MOTIUS
WE R&D.

Enabling Production Grade MLOps

4 guiding principles

Agenda

Scale Your MVP

MOTIUS
WE R&D.

01. End-to-End Process Model
The first principle

02. About Automation
The second principle

03. Experiment Tracking And Reproducibility
The third principle

04. Testing And Monitoring
The fourth principle

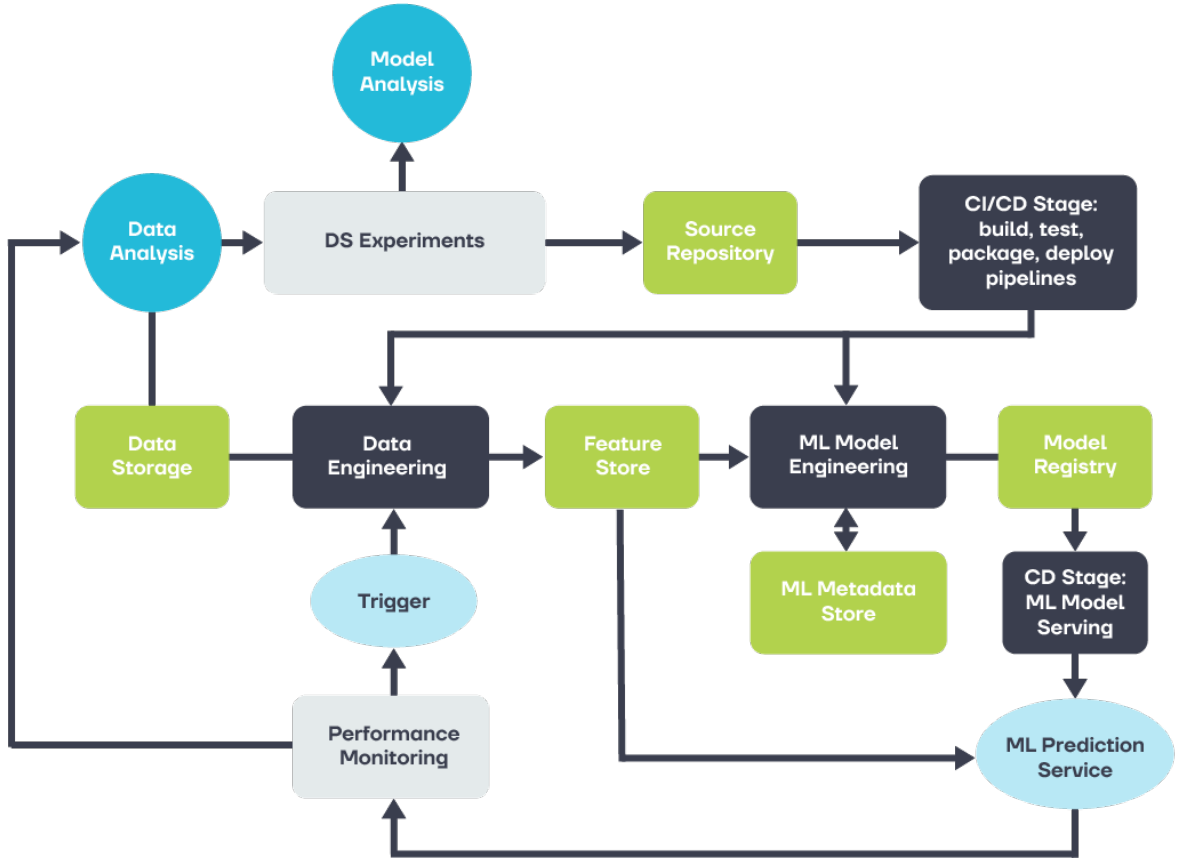
05. Time for Questions
Contact

4 Principles

For enabling production-grade MLOps

Wondering how to leverage the full potential of AI and ML approaches? Not only in exploration and research but also in production-grade settings? Then you'll need MLOps and these four guiding principles as a **standard along the entire lifecycle** of your use case. Be aware: these four principles are the necessary baseline for successful MLOps. However, you'll also need to **establish a lived and cultivated practice** within all teams working along the data lifecycle and ML applications. The same applies to their enabling systems.

This high-level architecture to the right illustrates the involved components and their relations to each other for an MLOps capable system.



MOTIUS
WE R&D.

01.

