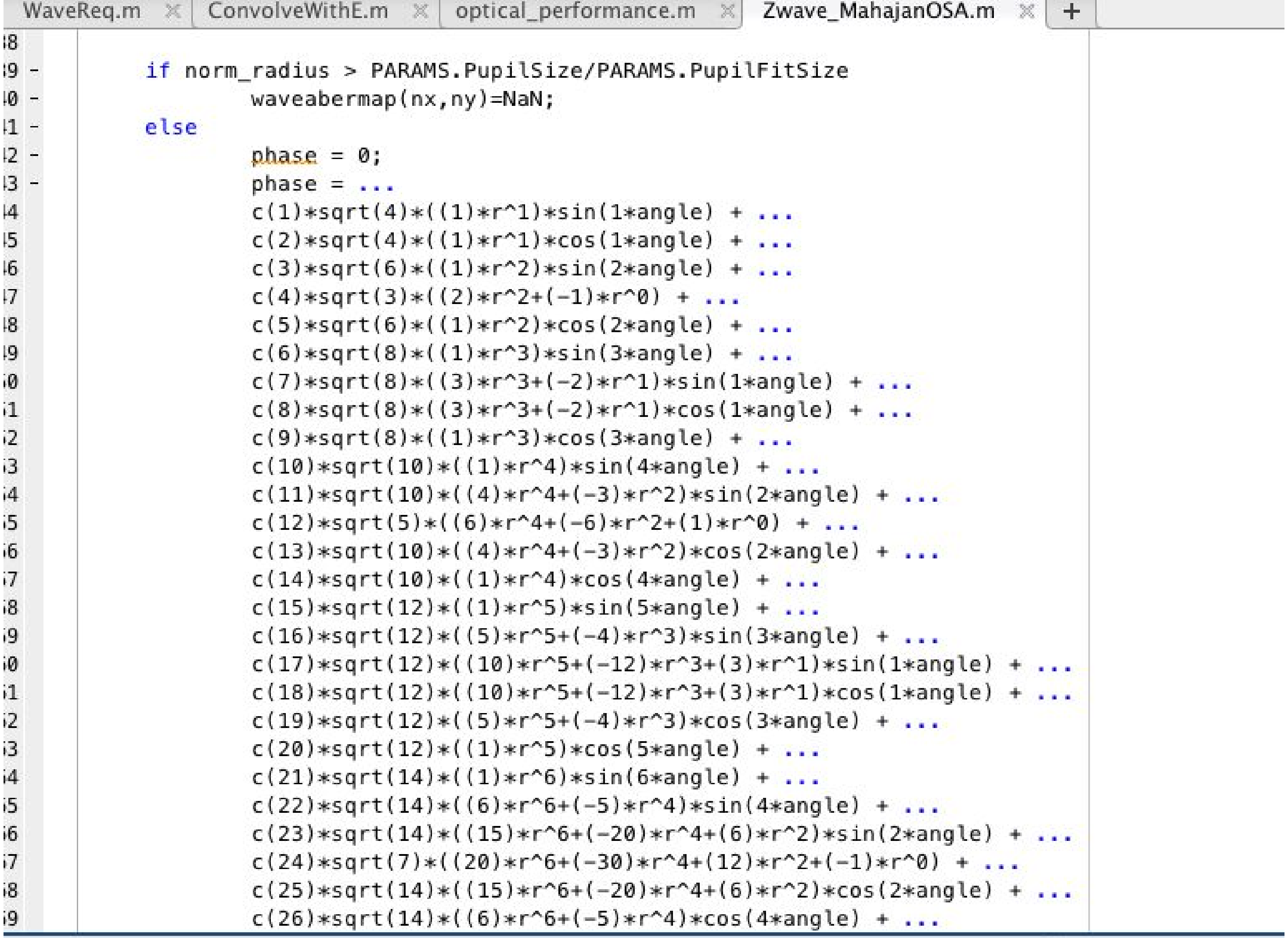
Build a user-friendly app for his students to use and learn about eye quality metrics without having to run MATLAB

