HaSuRiski: interactive app

Anton Akusok, Arcada UAS

anton.akusok@arcada.fi

Detecting acid sulfate soils

Knowledge gap:

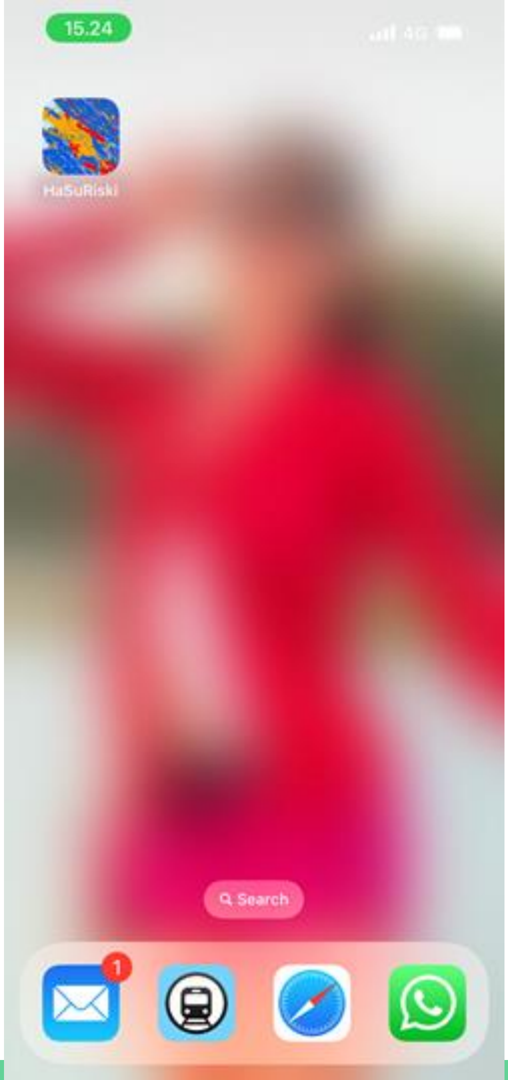
- use thousands of measurements by GTK
- add new point effortlessly
- usable by anyone

Scientific gap:

- make a live edge computing
- avoid server for rare but heavy computational tasks
- high quality visualizations
- investigate phone capabilities

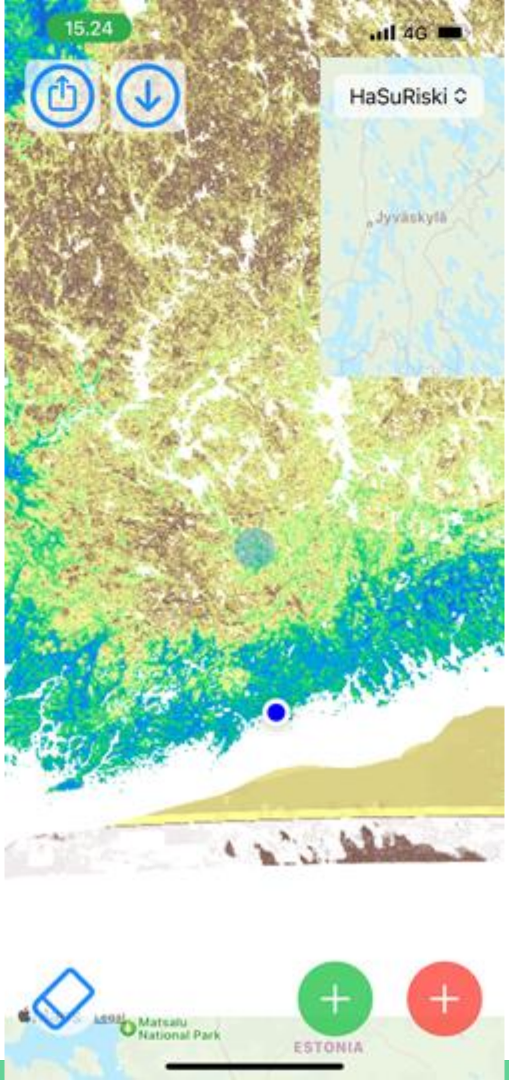
Create an app that

- works



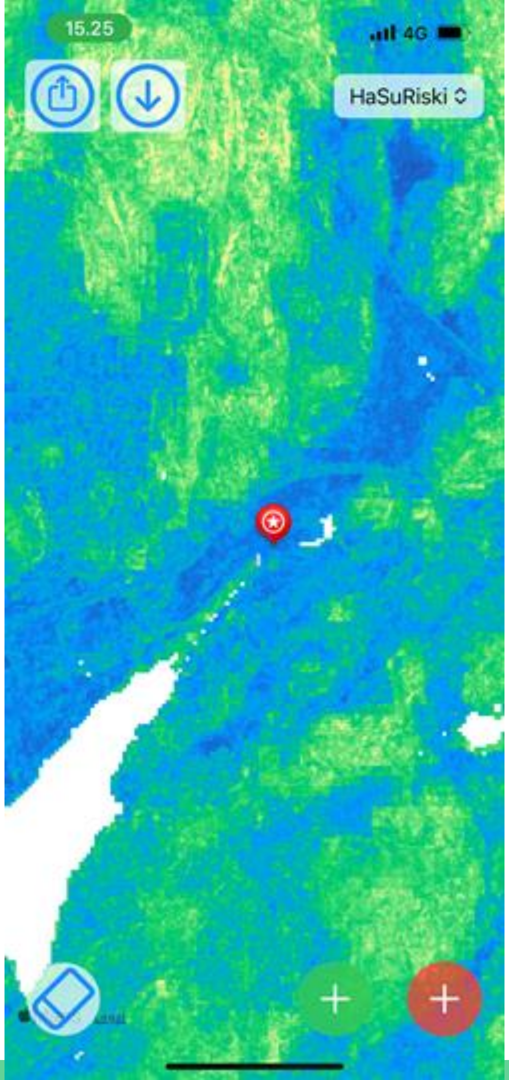
Create an app that

- works
- shows predicted maps



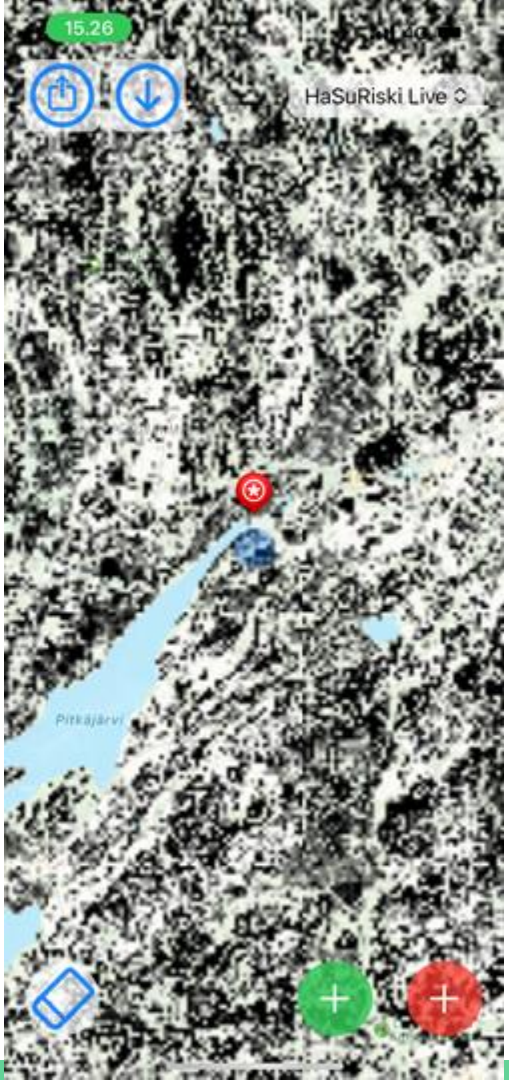
Create an app that

- works
- shows predicted maps
- add measurements, save/load



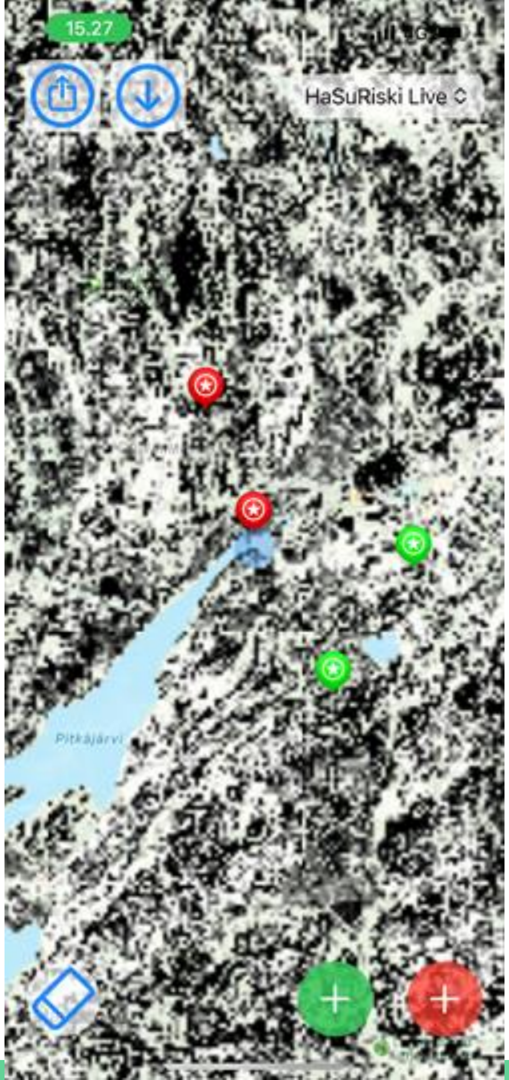
Create an app that

- works
- shows predicted maps
- add measurements, save/load
