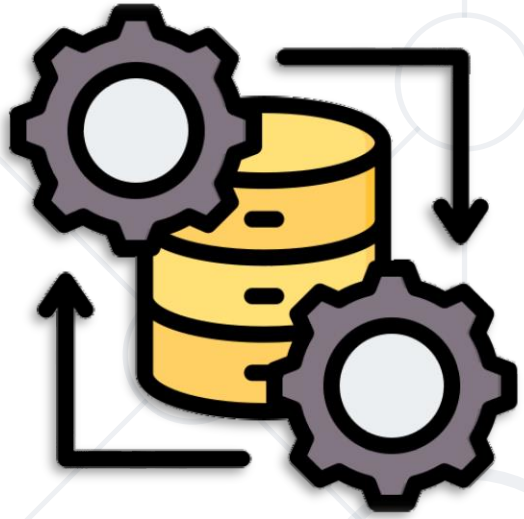
# Data Science in Production

## Data Pipelines and Processes




**Yordan Darakchiev**

**Technical Trainer**

Software University

SoftUni

Software University

https://softuni.bg

# Have a Question?

Software University

sli.do

#DataScience

# Table of Contents

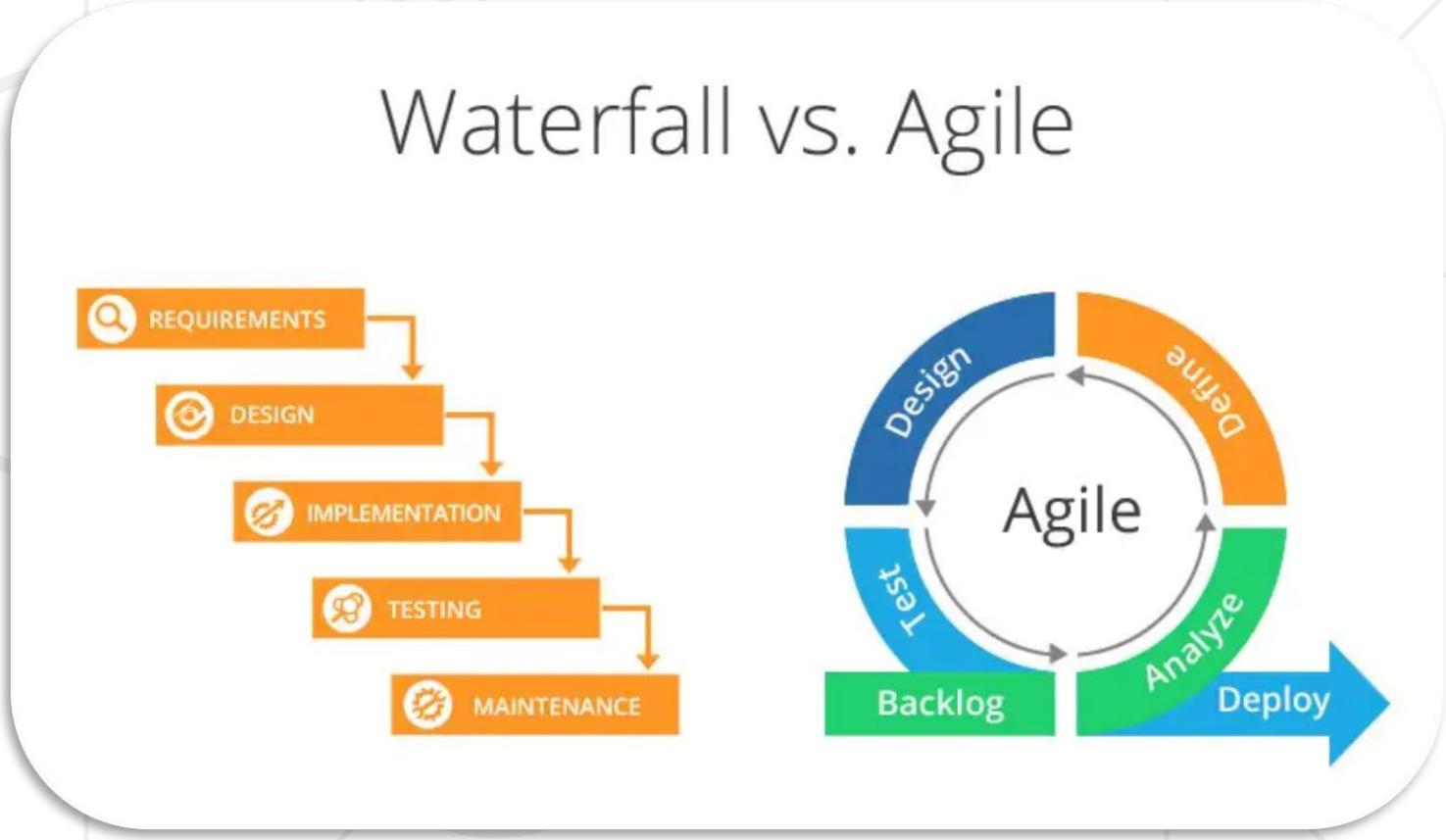


