Inhalt

[Methodology 2](#_Toc57540671)

[**Reproducibility** 2](#_Toc57540672)

[GitHub 2](#_Toc57540673)

[Docker 2](#_Toc57540674)

[Visual Studio Code 2](#_Toc57540675)

[Python 2](#_Toc57540676)

[Database SQLite 2](#_Toc57540677)

[**Collaboration** 2](#_Toc57540678)

[Asana 2](#_Toc57540679)

[LaTex 2](#_Toc57540680)

[**Workflow** 2](#_Toc57540681)

# TechStack

In his article Grudin (2010) depicts the changing “*seasons*” of artificial intelligence (AI), starting in 1950. He ends in 2009, where his out view for the upcoming years is a new “*AI Summer*” (Grudin 2010; Zhang et al. 2020, S. 24). In the context of AI, however, this article is already quite “old”. Since 2009, the end of Grudin (2010) analysis of AI seasons, a lot has happened in the field of AI. More and more businesses started to use machine learning technologies and artificial intelligence to ameliorate their businesses, meaning that these technologies also had to produce figures and numbers where decisions could be based on (Zhang et al. 2020, S. 24). In 2015 the decision-making process was mostly based on business reports, produced by “data scientists/data analysts”, which leveraged the then state-of-the-art algorithms (Dancho 2019). Already five years later decision-making processes require apps, which can visualize different findings, are interactive and can be used throughout all levels of the company, no matter the persons skill set (Dancho 2019). However, there is a great disparity between creating powerpoint presentation, which visualize certain findings, or developing certain apps for decision-making processes. This technologic progress led to a closer cooperation between different departments, like Data Scientist, Data Analysts and the so called “DevOps”, since different skills are required to develop an app (Zhang et al. 2020, S. 22, 2020, S. 23).

This fast development and the now needed cooperation brought a various of challenges with it. One of them is reproducibility. This can be defined as the need for “*Programmers [which] need applications to run no matter where they are deployed”* (Dancho 2019). Imagine a Data Scientist developing an algorithm which he wants to implement in the App, which is in turn managed by a Data Analyst. In that case the Data Analyst has to get the suggested algorithm from the Data Scientist up and running on his machine (and eventually in the cloud) (IBM 2019). One way of taking care of the before outlined problem is the containerization of development environments (IBM 2019). The biggest benefit of this solution is, that when an application is developed within such a container, that is portable between different systems (IBM 2019). Moreover, theses containers are easy to install and to use (IBM 2019). The idea of containerization, however, is only one brick in the foundation for successful collaborative work. Which tools where used within the project *Bikerus* are thus explained the chapter **Reproducibility**.

The aforementioned increased complexibility in creating and presenting decision-making useful information, led to another problem, namely the organization of collaboration (Zhang et al. 2020, S. 23). Different tasks are done by different people or even different departments. In order to be always updated about the current status of the development, a need for a project management tool, even within a Data Science department, arose (Zhang et al. 2020, S. 25). The tools used in the project *Bikerus* are outlined in the chapter **Collaboration**.

This chapter will end with an overview of a typical workflow using the before outlined tools. It will show, how to overcome the problems of reproducibility and collaboration effectively.

## **Reproducibility**

Zhang et al. (2020, S. 25) already state that nowadays a tool named *Github* is a widely used tool within the data science community. Its is used for code version control and it enables users to rollback their code in case of errors, merge their developed features with others people code and create an open code base, so that the code is publicly available. This tool will be explained in the subchapter **Github**.

As already slightly outline the core of reproducibility are containers. How do these work, what is behind their logical and how those can be used is described in the subchapter **Docker & Visual Studio Code**.

Notes:

* Git (benefits)
* Docker & VSCode (benefits)
* Python
* SQLite

As earlier defined, this is the need for programmers to run their application on all platforms, regardless of the platforms operating system (Dancho 2019).

### GitHub

### Docker

### Visual Studio Code

### Python

### SQLite

## **Collaboration**

### Asana

### Overleaf

## **Workflow**

Literaturverzeichnis

Dancho, Matt (2019): Part 3 - Docker for Data Scientists. (A Top Skill for 2020). Hg. v. business-science.io. Online verfügbar unter https://www.business-science.io/business/2019/11/22/docker-for-data-science.html, zuletzt aktualisiert am 22.11.2019, zuletzt geprüft am 29.11.2020.

Grudin, Jonathan (2010): AI and HCI: Two Fields Divided by a Common Focus. In: *AIMag* 30 (4), S. 48. DOI: 10.1609/aimag.v30i4.2271.

IBM (2019): The Benefits of Containerization and What It Means for You, 06.02.2019. Online verfügbar unter https://www.ibm.com/cloud/blog/the-benefits-of-containerization-and-what-it-means-for-you, zuletzt geprüft am 29.11.2020.

Zhang, Amy X.; Muller, Michael; Wang, Dakuo (2020): How do Data Science Workers Collaborate? Roles, Workflows, and Tools. Online verfügbar unter https://arxiv.org/pdf/2001.06684.