- real-time predicted map



Create an app that

- works
- shows predicted maps
- add measurements, save/load
- real-time predicted map
- update predicted map with more points



Create an app that

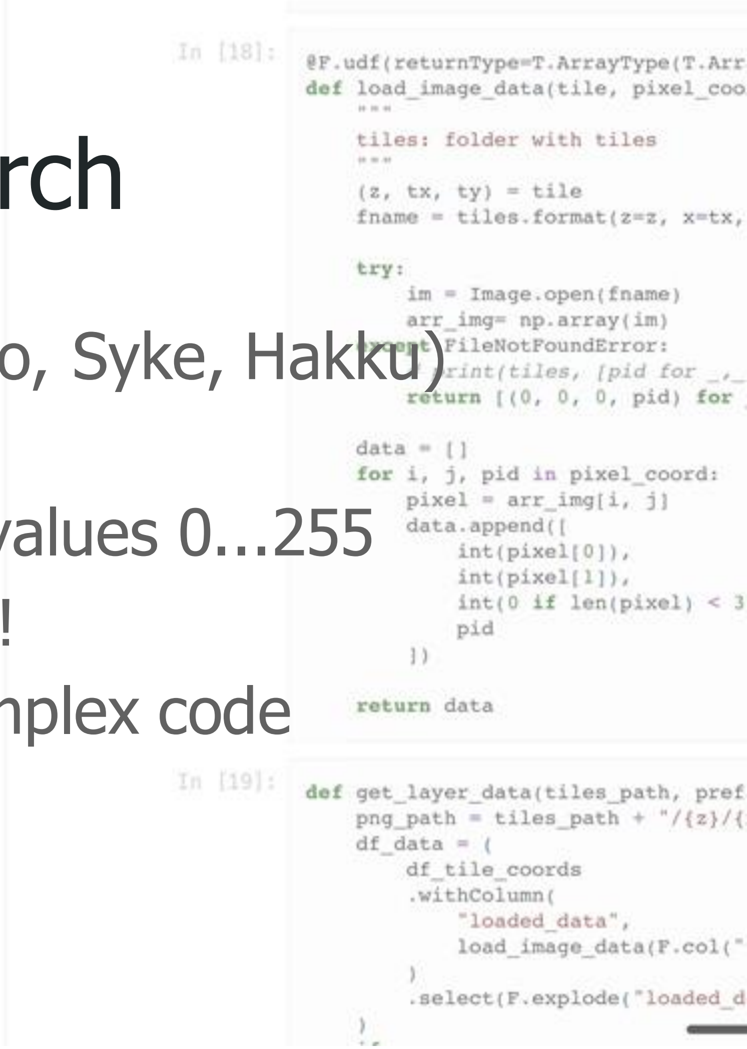
- works
- shows predicted maps
- add measurements, save/load
- real-time predicted map
- update predicted map with more points
- **looks gorgeous!**