# Work Processes

## Collaborating with Others

# Agile (vs. Waterfall)

- Incremental
- **Iterative**
- Focused on what our client needs
- *The Agile Manifesto*



# Scrum

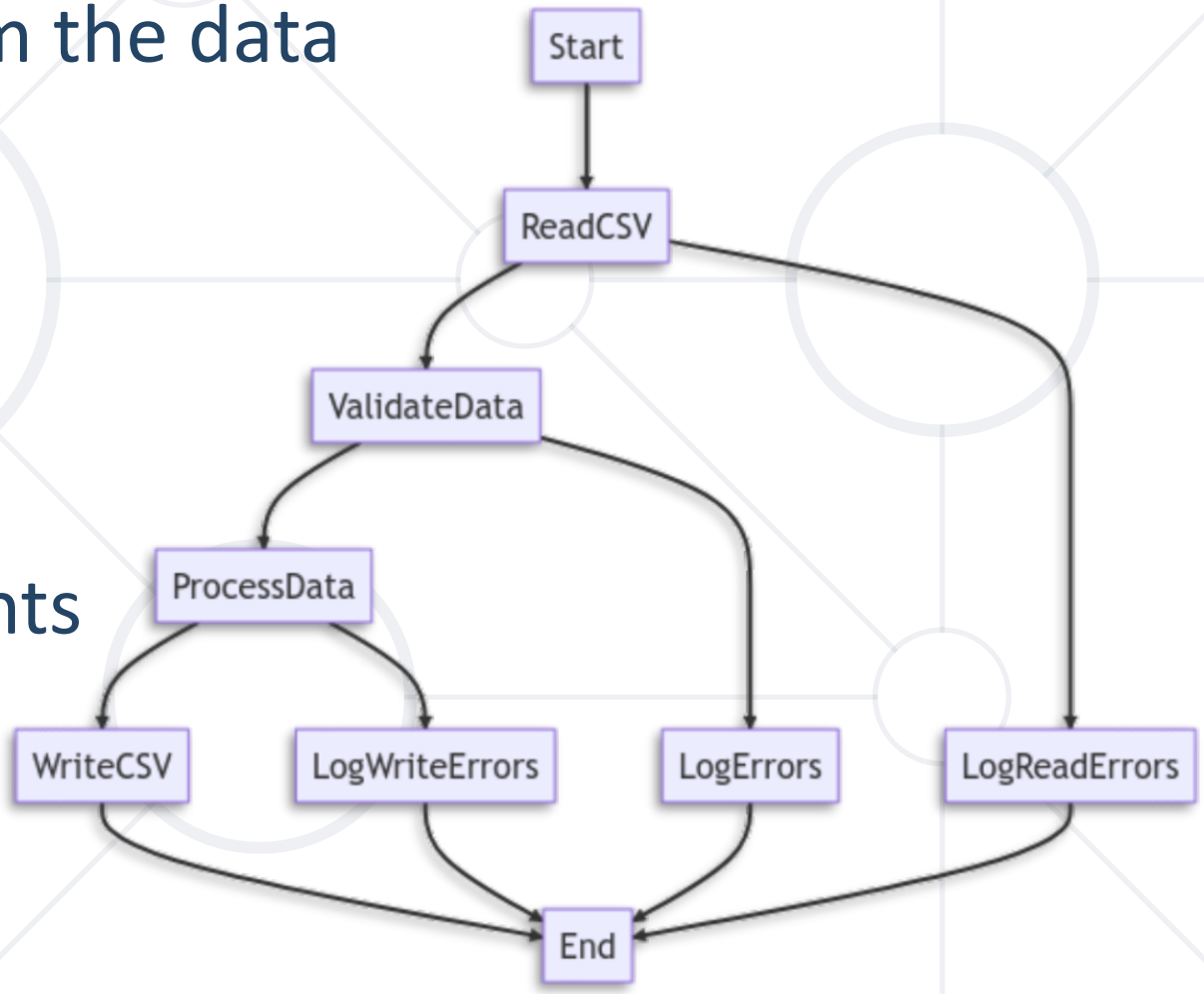
- Lightweight
- Experimental
  - Figure out as you go
- **Small pieces of work at a time**
- Feedback loop
- It's not hard to *get started*
- Scrum Master ↔ Product Owner ↔ Scrum team
- Product backlog ↔ sprint