Why

MATLAB is required coding environment, but his students as end-users lack MATLAB experience hence can’t run the code



```
function [filereturnlist stats]= WaveReq(fileid, coeffs,pupilfit,pupilcalc,defocus,wavelength,pixels,pupilfieldsize,lettersize,WF,PSF,MTF,PTF,MTFL,CO

% Austin Roorda, October 19, 2017

 % A example call to this function might look like:
 % [filelist stts]= WaveReq(TEST,1:65,5,5,0,0.55,200,20,5,1,1,1,1,1,1);

% this is the function that you will send to the Matlab server. The ltext
% list will tell you the names of the files to retrieve for your displays

% RETURNED DATA
% filereturnlist - the list of files that have been generated by the program
% stats - a text file containing what is to be displayed in the Stats window

% SENT DATA
% fileid: unique file identifier (characterstring) for the images and text files.
% coeffs: 65 element vector of the zernike coefficients
% pupilfit: the pupil size that the zernike coefficients are relevant for
% pupilcalc: the pupil size (or size range) for calculation
% defocus: the defocus (or defocus range) for calculation
% wavelength: wavelength (in microns) for the calculation (typically 0.55)
% pixels: size of the images in pixels (typically 256)
% pupilfieldsize: pupil field size (in mm) (typically 20)
% lettersize: size of letter E (in arcminutes) for the convolution (typically 5, which is a 20/20 letter)
% WF: 1 - save image of wavefront; 0 - do not save image of wavefront
% PSF: 1 - save image of PSF; 0 - do not save image of PSF
% MTF: 1 - save image of MTF; 0 - do not save image of MTF
% PTF: 1 - save image of PTF; 0 - do not save image of PTF
% MTFL: 1 - save image of MTFL; 0 - do not save image of MTFL
