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 other people’s code and create an open code base, so that the code is publicly available. How this tool works, will be explained in the subchapter **GitHub**.

As already slightly outlined, 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**.

As mentioned earlier, in 2020 decision-makers demanded full functioning application for business decision (Dancho 2019). One tool which is mostly used for such tasks is Python. It has the ability to “*store, access, and manipulate data* *[out of the box]*” and is thus widely used (Chiu 2019). The various libraries for visualization, statistics, and standalone applications (e.g. flask) is one reason why this programming language was used within this project *Bikerus*. More about **Python** will be explained in the same named subchapter.

The last subchapter of this part will be about *SQLite*. When starting a Data Science/Analytics process, mostly a lot of files are created. These mostly have similar names like *X\_train\_1*, *X\_train\_2*, and one can easily lose sight of what are the files, which are relevant for the main workflow. Over the long run this procedure creates unnecessary chaos, not only regarding naming but by just the vast amounts of files which were created. To deal with this, a SQL database was used. The advantages of such a way of storage are discussed in the subchapter **SQLite**.

### GitHub

The word *GitHub*, as one can already guess, is the addition of two different words – *Git* and *Hub*. In order to understand this portmanteau, it is necessary to understand the function of *Git* first. The tool *Git* is a version-control tool and has powerful implications. In particular, there are two specific usages, which are important.

The first use is the personal one. Each person, who is programming, can use *Git* for their own. This program will create a snapshot of the written code and will save it in a *.git*-folder within the original folder. That has the benefit, that if for some reason a program breaks, one can always go back, or see even the difference between the working and the development code state. This makes it easier to identify the broken part in the written code. To each taking of the snapshot (commit) the programmer himself/herself can write a commit message, in which she/he clarifies what differences has been made.

Another use case, as Zhang et al. (2020, S. 25) points out, is that nowadays there are various departments involved in developing applications. A showcase for this is the development of a business intelligence app. In such a development process are involved the data analysts, which develop the different plots and their layout, the data scientists, which develop the models to produce the later on visualized data and the development engineers, which provide the fundamentals of the app. This list in incomplete, however, it showcases, that a lot of people work on different features within the same project. In order to track their steps and, most importantly, merge their code, *Git* can be used. It provides the possibility to easily join different code parts and raises the user’s attention when there is a conflict.

Overall, everyone how is working on a project has a copy (repository) from the main project on their local machine. He/She develops their features using the given source code and track her/his changes over time. *GitHub* provides a free remote service, where all the code and their different branches can be stored. This remote repository enables the members of the team to work collaboratively more efficiently.

### Docker & Visual Studio Code

### Python

### SQLite

## **Collaboration**

### Asana

### Overleaf

## **Workflow**

Literaturverzeichnis

Chiu, Joyce (2019): Why Your Company Needs Python for Business Analytics. Online verfügbar unter https://www.datacamp.com/community/blog/why-your-company-needs-python-for-business-analytics, zuletzt aktualisiert am 03.12.2019, zuletzt geprüft am 03.12.2020.

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.