Gliwice, 28.10.2017

Software Engineering

Analysis and Requirements

Section 3
Tomasz Mańka
Dominik Baran
Dawid Czyrny
Rafał Kubas
Wojciech Prażuch
Karol Folkert
Tomasz Tarasiewicz
Kamil Grzechnik
Karol Kostorz

General product description

Product that we have to release is multiple-image viewer. Solution should be implemented in C# language. Application should be able to display several images of various formats at once. Team should implement basic operation that helps reviewing images(i.e. zooming, translating). User should have opportunity to synchronize these operations. Program should allow user to select certain regions of the image, save them and investigate basic values of them like average, deviation.

Our vision of understanding requirements

Below presented all requirements of application and our way of understanding it.

1. A variety of file formats must be supported
    a. Functional
    - We will support basic graphic formats, namely: jpg, bmp, png, gif, raster images
    - Max supported file size will be 5MB

    b. Non functional
    - Loading files should be quite fast and fluent. Files should be loaded up to 1 seconds.

2. The number of windows(and their arrangement) to be flexible.
    a. Functional
        i. User should be able to choose windows arrangement at start of the application
        ii. Possible arrangements: 1 window, 2 windows next to each other, 2x2 windows, 3x3 windows
    b. Non functional
        i. At starts of application proper modal asks user about windows arrangement

3. We may attach a file or a file list to each window
    a. Functional
        i. Each window has proper icon which starts dialog where we can choose list of files that will be attached to it

   b. Non functional
      i. User should easily switch between files in window(with mouse and keyboard keys)

4. Zooming and translations should be synchronized(with and option to desynchronize if necessary and resynchronize)
   a. Functional
      i. There should be icon that allows to switch between synchronized and desynchronized mode
      ii. Icon should indicates actual state(open, closed lock)
   b. Non functional
      i. User should be able to easily find icon
      ii. Icon should has tooltip which informs about its functionality

5. Possibility to measure the pixel values (synchronized position across particular views).
   a. Functional
      i. User should be able to read pixel values of pixel selected with cursor
      ii. When user selects certain pixel in one window, values of corresponding pixels in other windows should be displayed
   b. Non functional
      i. Pixel values should be displayed in small window next to the pixel

6. Possibility to annotate a region of interest and measure the statistics inside (average, min, max, deviation, etc.). This is also to be synchronized across the views.
   a. Functional
      i. There should be information about statistics of each window
      ii. If user select any region of the image statistics should be connected to this region
      iii. Statistics should contain following information: min deviation, max deviation, average values
   b. Non functional
      i. Statistics should be presented in clear, intuitive and eye catching way
      ii. It should be clear which statistic belongs to which window/region

7. Option to serialize the outcome (zoomed image regions, properties) to files.
    a. Functional
        i. User should be able to export selected region of the image
        ii. Exported region should be saved in bpm lossless format
    b. Non functional
        i. Next to selected region there should be intuitive icon for exporting

8. Option to save the position and keep a list of recent positions to make it possible to browse quickly between them.
    a. Functional
        i. User should be able to save current region with unique name
        ii. User should be able to choose name of saving region
        iii. User should be able to browse saved regions
        iv. User should be able to load one of saved regions
    b. Non functional
        i. Saving should work like bookmarks, there should be list of saved regions labeled by names
        ii. After clicking on one region it should appear on all windows or on active one(depends on selected mode - described in point 4ai)

Flow in issue tracker

Our SCRUM table is divided for following sections: Reopened , Blocked, To do, In development, Code Review, Ready for QA, In Testing, Task done. Flow of the single task in our project is presented below.

First task after creation goes to section **To Do** if developer can immediately start work on it or to **Blocked** if it waits for other task(it should be indicated for which task it waits). When developer starts to work on task he should drag it to **In development** section. When task is ready and Pull Request is created in repository developer should move task to **Code Review**. From there head of team should move it to **Ready for QA** if everything is correct, if not he should force developer to make improvements. Next head of QA should assign task to certain employees and task should wait there until one of testers starts work. When QA starts testing particular task he should move it to **In Testing** section. If everything works as it should task goes to **Task done** section. If task didn't pass tests it should go to **Reopened** section when it waits for developer to work on it again.

Moving task to proper section and making sure that it has correct state in each time is crucial and it’s the only wait to have order in project, know what progress of implementing solution do we have, how much time left and what is next step of our work.