% CONV: 1 - save image of convolution; 0 - do not save image of convolution
```

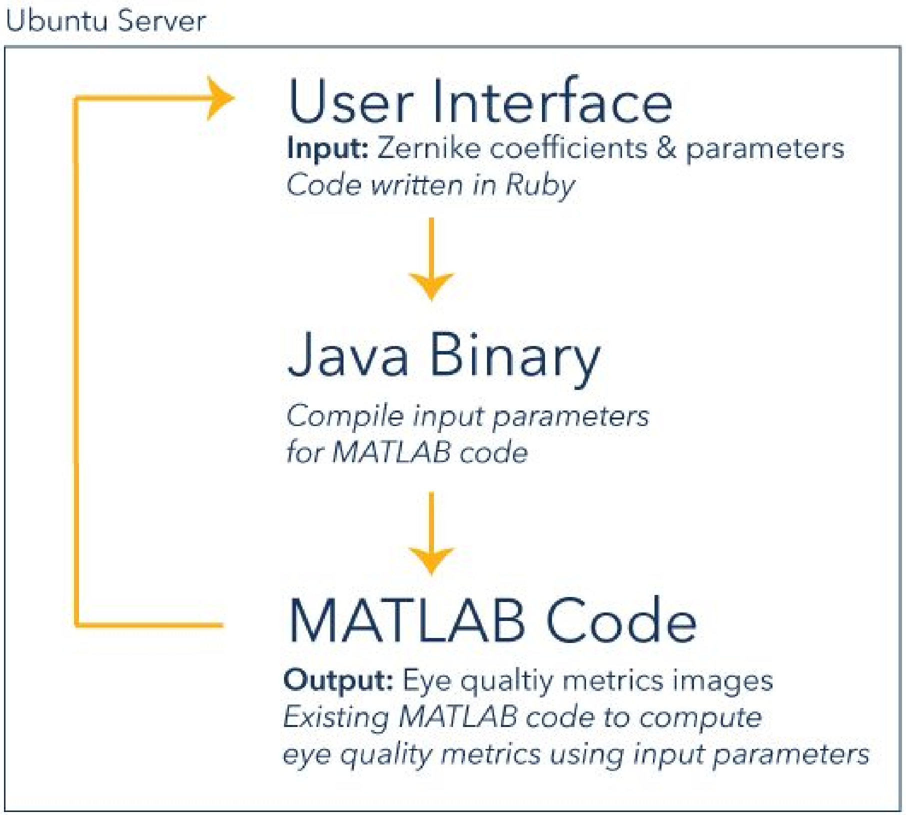


```
    if norm_radius > PARAMS.PupilSize/PARAMS.PupilFitSize
        waveabermap(nx,ny)=NaN;
    else
        phase = 0;
        phase = ...
        c(1)*sqrt(4)*((1)*r^1)*sin(1*angle) + ...
        c(2)*sqrt(4)*((1)*r^1)*cos(1*angle) + ...
        c(3)*sqrt(6)*((1)*r^2)*sin(2*angle) + ...
        c(4)*sqrt(3)*((2)*r^2+(-1)*r^0) + ...
        c(5)*sqrt(6)*((1)*r^2)*cos(2*angle) + ...
        c(6)*sqrt(8)*((1)*r^3)*sin(3*angle) + ...
        c(7)*sqrt(8)*((3)*r^3+(-2)*r^1)*sin(1*angle) + ...
        c(8)*sqrt(8)*((3)*r^3+(-2)*r^1)*cos(1*angle) + ...
        c(9)*sqrt(8)*((1)*r^3)*cos(3*angle) + ...
        c(10)*sqrt(10)*((1)*r^4)*sin(4*angle) + ...
        c(11)*sqrt(10)*((4)*r^4+(-3)*r^2)*sin(2*angle) + ...
        c(12)*sqrt(5)*((6)*r^4+(-6)*r^2+(1)*r^0) + ...
        c(13)*sqrt(10)*((4)*r^4+(-3)*r^2)*cos(2*angle) + ...
        c(14)*sqrt(10)*((1)*r^4)*cos(4*angle) + ...
        c(15)*sqrt(12)*((1)*r^5)*sin(5*angle) + ...
        c(16)*sqrt(12)*((5)*r^5+(-4)*r^3)*sin(3*angle) + ...
        c(17)*sqrt(12)*((10)*r^5+(-12)*r^3+(3)*r^1)*sin(1*angle) + ...
        c(18)*sqrt(12)*((10)*r^5+(-12)*r^3+(3)*r^1)*cos(1*angle) + ...
        c(19)*sqrt(12)*((5)*r^5+(-4)*r^3)*cos(3*angle) + ...
        c(20)*sqrt(12)*((1)*r^5)*cos(5*angle) + ...
        c(21)*sqrt(14)*((1)*r^6)*sin(6*angle) + ...
        c(22)*sqrt(14)*((6)*r^6+(-5)*r^4)*sin(4*angle) + ...
        c(23)*sqrt(14)*((15)*r^6+(-20)*r^4+(6)*r^2)*sin(2*angle) + ...
        c(24)*sqrt(7)*((20)*r^6+(-30)*r^4+(12)*r^2+(-1)*r^0) + ...
        c(25)*sqrt(14)*((15)*r^6+(-20)*r^4+(6)*r^2)*cos(2*angle) + ...
        c(26)*sqrt(14)*((6)*r^6+(-5)*r^4)*cos(4*angle) + ...
```