End-to-End Process Model

End-to-End Process Model

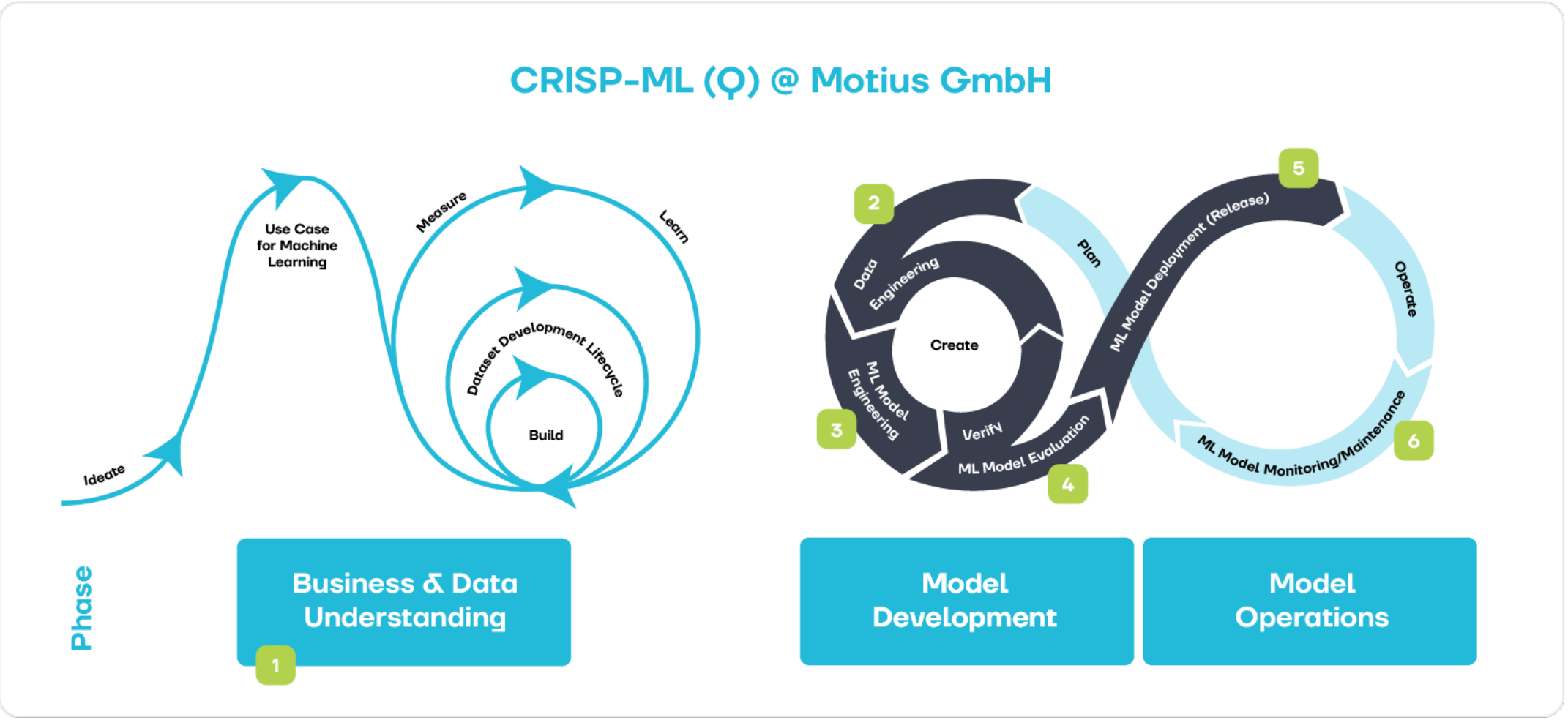
The first principle

The first principle focuses on the process steps needed along the lifecycle of an ML application. We encourage development teams to **take the full picture into account**. Starting with the *"Ideation"* as part of the *"Business & Data Understanding"* part, up until the *"Deployment"* and *"Monitoring & Maintenance"*, as part of the "*Model Operations"*. Given the iterative nature of ML applications, MLOps can be considered as an infinite (or at least long-lasting) cycle of further improvements.

Sounds great, but how does it work? At Motius, we work with the following (see next slide) end-2-end process view following the **CRISP-ML(Q) methodology**. And we suggest you to do so, too. It ensures iterative improvements after the initial deployment of the created ML model, based on collected end-user feedback.

The Process Steps of CRISP-ML(Q)

A brief explanation



The Process Steps of CRISP-ML(Q)

A brief explanation

1. **Business and Data Understanding**
   Identify problems to solve using ML, collect and explore data, and ensure the feasibility of the project.

2. **Data Engineering & Preparation**
   Consolidate and transform data into something usable in the modeling stage.

3. **ML Model Engineering**
   Develop models to solve problems and business objectives.

4. **Evaluation**
   Determine if the model is ready for deployment or if further iterations are necessary.

The Process Steps of CRISP-ML(Q)

A brief explanation

5

Deployment

Integrate the model into a larger system where the model predictions are actually used to drive further applications/processes

6

Monitoring and Maintenance

Track certain metrics over time to identify if the model needs to be retrained/updated due to changes in the system.

Some components become more relevant with a rising level of maturity, but the overall structure remains the same. This means it **can also be used in the early stages** of development when there are still a lot of experiments needed to solve the business problem at hand.

02.

About Automation

Automation

The second principle

The second principle is about automation. Within the development process of ML models and their serving for use by applications and downstream services, we have two major areas to increase the level of automation.

- **Data & ML Model Creation Pipelines**

  Sure, the “Data Engineering” part within the ML area might not be as complex as overall data engineering pipelines within your company because ideally they build on already (semi-)prepared data. But they should still run in an automated, possibly scheduled way, based on a data pipeline orchestrator. That also applies to the processing pipeline that handles the “ML Model Engineering” to prepare and train the ML models.

- **CI/CD**

  Like in software development, we want to leverage CI/CD pipelines for automated builds, tests, and deployments for data and ML workloads. This includes artifacts needed for the data processing and services created for serving that require additional measures (i.e. quality gates).

MOTIUS
WE R&D.

03.

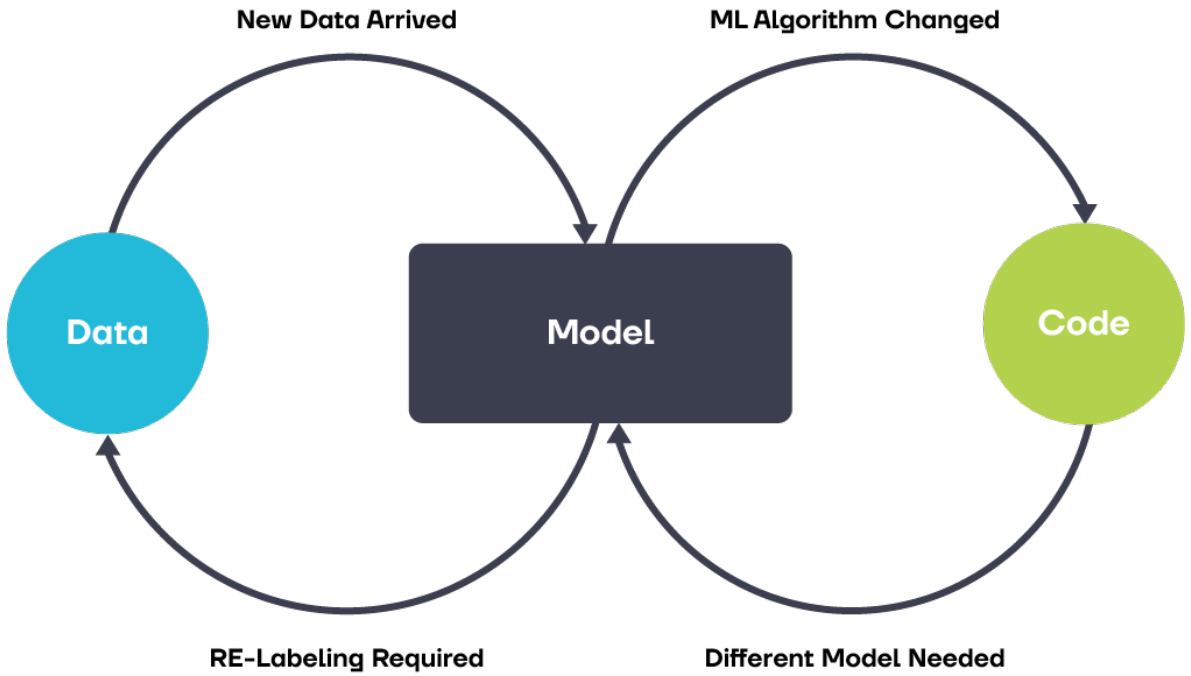
Experiment Tracking
And Reproducibility

Machine Learning Applications

The third principle

The third principle focuses on **reproducibility and tracking** of experiments. ML systems have changing data, model, and code, which have to be actively managed and overseen to keep the ML results reproducible. Therefore, we highly depend on experiment tracking, code versioning, and data lineage.

An essential element of data lineage is the versioning of the data, which links this topic with cataloging data and governance processes also needed throughout the entire data processing journey.



MOTIUS
WE R&D.

04.

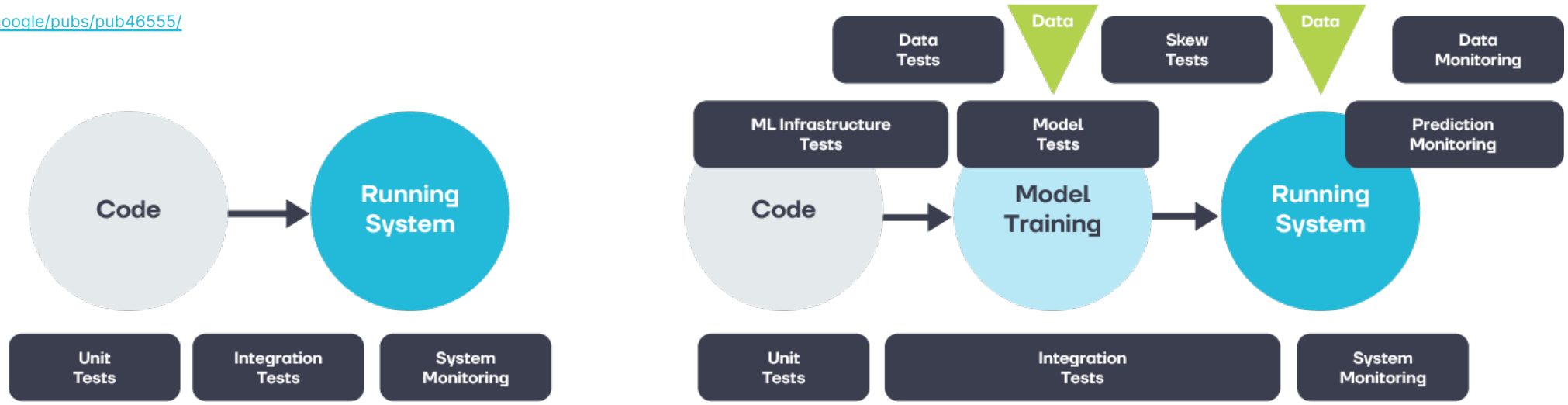
Testing
And Monitoring

Testing and Monitoring

The fourth principle

The fourth and last principle targets testing and monitoring. Below you can see a simplified view of the difference between a classical software system and one that involves machine learning. The list of possible metrics and tests, to monitor, track and test for is very extensive and can be implemented over time. Equivalent to reproducibility (see above), we have to **take care of data, code, and model simultaneously**. This means we need new kinds of tests and that we have to monitor further aspects. Remember we pointed out the importance of automating your tests earlier? Well, this is one of the reasons why automation is so essential: the overall amount of testing is enormous.

Source:
https://research.google/pubs/pub46555/

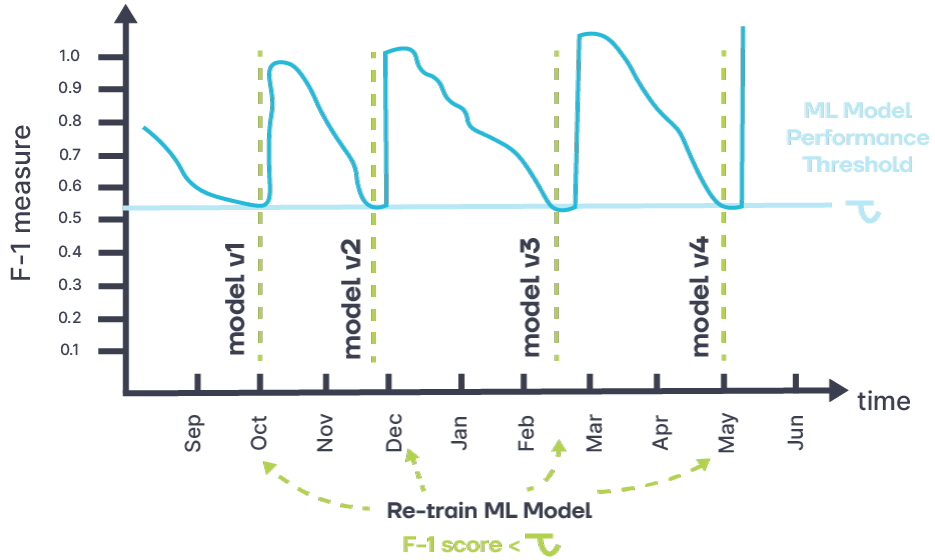


Monitoring Drift Detection

Sounds too abstract? Let's illustrate this in more detail by outlining an issue that needs to be monitored within an ML system.

One particular subject in the monitoring of ML models is detecting different kinds of drifts, mainly drift of data and drift of concept.

> **Data drift** = when the data distribution during the training time of the model does not represent the data distribution at inference time anymore. This can happen if it takes e.g. a full year from data collection until the model is used within a use case in production.



This data drift can **decrease model performance** as the model might do a good job overall but has weaknesses for certain classes it is supposed to predict. If those classes are now represented more in the data set seen during production than it was at training time, the overall model performance will decrease. This change in the data distribution needs to be monitored over time to **assess if and when retraining the model is necessary** to keep its performance above a set threshold.

MOTIUS
WE R&D.

04.

Time For Questions

Are You Interested?

Matthias Wissel

Senior Machine Learning Engineer @ Motius

LET‘S TALK

Walter-Gropius-Straße 17
80807 München

info@motius.de

www.motius.de