# Data Pipelines

## From Notebooks to Automation

# Data Pipelines

- DAG of functions which transform the data
  - Automated
  - Reusable
  - Testable / tested
- Can be arbitrarily complex
- Can be used to quickly gain insights
- Can run anywhere
  - Cloud pipelines



# Luigi

- Installation

```
conda install anaconda::luigi
```

- Tasks and targets
- Getting started

```
class HelloLuigi(luigi.Task):

    def output(self):
        return luigi.LocalTarget('hello-
luigi.txt')

    def run(self):
        with self.output().open("w") as outfile:
            outfile.write("Hello Luigi!")
```

```
python -m luigi --module <file name> HelloLuigi
```

# Luigi

- Parameters

```
class ParamDemo(luigi.Task):
    my_param = luigi.IntParameter(
        42,
        visiibility=luigi.parameter.ParameterVisibility.PRIVATE)

    def output(self):
        print(self.my_param) # Usage: directly, with self
```

- Configuration
  - luigi.cfg
  - Parameter values can be overwritten (code config overwrites file config)

# Big(ger) Data

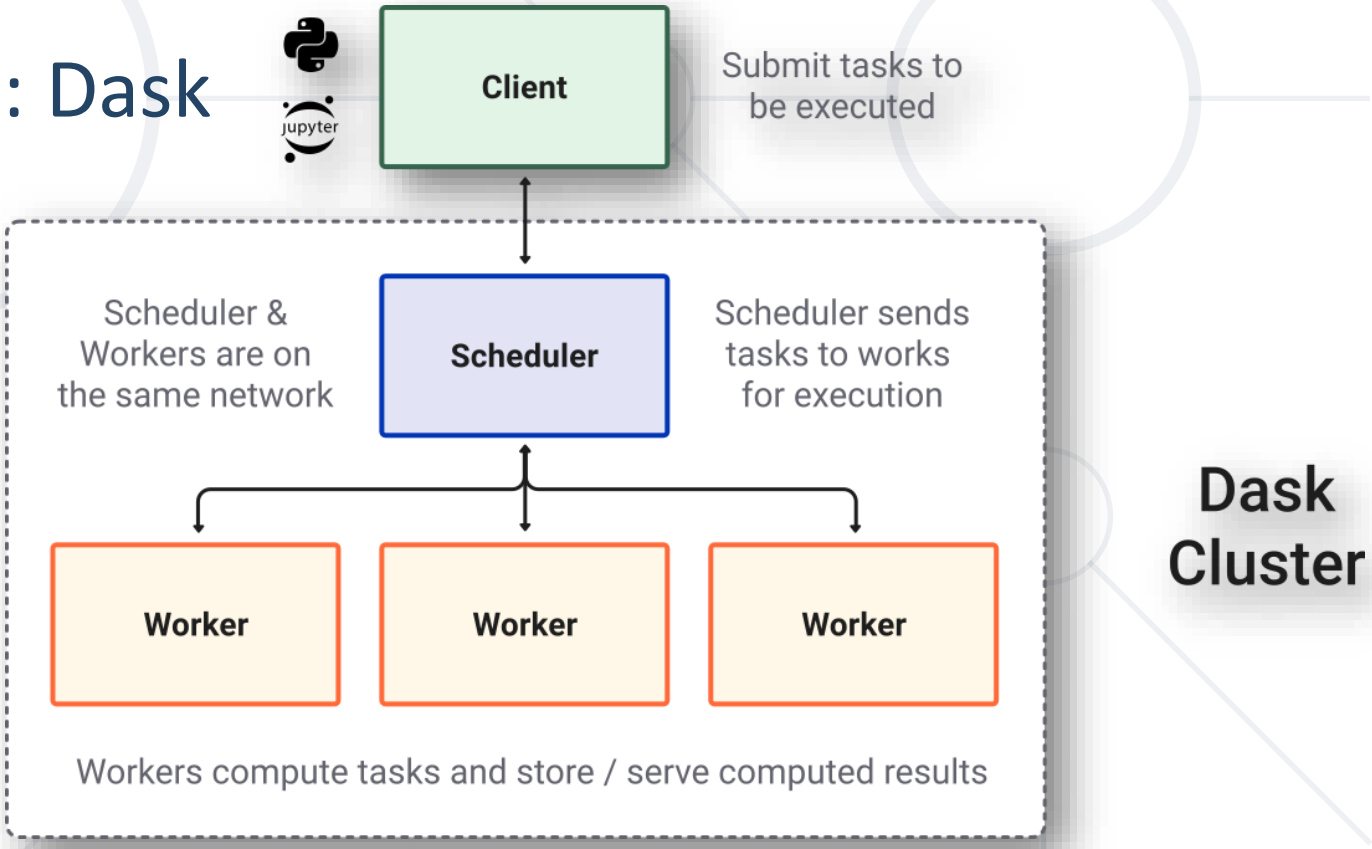
## Scaling up our Scripts and Computation