Requirements

- MATLAB is required for eye quality metrics computations
- A simple UI which hides the MATLAB layer and easily allows optometry students to input data for computation
- Display computed eye quality metrics images
- Save images for easy download

Technical Design Overview



Ubuntu Server

User Interface

Input: Zernike coefficients & parameters

Code written in Ruby

Java Binary

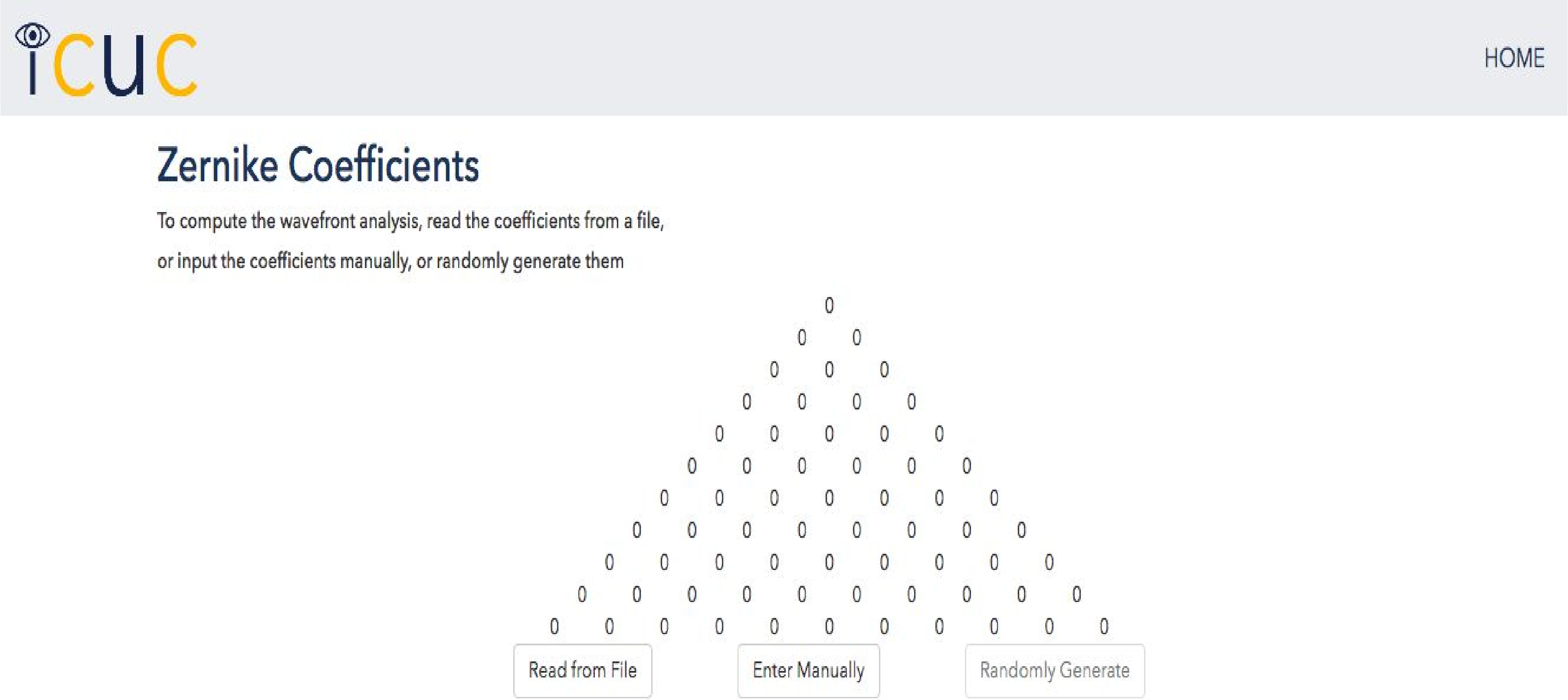
Compile input parameters
for MATLAB code