Application

Development process

Development: Map Research

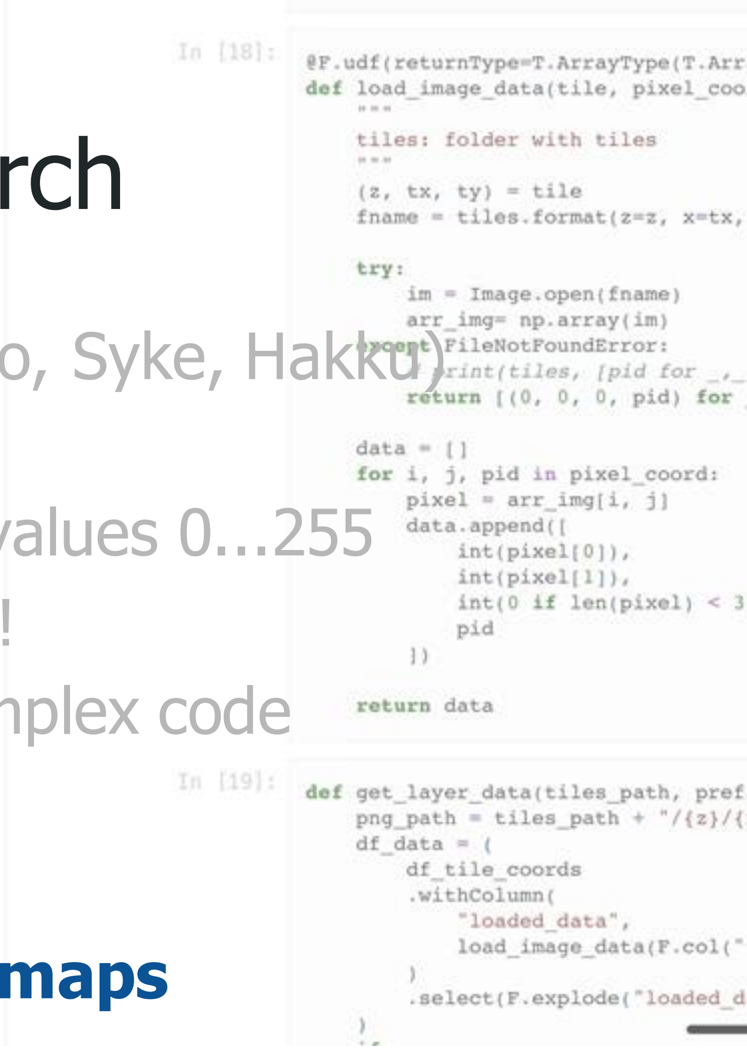
- map layers include open data (Lapio, Syke, Hakku)
- labeled points provided by GTK
- preprocessing map data into pixel values 0...255
- open-source programming libraries!
- but lots of specific libraries and complex code



```
In [18]: @F.udf(returnType=T.ArrayType(T.Arr
         def load_image_data(tile, pixel_coo
             """
             tiles: folder with tiles
             """
             (z, tx, ty) = tile
             fname = tiles.format(z=z, x=tx,

             try:
                 im = Image.open(fname)
                 arr_img= np.array(im)
             except FileNotFoundError:
                 # print(tiles, [pid for _,_
                 return [(0, 0, 0, pid) for

             data = []
             for i, j, pid in pixel_coord:
                 pixel = arr_img[i, j]
                 data.append([
                     int(pixel[0]),
                     int(pixel[1]),
                     int(0 if len(pixel) < 3
                     pid
                 ])

             return data

In [19]: def get_layer_data(tiles_path, pref
             png_path = tiles_path + "/{z}/{
             df_data = (
                 df_tile_coords
                 .withColumn(
                     "loaded_data",
                     load_image_data(F.col("
                 )
                 .select(F.explode("loaded_d
             )
```

Development: Map Research

- map layers include open data (Lapio, Syke, Hakku)
- labeled points provided by GTK
- preprocessing map data into pixel values 0...255
- open-source programming libraries!
- but lots of specific libraries and complex code
- prepare them as input data files
- clean GitHub repo:
  github.com/akusok/hasuriski_maps



```
In [18]:
@F.udf(returnType=T.ArrayType(T.Arr
def load_image_data(tile, pixel_coo
    """
    tiles: folder with tiles
    """
    (z, tx, ty) = tile
    fname = tiles.format(z=z, x=tx,

    try:
        im = Image.open(fname)
        arr_img= np.array(im)
    except FileNotFoundError:
        # print(tiles, [pid for _,_
        return [(0, 0, 0, pid) for

    data = []
    for i, j, pid in pixel_coord:
        pixel = arr_img[i, j]
        data.append([
            int(pixel[0]),
            int(pixel[1]),
            int(0 if len(pixel) < 3
            pid
        ])

    return data
```

```
In [19]:
def get_layer_data(tiles_path, pref
    png_path = tiles_path + "/{z}/{
    df_data = (
        df_tile_coords
        .withColumn(
            "loaded_data",
            load_image_data(F.col("
        )
        .select(F.explode("loaded_d
    )
```

Development: App Research

- standard map with custom tiles
- several tile maps (GTK and project predictions)
- Live predict tiles computed at runtime per-pixel
- model updates with adding/removing data points
- cache for fast browsing of the same area
- code + setup Readme:
  github.com/akusok/HaSuRiski
- map data files: Arcada’s Google Drive (by request)

Improvements after Oulu demo

- local files: no Internet!
- user location!
- app icon
- map resolutions
- works on iPad

Next steps

Plans for app development in upcoming projects

- better colors
- load maps without special data
- faster model for full resolution
- NN image features

Thanks for your attention!

Questions?