# Getting to Production

- Data versioning: DVC
- Experiment tracking: MLFlow
- Concurrent data manipulation: Dask

```
import dask

df = dask.datasets.timeseries()
print(df[df.y > 0])
```

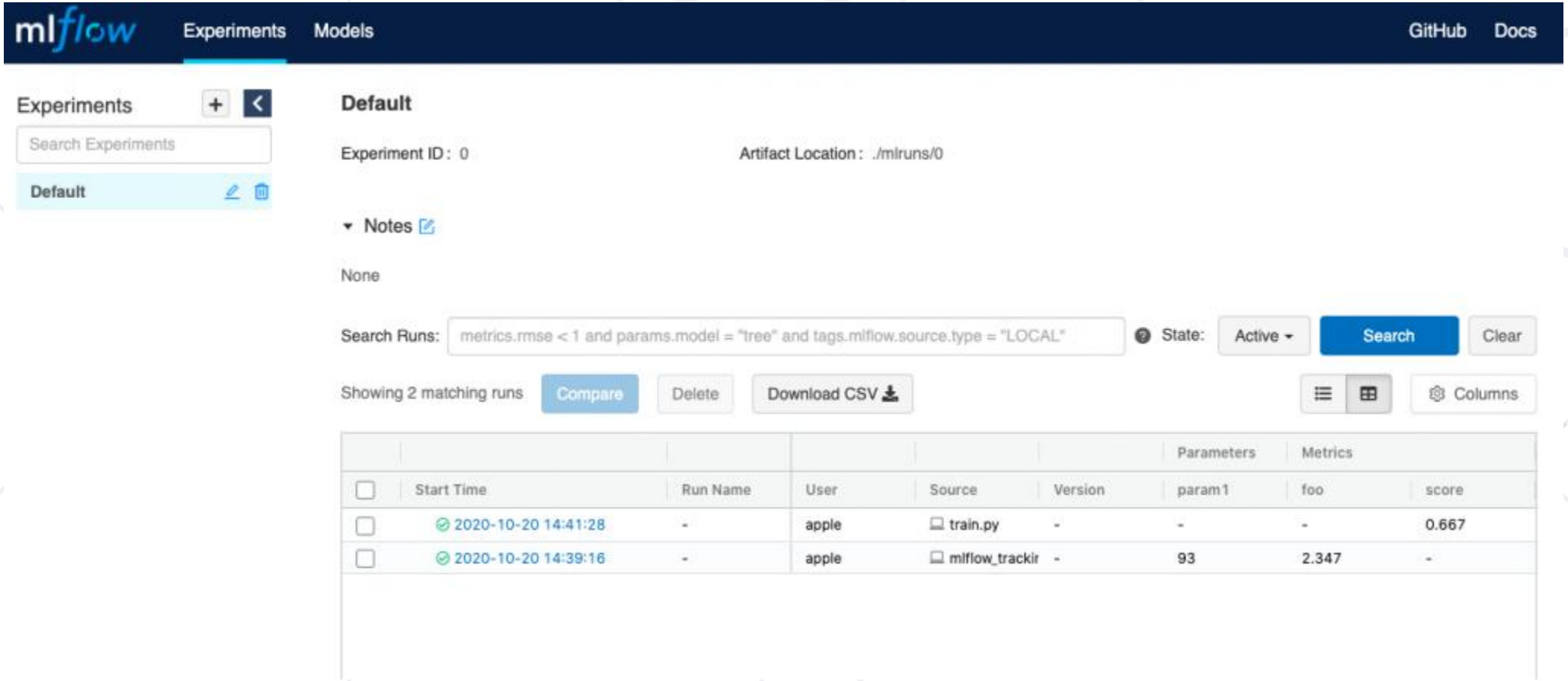


# Monitoring

## Getting Insights

# Monitoring

- MLFlow tracking can do model / data / artifact monitoring



# Dashboards

- Many libraries can do that
- Plotly's dash is really popular
  - *Examples*, *tutorial*