MATLAB Code

Output: Eye qualtiy metrics images

Existing MATLAB code to compute
eye quality metrics using input parameters

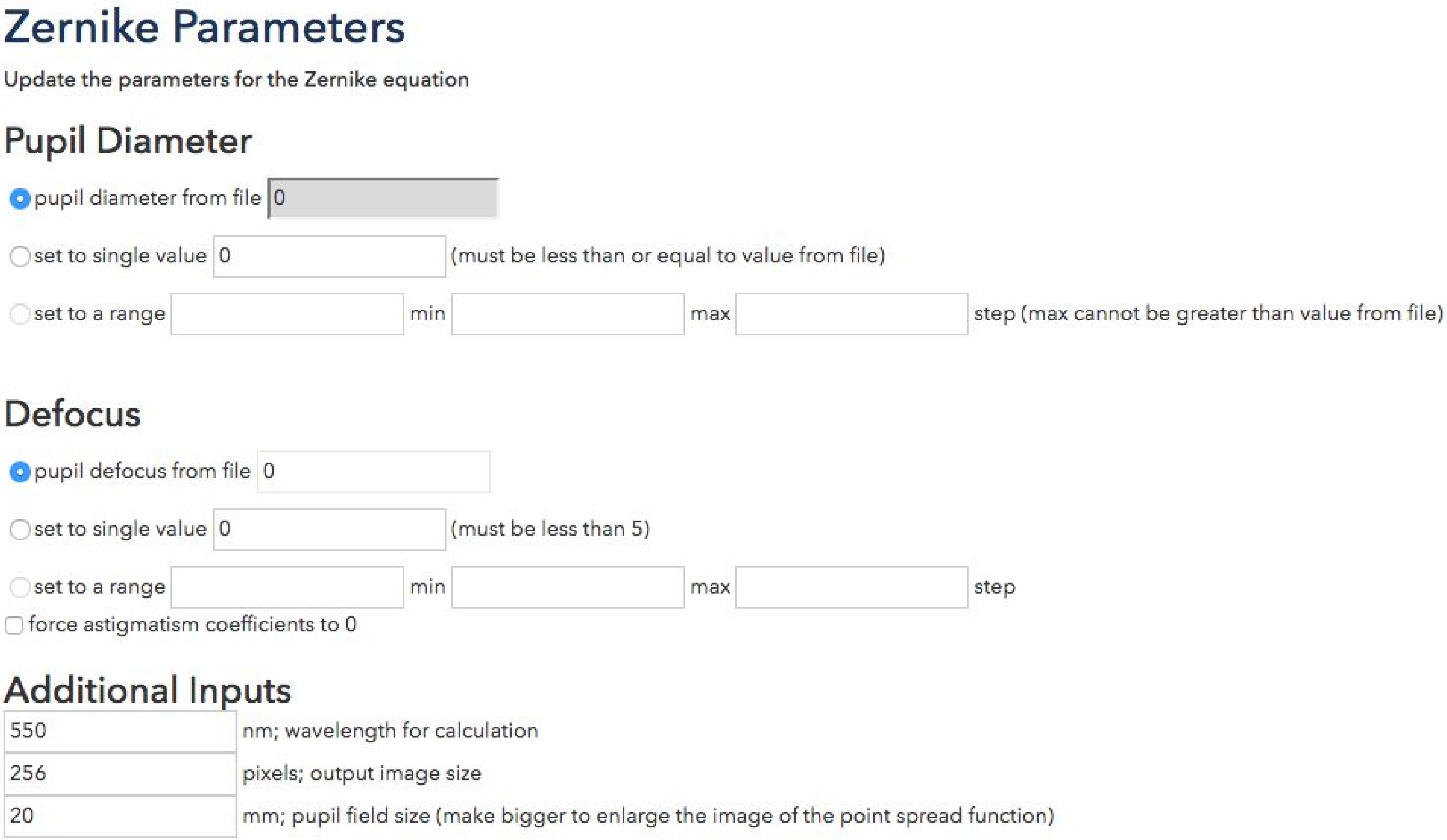
Design Overview cont.



Zernike Coefficients

To compute the wavefront analysis, read the coefficients from a file, or input the coefficients manually, or randomly generate them

```
                    0
                  0   0
                0   0   0
              0   0   0   0
            0   0   0   0   0
          0   0   0   0   0   0
        0   0   0   0   0   0   0
      0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0
```



Zernike Parameters

Update the parameters for the Zernike equation

Pupil Diameter

● pupil diameter from file 0

○ set to single value 0 (must be less than or equal to value from file)

○ set to a range ___ min ___ max ___ step (max cannot be greater than value from file)

Defocus

● pupil defocus from file 0

○ set to single value 0 (must be less than 5)

○ set to a range ___ min ___ max ___ step

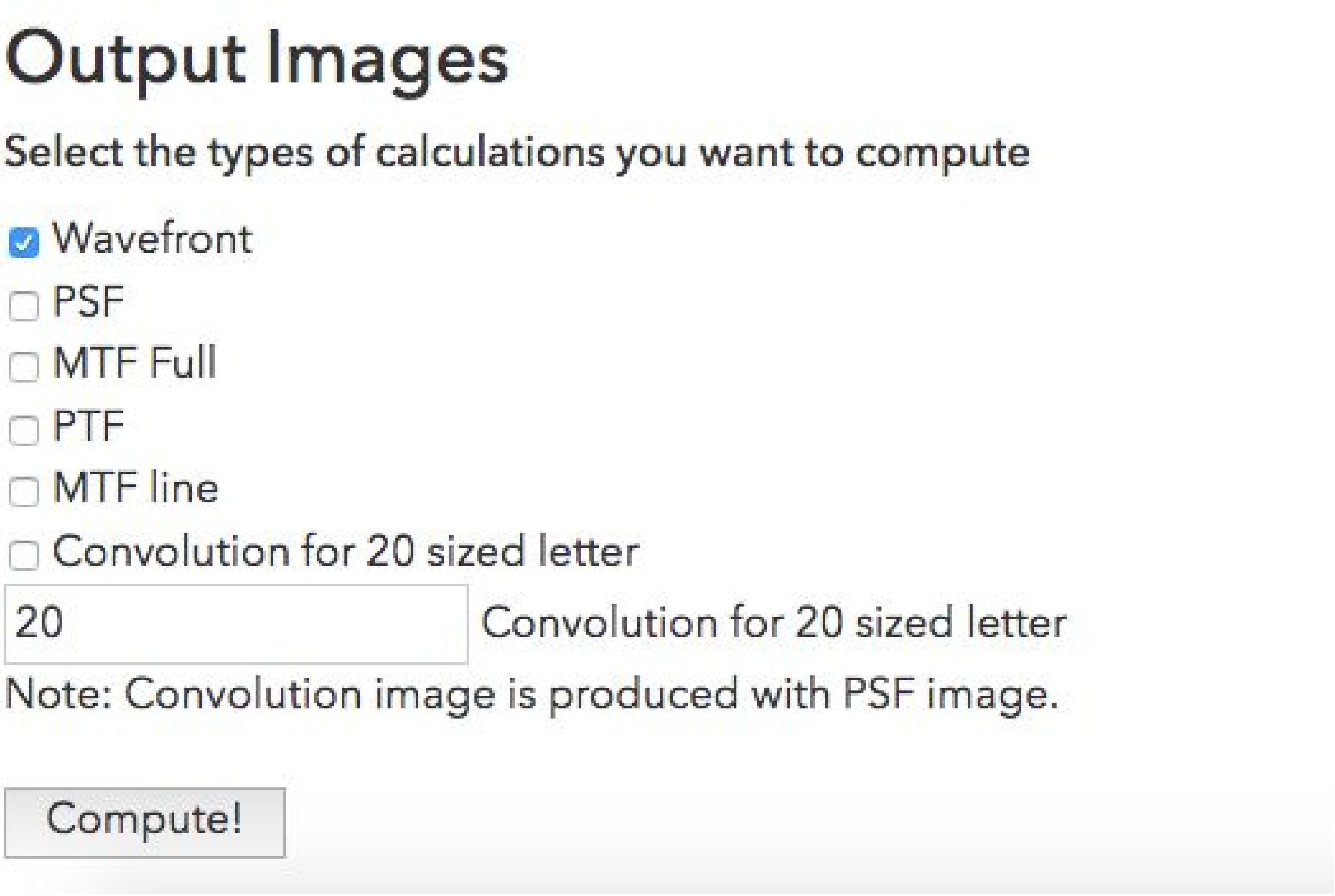
□ force astigmatism coefficients to 0

Additional Inputs

550 nm; wavelength for calculation

256 pixels; output image size

20 mm; pupil field size (make bigger to enlarge the image of the point spread function)



Output Images

Select the types of calculations you want to compute

☑ Wavefront

□ PSF

□ MTF Full

□ PTF

□ MTF line

□ Convolution for 20 sized letter

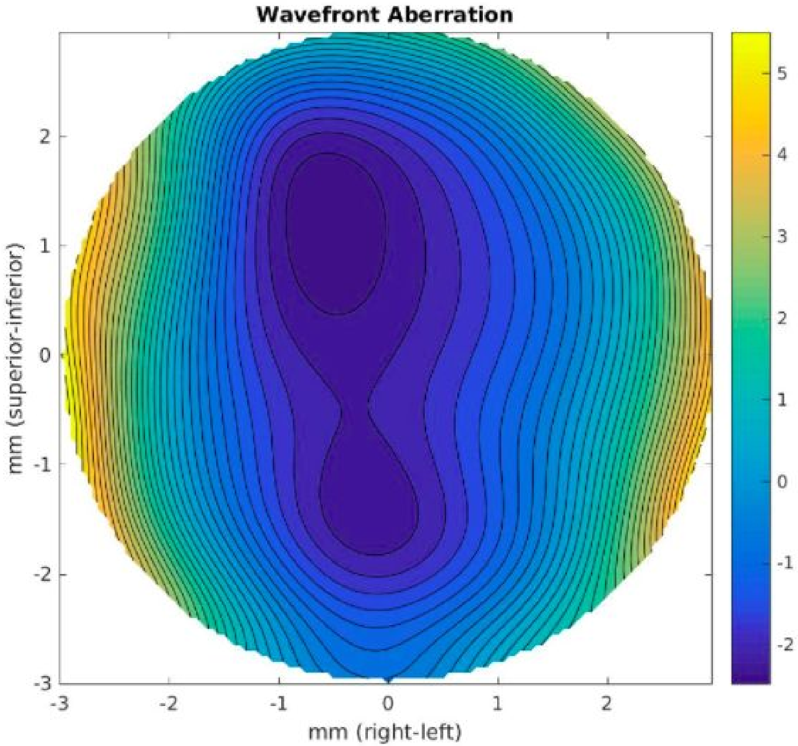
20 Convolution for 20 sized letter

Note: Convolution image is produced with PSF image.

Output Images of the Zernike Equation

1/3

‹



›

Download computedOCQQTEHR-6.141-astig-axis-60-WF.jpg

Technical Challenges and Solutions

Challenges

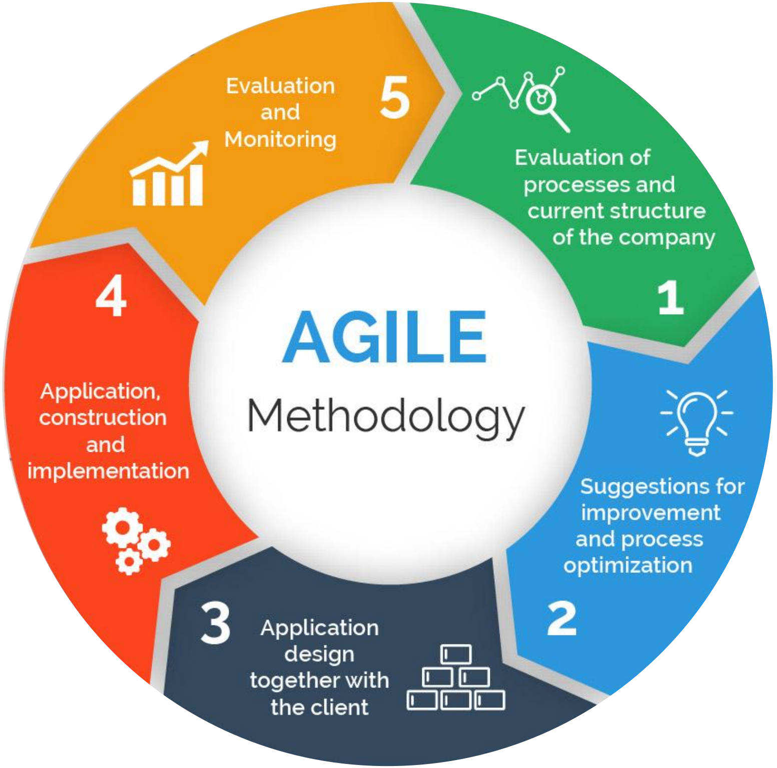
- MATLAB can’t run on Heroku
- Customer wants to run on Windows, but Ruby is incompatible
- Set up new app on new server (Apache, Passenger, permissions)
- MATLAB computation and Java compilation time are slow

Solutions

- Ran on optometry lab’s server
- Customer set-up Ubuntu, which is compatible w/ Ruby
- Ben Mehne (GSI)) helped with server set-up
- Added loading page & limited number of user computations

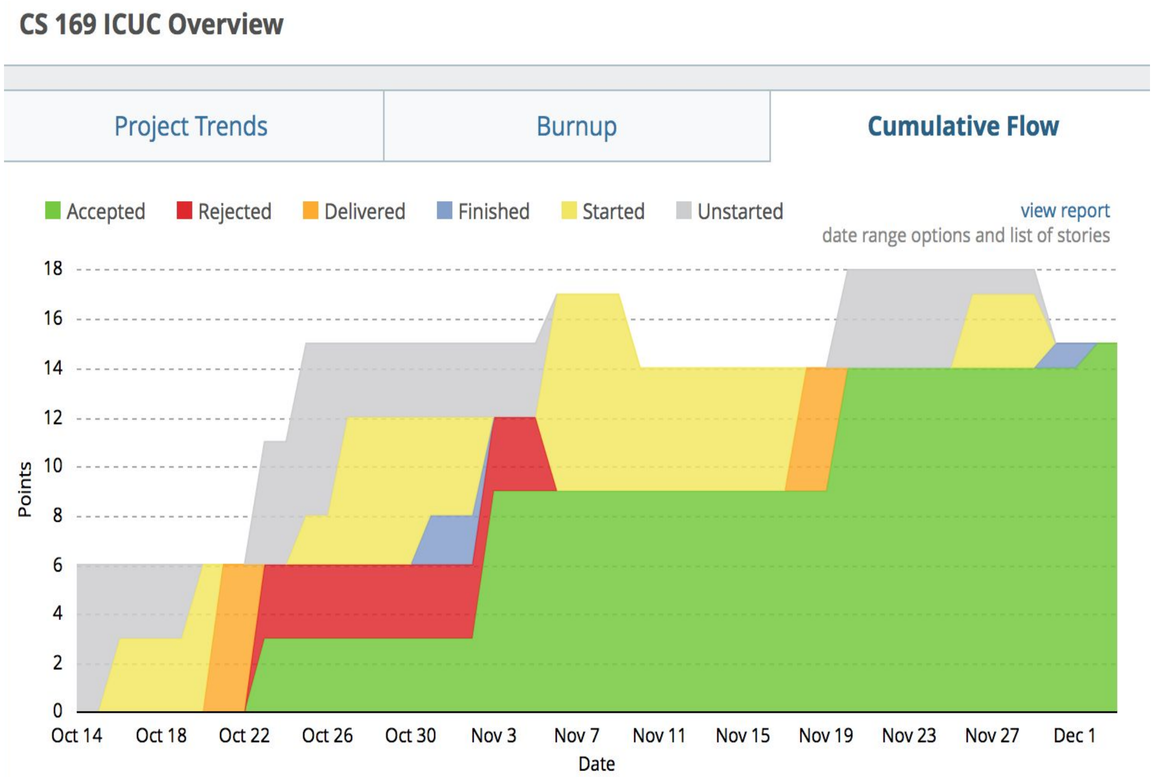
Agile Practices Checklist

- ✓ Simple design
- ✓ Continuous integration
- ✓ Code refactoring
- ✓ Test driven development
- ✓ Efficient division of labor
- ✓ Used pair programming
- ✓ Effective teamwork & team communication


AGILE
Methodology
1
Evaluation of processes and current structure of the company
2
Suggestions for improvement and process optimization
3
Application design together with the client
4
Application, construction and implementation
5
Evaluation and Monitoring

Tools Usage Checklist

- ✓ Git for version control
  - ○ **314** commits
  - ○ **12** branches
- ✓ Code Climate
  - ○ Maintainability: **A**
  - ○ Test coverage: **91%**
- ✓ Travis CI
  - ○ Status: **Passing**
- ✓ Pivotal Tracker
  - ○ **15/18 points** accepted



CS 169 ICUC Overview

Customer Interaction

Communication

- Built positive relationship with customer from beginning
- Established designated point of contact with customer
- Regularly informed customer on agenda and progress in a timely manner and

Engagements

- Customer provided specific high level requirements
- Customer provided concise feedback at each meeting
- Team prepared for and actively participated in meetings
- Customer provided a technical point of contact

Live Demo

Thank You!

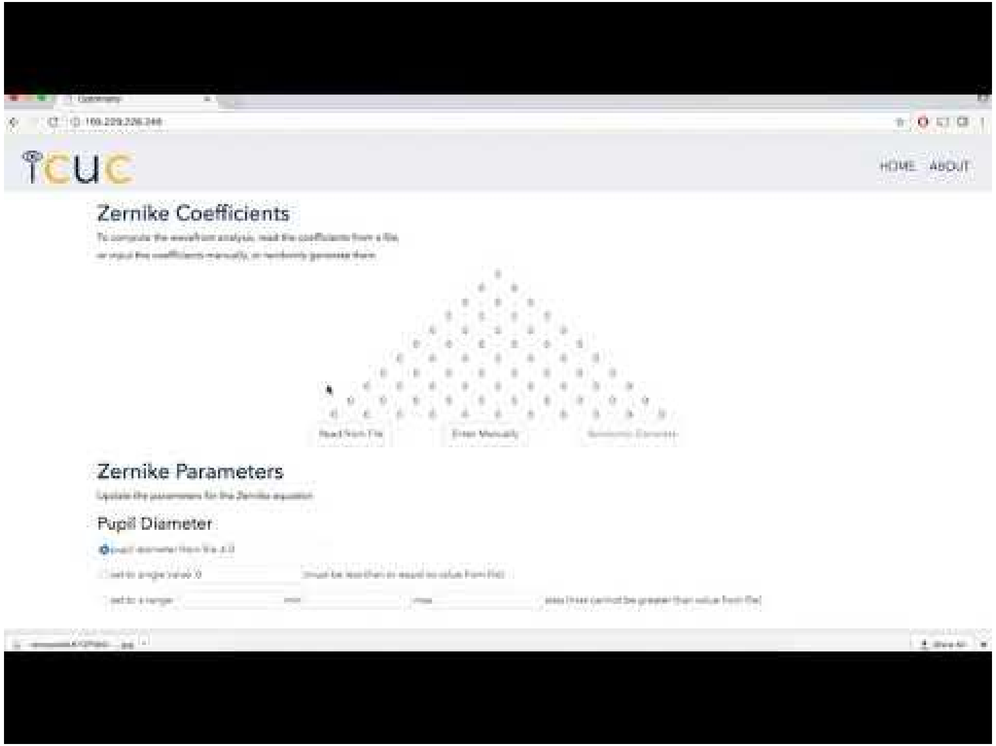
Professor Fox

GSIs: Steven, An, & Ben

Professor Roorda, and Pavan

Github repo: vicpark/icuc

Recorded Demo



Zernike Coefficients

To compute the wavefront analysis, read the coefficients from a file, or input the coefficients manually, or randomly generate them.

Zernike Parameters

Update the parameters for the Zernike equation

Pupil Diameter

● pupil diameter from file 4.0

○ set to single value 0 (must be less than or equal to value from file)

○ set to a range: min max (max cannot be greater than value from file)