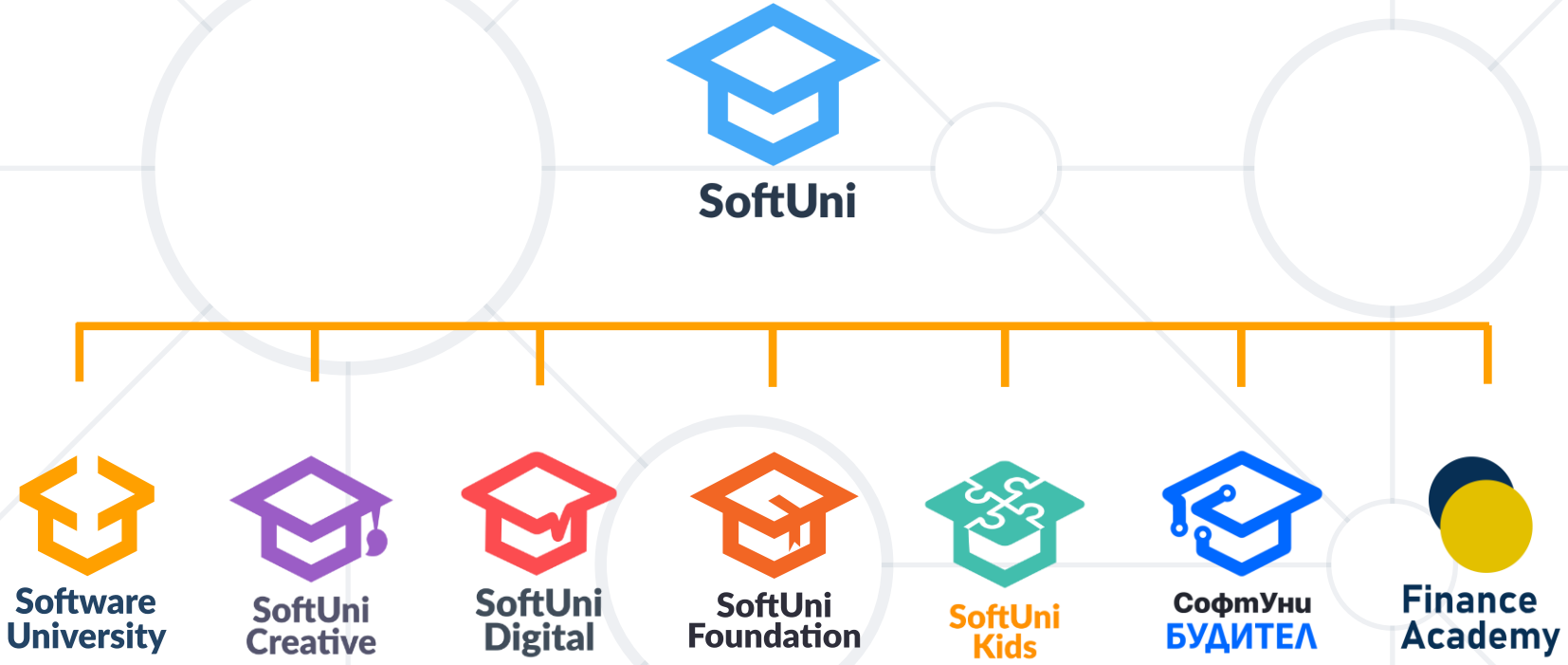
```
from dash import Dash, html, dash_table, dcc
import pandas as pd
import plotly.express as px

df = pd.read_csv("...")
app = Dash()
app.layout = [
    html.Div(children="My First App with Data and a Graph"),
    dash_table.DataTable(data=df.to_dict("records"), page_size = 10),
    dcc.Graph(figure = px.histogram(df, x = "continent", y = "lifeExp",
histfunc = "avg"))
]
if __name__ == "__main__":
    app.run(debug = True)
```

# Summary

- {Dev, Data, ML}Ops Fundamentals
- Work Processes
- Data Pipelines
- Scaling Workflows to Larger Datasets
- Monitoring and Dashboards

# Questions?



# SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank
Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR
Serving the high achievers

SOFTWARE GROUP

PHAR VISION

# Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

# License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg