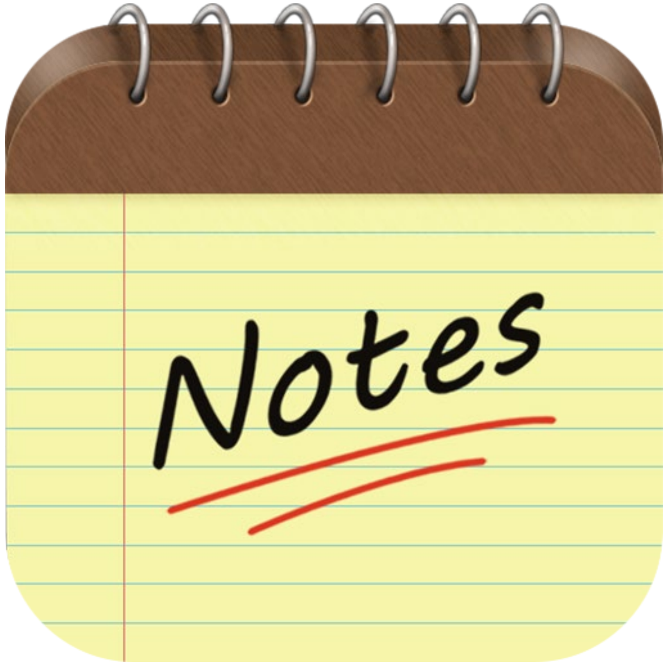
Software Engineering for AI-Enabled Systems

SOFTWARE SYSTEME

UNIVERSITÄT LEIPZIG

Prof. Dr.-Ing. Norbert Siegmund
Software Systems

Organisation



Dates:

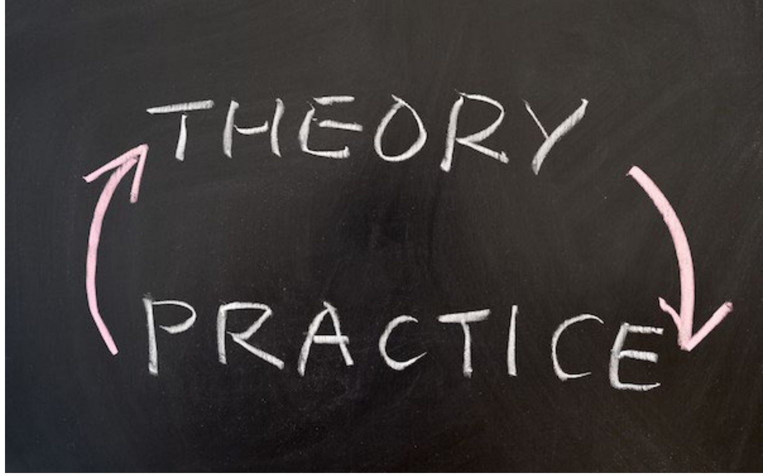
- Lectures on Monday 11 - 13 and Tuesday 9 - 11
- Exercises on Friday 11 – 13 and 13 - 15

Lectures (slides in English, spoken language your choice):

Exercises:

- Practical implementation and setup of SE/ML projects
- Interactive tutorials on key frameworks and tools
- Preparation and discussion on project

Projects I



General information:

- 10 teams with 4 members (2 data science students , 2 CS students)
- No grading, but **prerequisite** for doing the exam
- Goal: Transfer of theory to a practical setting
- One project with two phases

First phase: Inference pipeline using a given ML endpoint

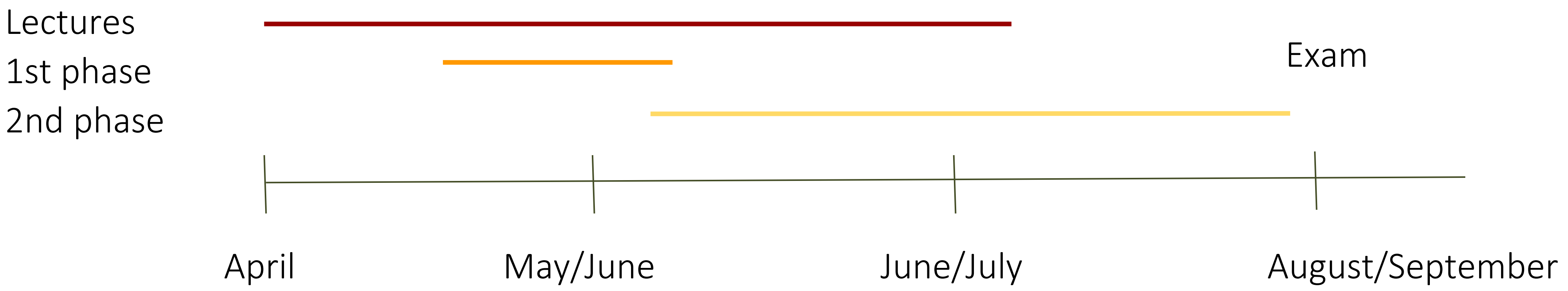
- Automate a pipeline from data scraping / gathering to sanitizing over feature engineering till output generation
- Focus is on the tooling and not on the business process
- Due in end of May
- Possible topics will be announced in the accompanied excercise

Projects II

Second phase: Integration of the previously defined model into a software system

- Focus is on the whole project life cycle
- Include proper requirements engineering and software components
- Interaction between software components and AI component
- Feedback loop from deployed software to improve inference pipeline
- Expected delivery: End of July

Timeline:



Code of honor

Valid:

- Get help & resources from the Web / forums, etc., **but cite them all**
- Discuss questions & topics with classmates
- Reuse existing solutions in the project, **but make your contributions clear**
- Publishing your project
- Use of ChatGPT to **support** your work
- Ask us if unsure in anyway

Forbidden:

- Asking another person (or AI) to do your project
- Copying other solutions
- Pretending that someone's solution is yours
- Sharing your solution with others

Exam

120 minutes going through *all* topics of the course

No coding examples (this is what the projects are for)

If you follow all lectures (in person!), participate in the exercises and project, and recap everything prior to the exam, you will manage it!

Prerequisites:

This is not an ML/AI lecture, so you need

- Basic experience in ML
  - Regression vs. classification (know some models)
  - Supervised vs. unsupervised learning
- Python or Java
  - Know how to set up a simple ML project (e.g., via Jupyter Notebooks)
  - Know how to manipulate and visualize data (read data with pandas, manipulate with numpy, plot with matplotlib)
- Software engineering basic knowledge
  - DevOps pipeline (experience with Git, heard or used Docker, implemented simple CI/CD)
  - Testing (know the differences and implemented at least one of unit, integration, acceptance)
  - Know basic software architectures (microservices, client-server, etc.) and patterns
- Brief recap of these topics will follow in the upcoming lectures and exercises

Interactivity Check

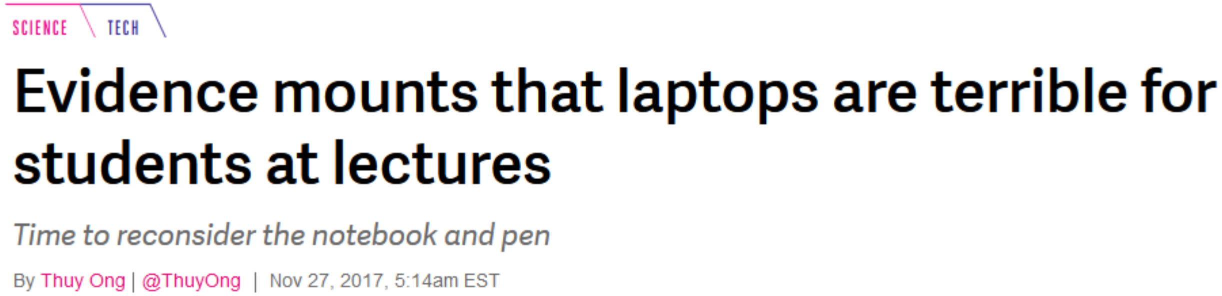
Live coding and (industry) workshops throughout the course, so:

Who has a notebook / laptop available for use during the lecture?

What OS do you have installed?

What language and virtualization environments do you have installed?

But, remember:



SCIENCE \ TECH

Evidence mounts that laptops are terrible for students at lectures

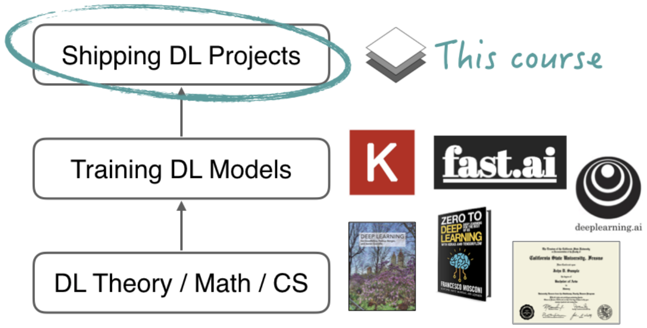
Time to reconsider the notebook and pen

By Thuy Ong | @ThuyOng | Nov 27, 2017, 5:14am EST

Scoping and Comparison

Full Stack deep learning

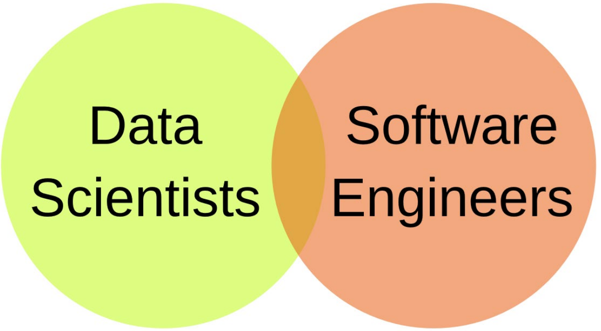
by Sergey Karayev and Josh Tobin and Pieter Abbeel



Data science perspective
Deep learning basics
ML project essentials

Software Engineering for AI-Enabled Systems

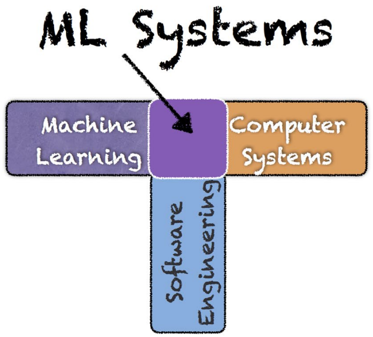
by Christian Kästner



SE perspective
ML basics
Discussion-driven

Machine Learning Systems

by Pooyan Jamshidi

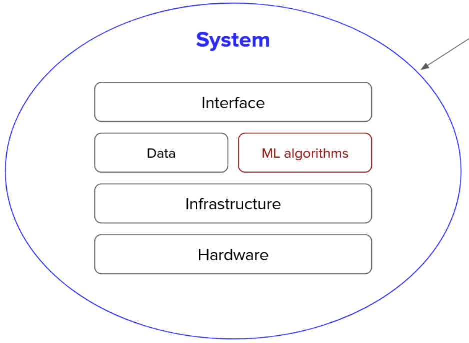


Systems perspective
ML basics
ML HW, Testing&Setup

Machine Learning Systems Design

by Chip Huyen

We'll learn about all of this



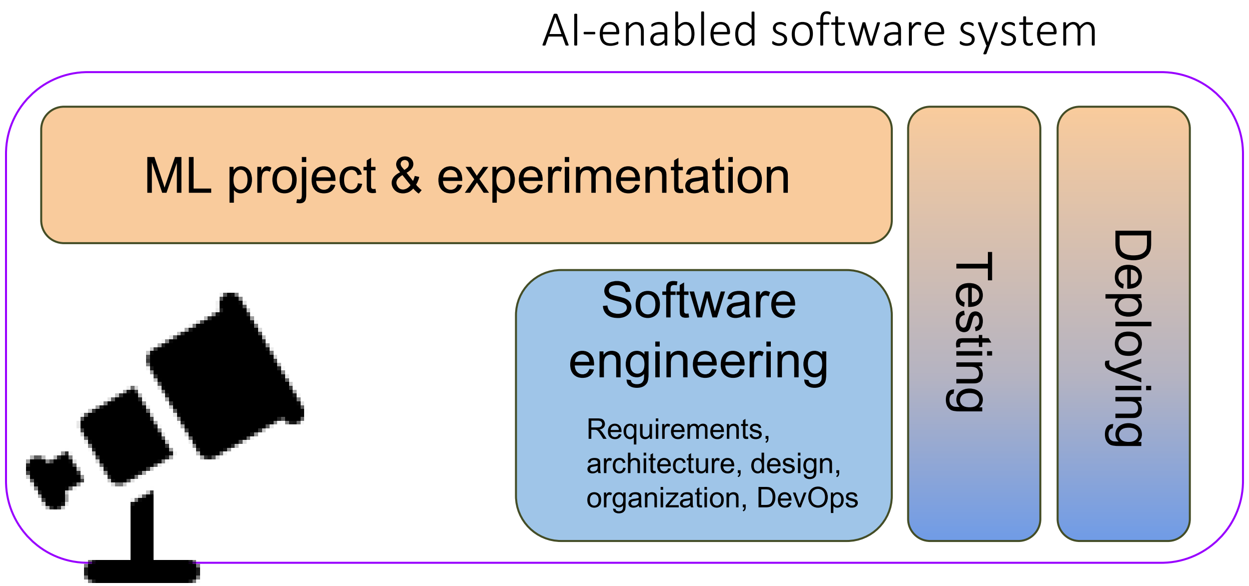
Data science perspective
MLOps resource

By all means, follow these lectures as they are a fantastic source of information!

Scoping and Comparison

SE Perspective on the interface of data science (DS) and software engineering (SE)

May divert from ML / DS perspective, which is okay and intended!


AI-enabled software system
ML project & experimentation
Software engineering
Requirements, architecture, design, organization, DevOps
Testing
Deploying

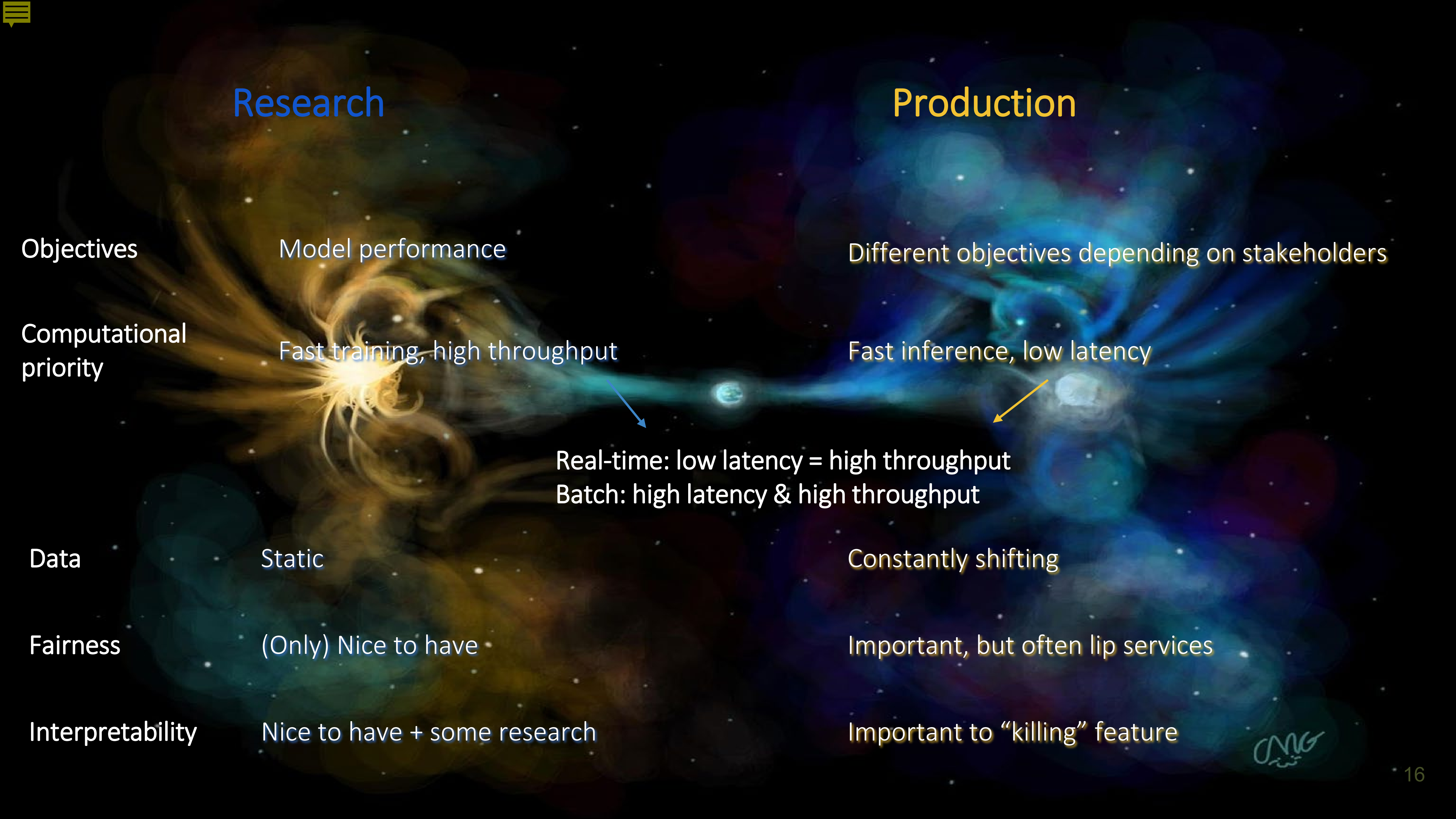
Differences in what you might expect

Research (academia):

- Model selection and training
- Accuracy metrics, no additional software requirements
- Clean data sets, single goal
- No maintenance, real deployment, user feedback, and bug fixing pressure

Production (industry):

- Data collection, cleaning, and labelling the main driver
- Handle multiple metrics and requirements (memory, inference time, accuracy)
- Polluted and biased data sets, goals with ethical implications
- Deployment issues, monitoring, retraining, maintenance of software-ai part



	Research	Production
Objectives	Model performance	Different objectives depending on stakeholders
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	(Only) Nice to have	Important, but often lip services
Interpretability	Nice to have + some research	Important to "killing" feature

Real-time: low latency = high throughput
Batch: high latency & high throughput

Course Topics

How do teams organize?

How to integrate an AI component with other software modules?

How can we automate data collection, training, deployment?

How to deploy and monitor?

How to keep track of experiments and version data and code?

How do we know that we develop the AI we actually need?

How to design and architecture the system?

How to design and build the entire software?

How to log, monitor, and display the state of a complex, distributed system?

How to increase resilience?

How to communicate among distributed modules?

How to test in such a landscape?

Disclaimer

Second time lecture here

The subject is new and continuously evolving (a year ago was not GPT-4 or ChatGPT!), we don’t have all the answers

We are all learning as welll!

We appreciate your:

enthusiasm for trying out new things

patience bearing with things that don’t quite work

feedback to improve the course

Resources & Contact

Resources:

- Slides will be available in Moodle
- See **slide notes for references** on papers, books, blog posts, etc. where the information originates from
- Additional slide after each major topic summarizing all resources

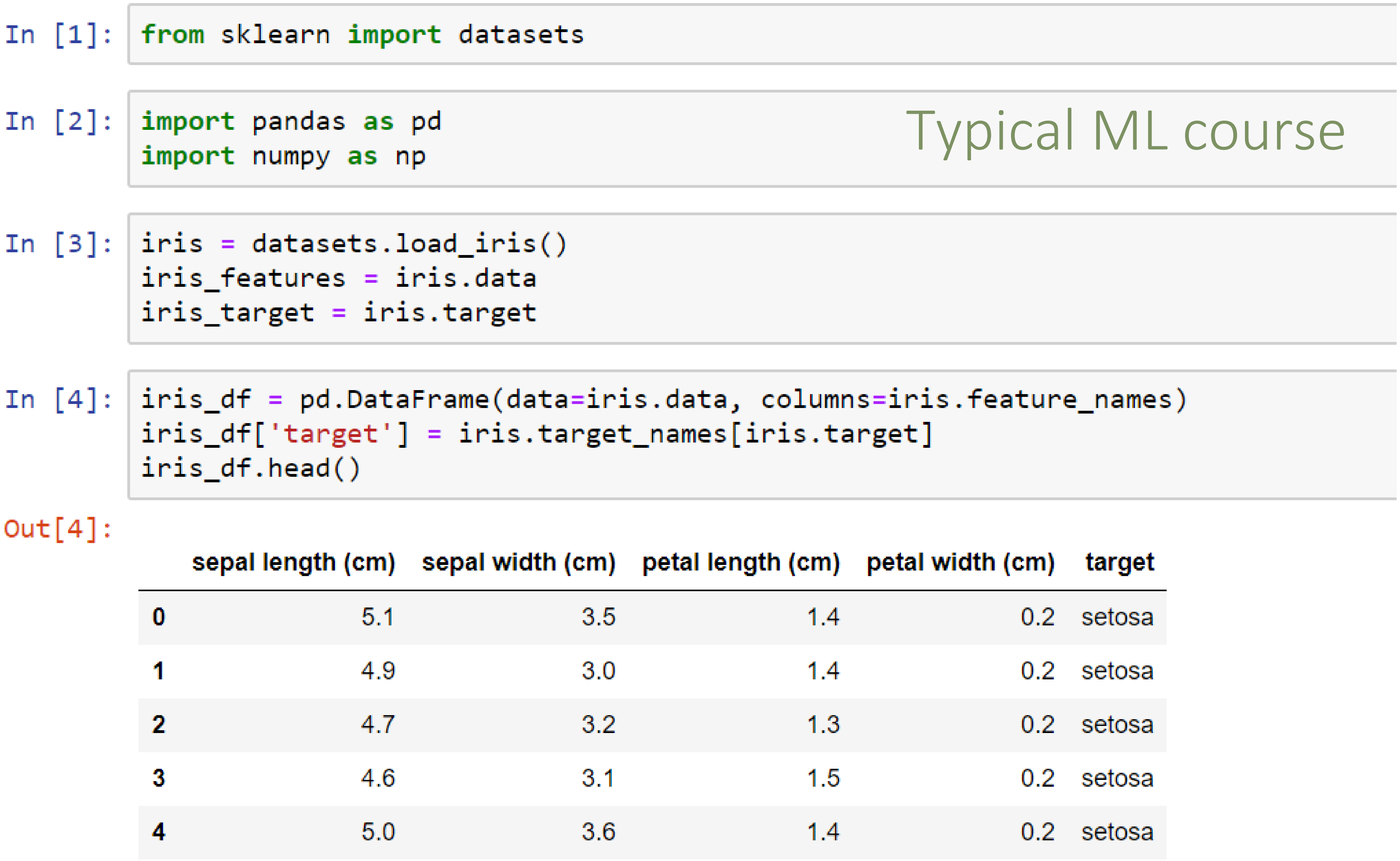
Contact:

- Forum in Moodle
- Chat with your supervisors at exercises
- Approach me directly after the lecture!

PART I:

Software Engineering for AI-Enabled Systems

Typical ML course



In [1]:
```
from sklearn import datasets
```

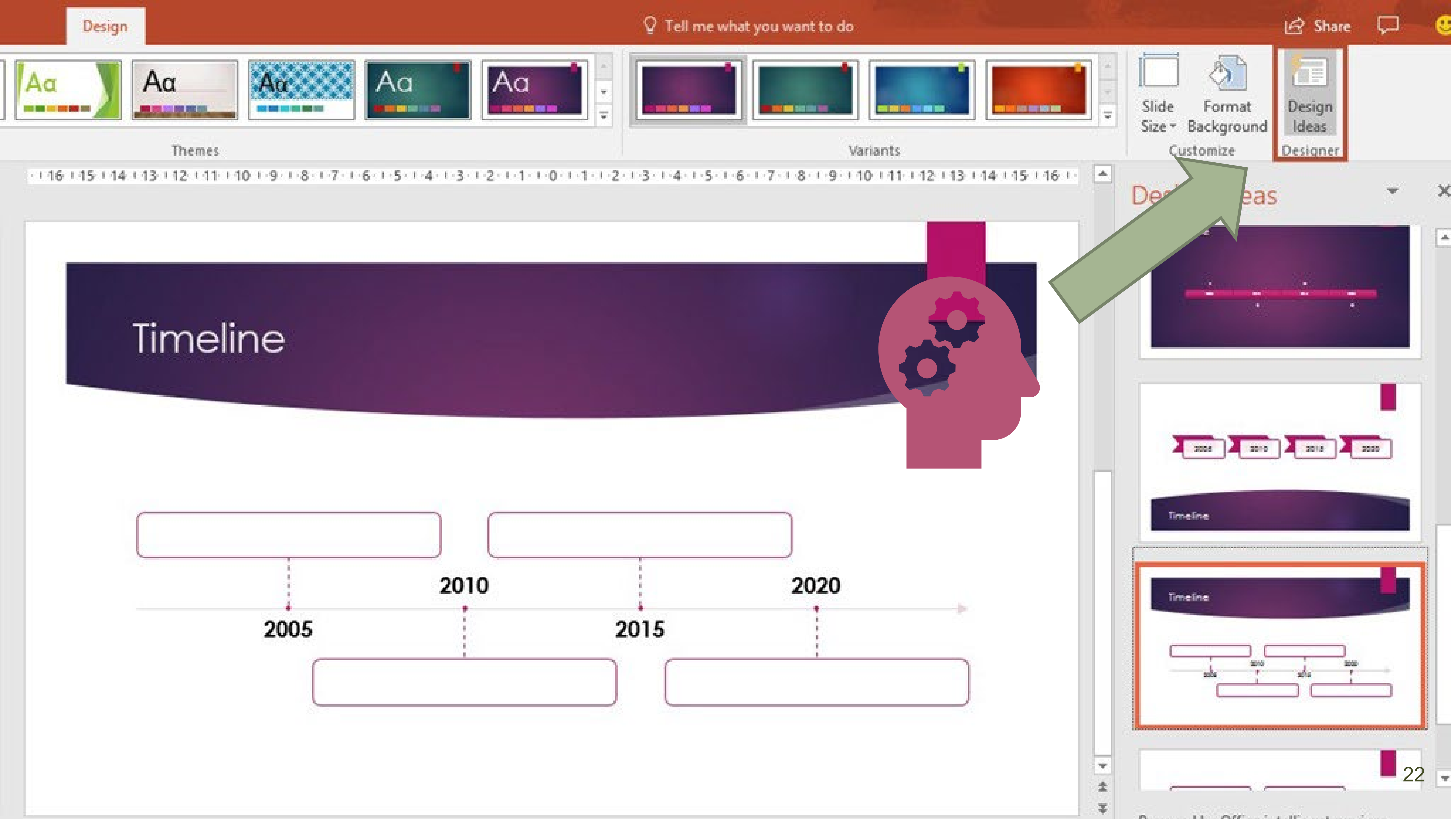
In [2]:
```
import pandas as pd
import numpy as np
```

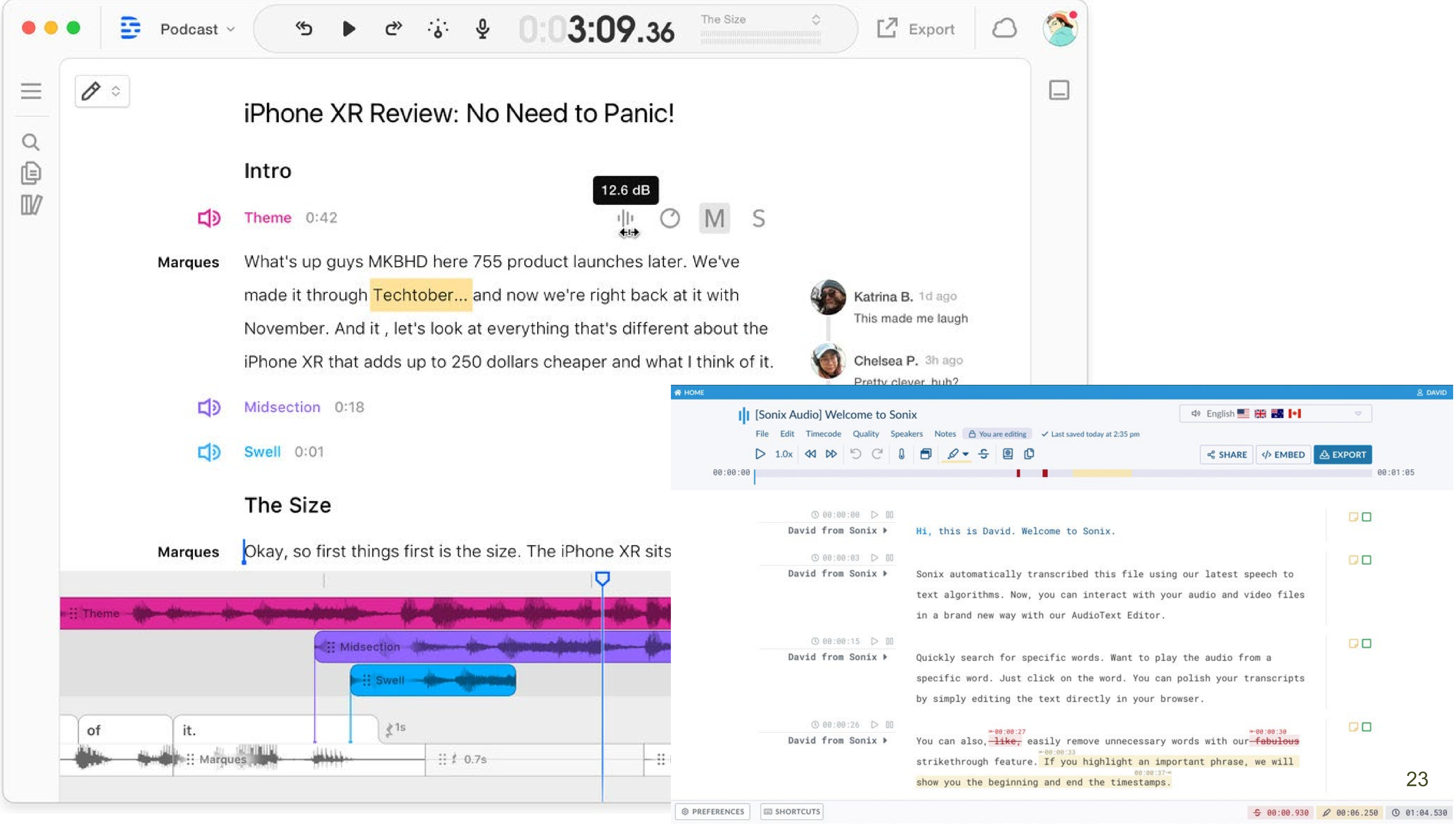
In [3]:
```
iris = datasets.load_iris()
iris_features = iris.data
iris_target = iris.target
```

In [4]:
```
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
iris_df['target'] = iris.target_names[iris.target]
iris_df.head()
```

Out[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa


Design
Tell me what you want to do
Share
Themes
Variants
Slide Size
Format Background
Design Ideas
Customize
Designer
Design Ideas
Timeline
2005
2010
2015
2020



iPhone XR Review: No Need to Panic!

Intro

Marques What's up guys MKBHD here 755 product launches later. We've made it through Techtober... and now we're right back at it with November. And it , let's look at everything that's different about the iPhone XR that adds up to 250 dollars cheaper and what I think of it.

Katrina B. 1d ago
This made me laugh

Chelsea P. 3h ago
Pretty clever, huh?

The Size

Marques Okay, so first things first is the size. The iPhone XR sits

[Sonix Audio] Welcome to Sonix

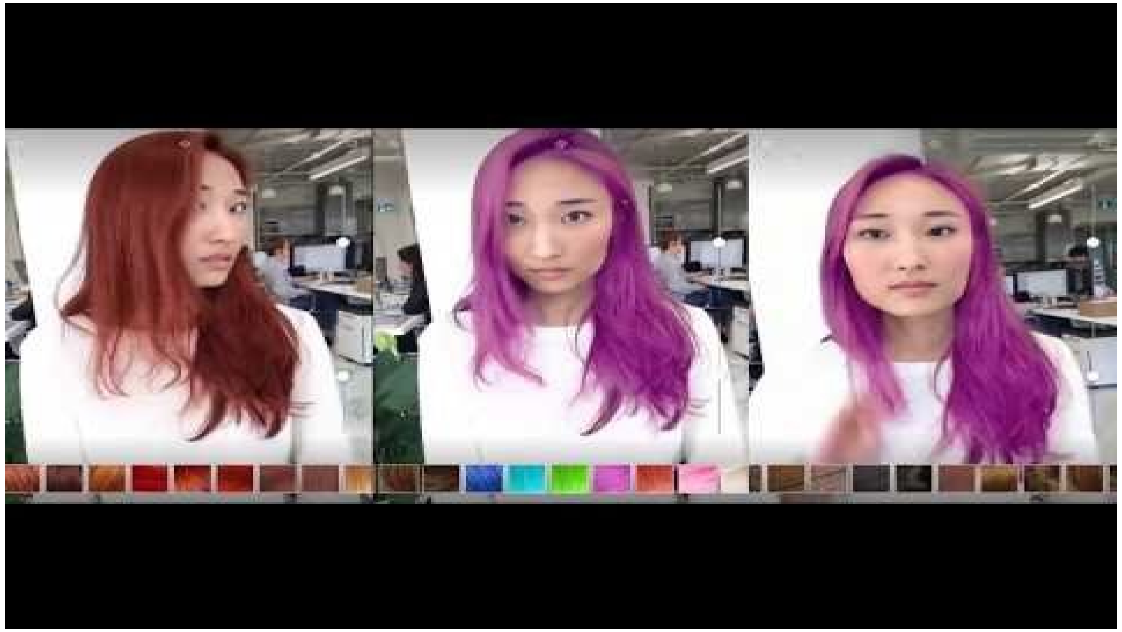
00:00:00 David from Sonix
Hi, this is David. Welcome to Sonix.

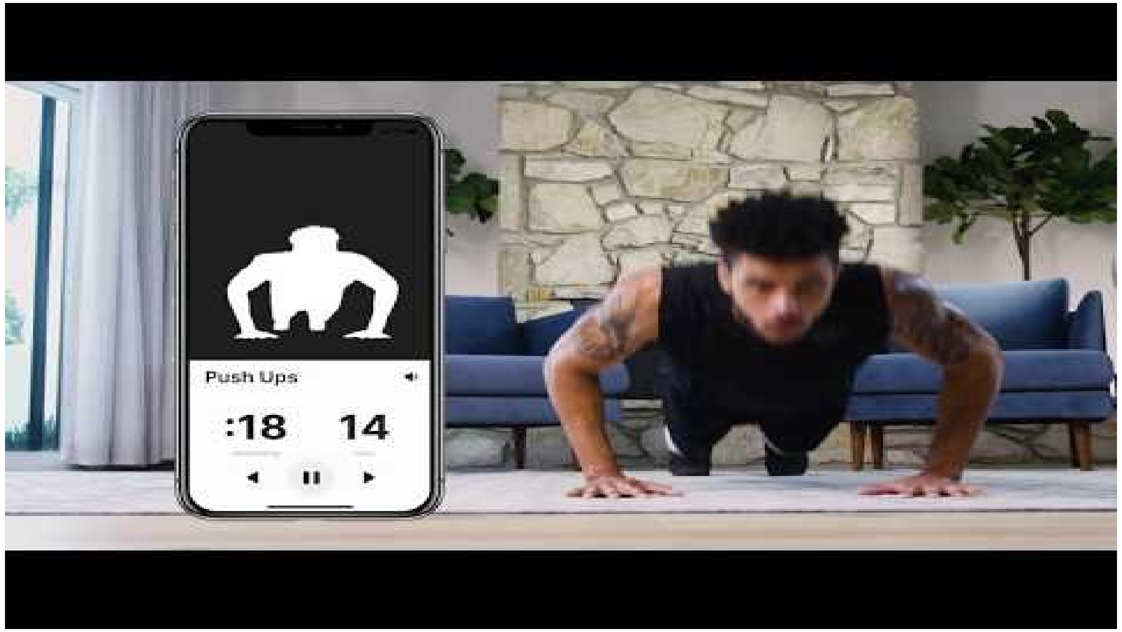
00:00:03 David from Sonix
Sonix automatically transcribed this file using our latest speech to text algorithms. Now, you can interact with your audio and video files in a brand new way with our AudioText Editor.

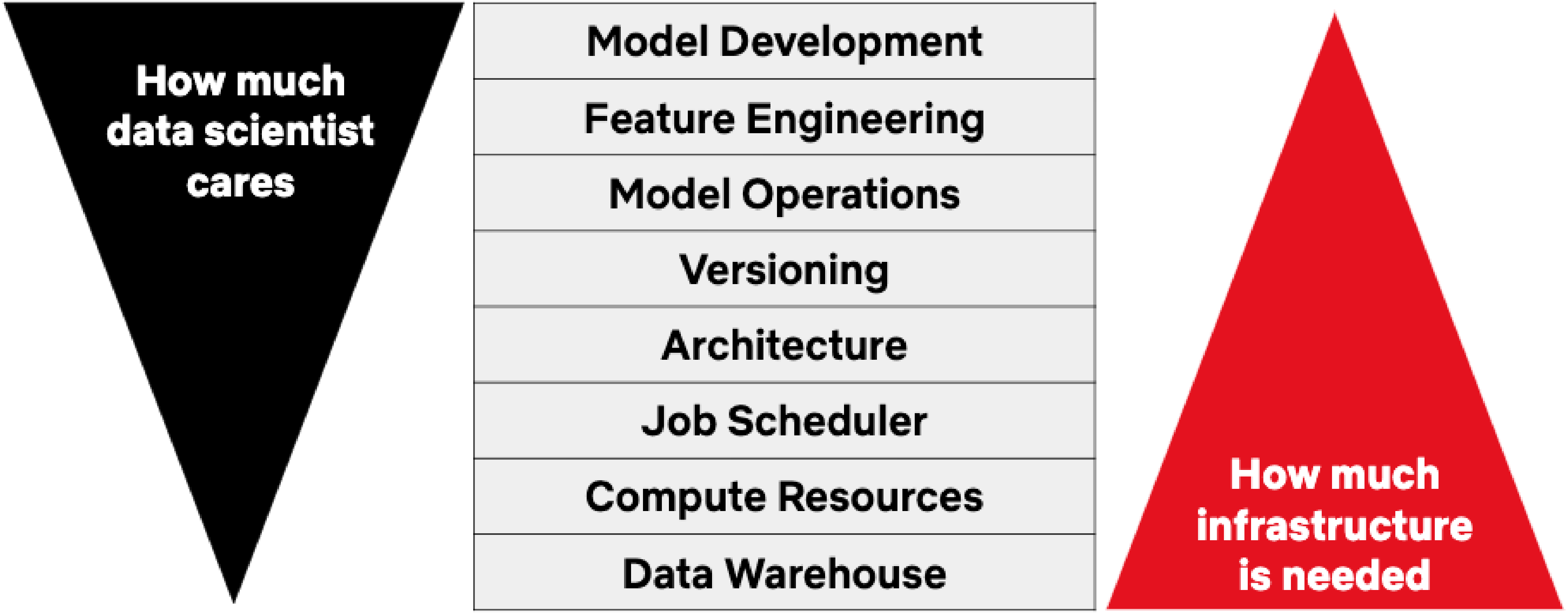
00:00:15 David from Sonix
Quickly search for specific words. Want to play the audio from a specific word. Just click on the word. You can polish your transcripts by simply editing the text directly in your browser.

00:00:26 David from Sonix
You can also, 00:00:27 ~~like,~~ easily remove unnecessary words with our 00:00:30 ~~fabulous~~ strikethrough feature. 00:00:33 If you highlight an important phrase, we will show you the beginning and end the timestamps. 00:00:37

Augmented SW Products with AI







How much data scientist cares

Model Development

Feature Engineering

Model Operations

Versioning

Architecture

Job Scheduler

Compute Resources

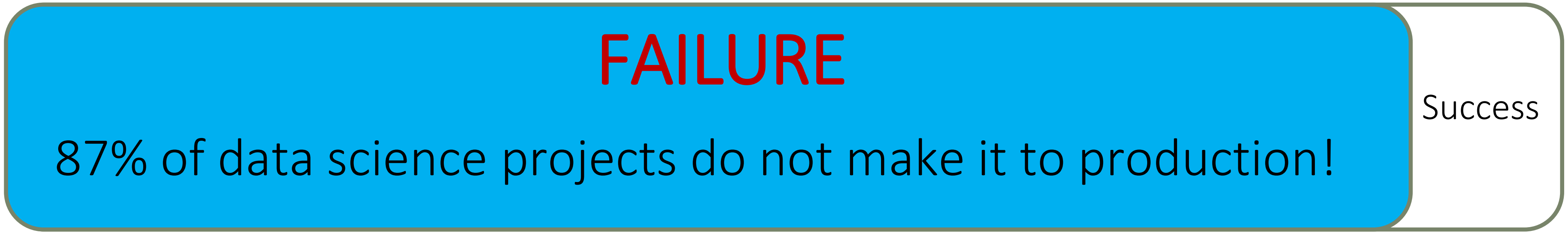
Data Warehouse

How much infrastructure is needed

Source: https://docs.metaflow.org/introduction/what-is-metaflow

Projects Failure Rate?

Guess, how many data science projects fail?



FAILURE

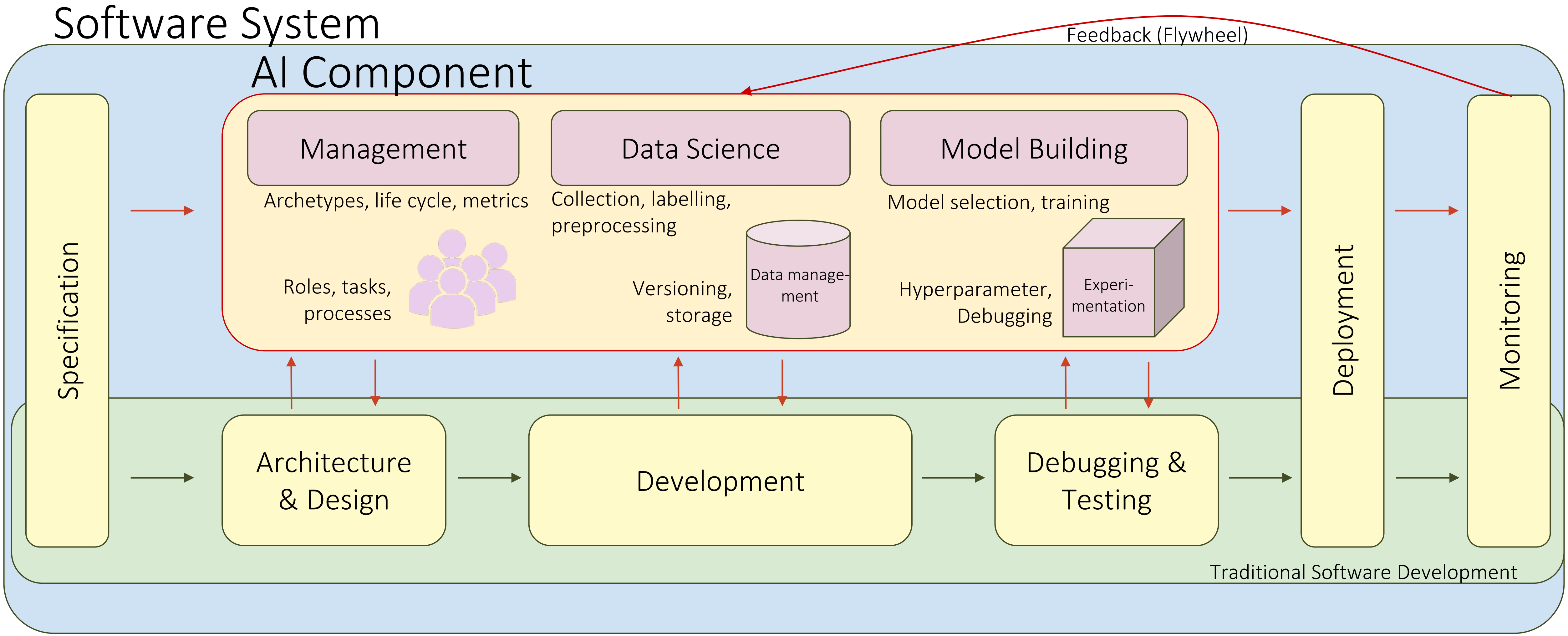
87% of data science projects do not make it to production!

Success

0%

https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production/

100%



Software System

AI Component

Feedback (Flywheel)

Specification

Management

Archetypes, life cycle, metrics

Roles, tasks, processes

Data Science

Collection, labelling, preprocessing

Versioning, storage

Data management

Model Building

Model selection, training

Hyperparameter, Debugging

Experimentation

Deployment

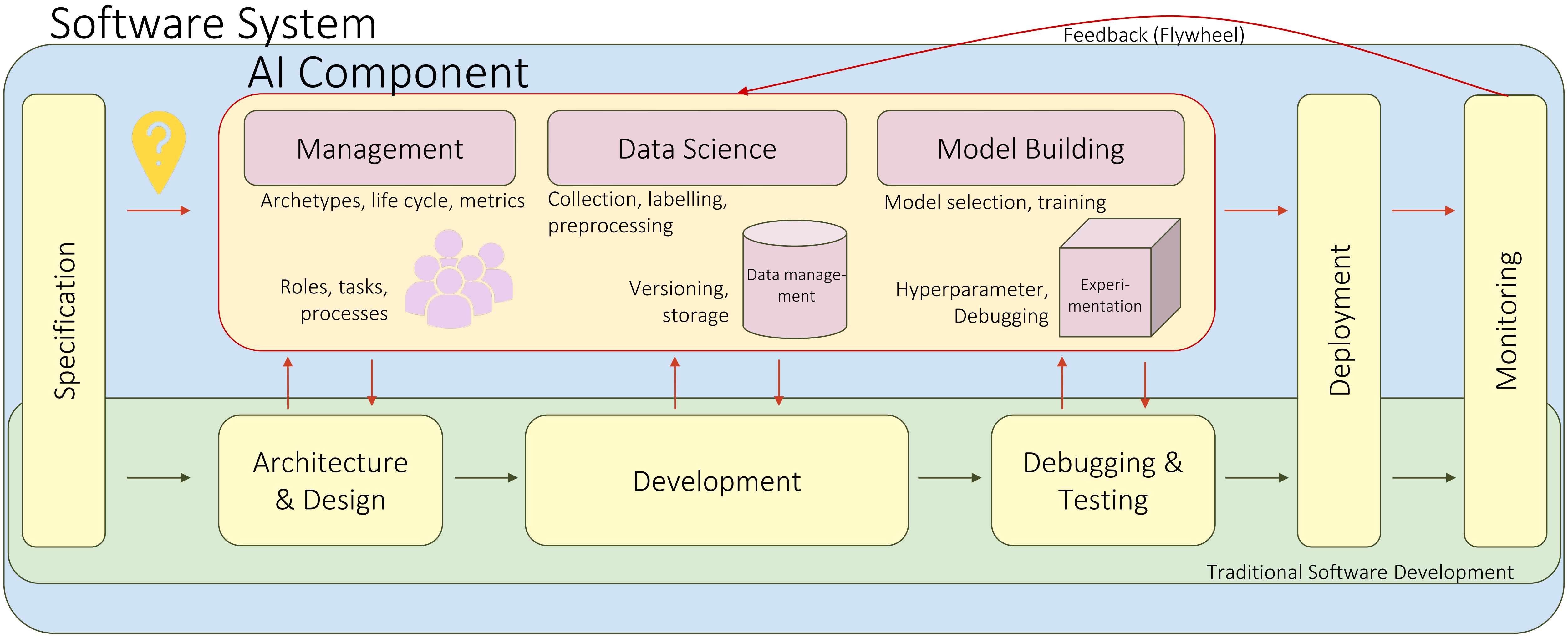
Monitoring

Architecture & Design

Development

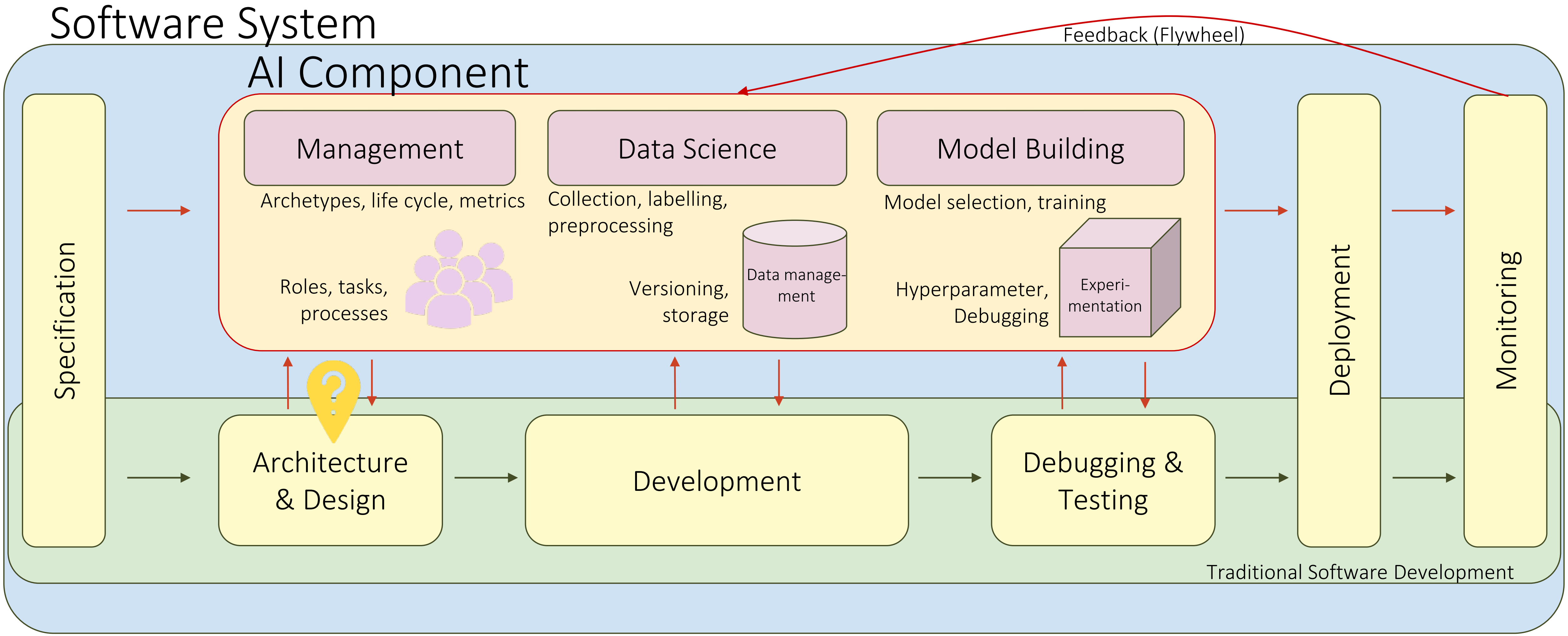
Debugging & Testing

Traditional Software Development



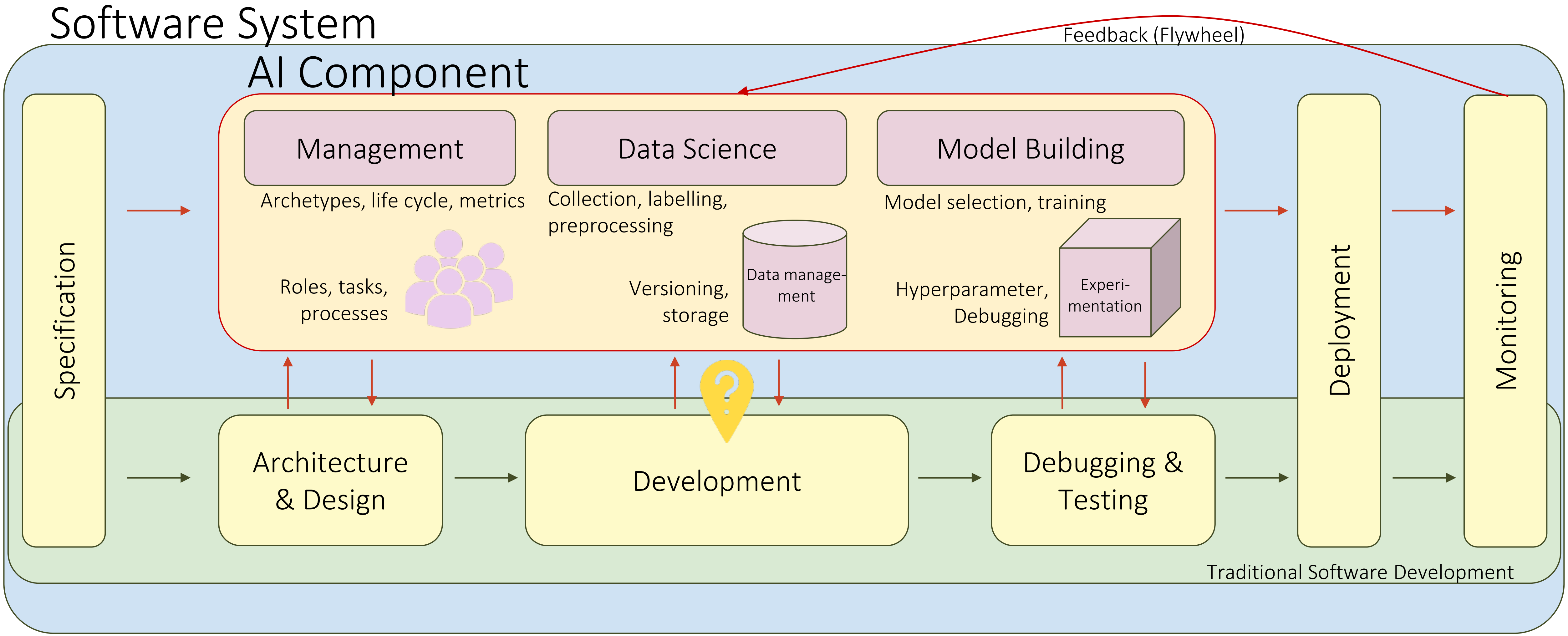
What is a proper specification for an AI model / component?

- Specify accuracy levels
- Specify success metrics related to business value
- Specify how to measure failure
- Specify how to detect failure
- Specify hardware and non-functional properties, such as inference time



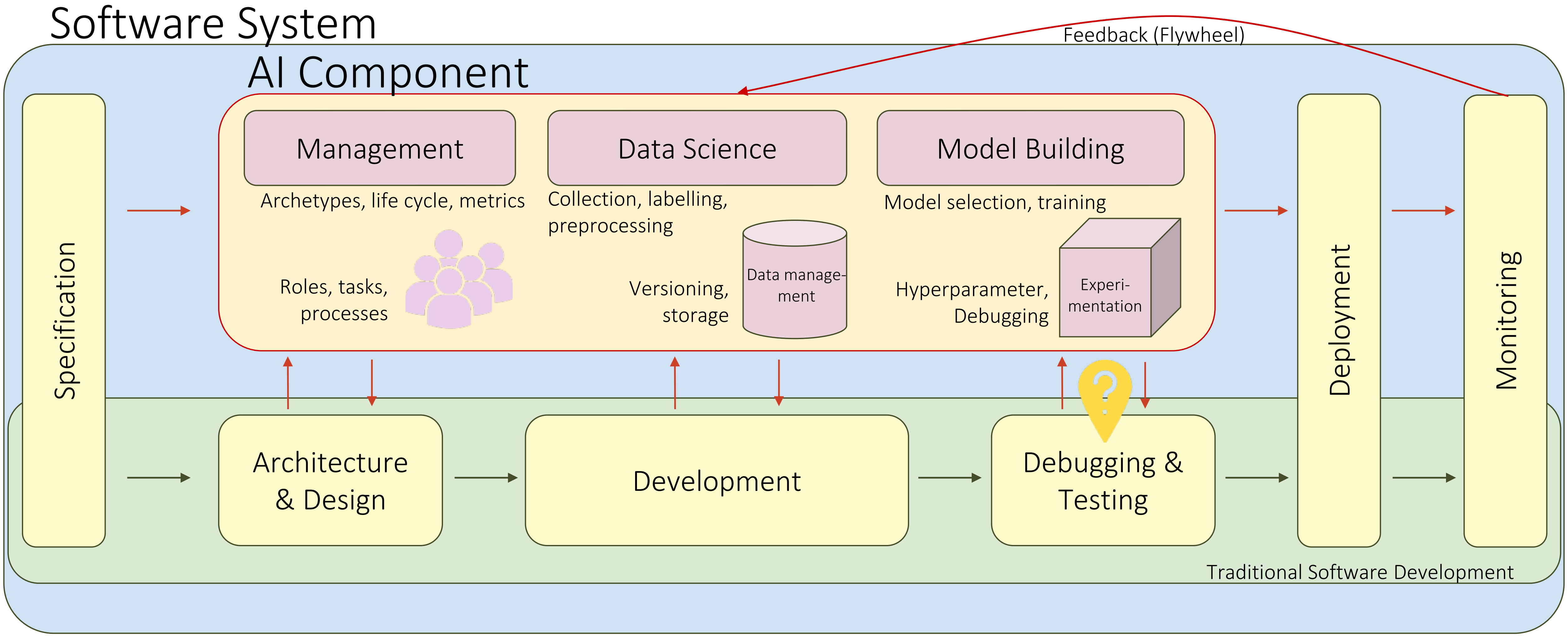
How to incorporate the AI component into the software system?

- How to manage information flow?
- How to bridge programming languages and system boundaries?
- How to design the system with modularity and software quality into account?
- SaaS / SaaFunction called by the system vs. embedded into the application vs. device deployments (embedded, mobile, server, etc.)



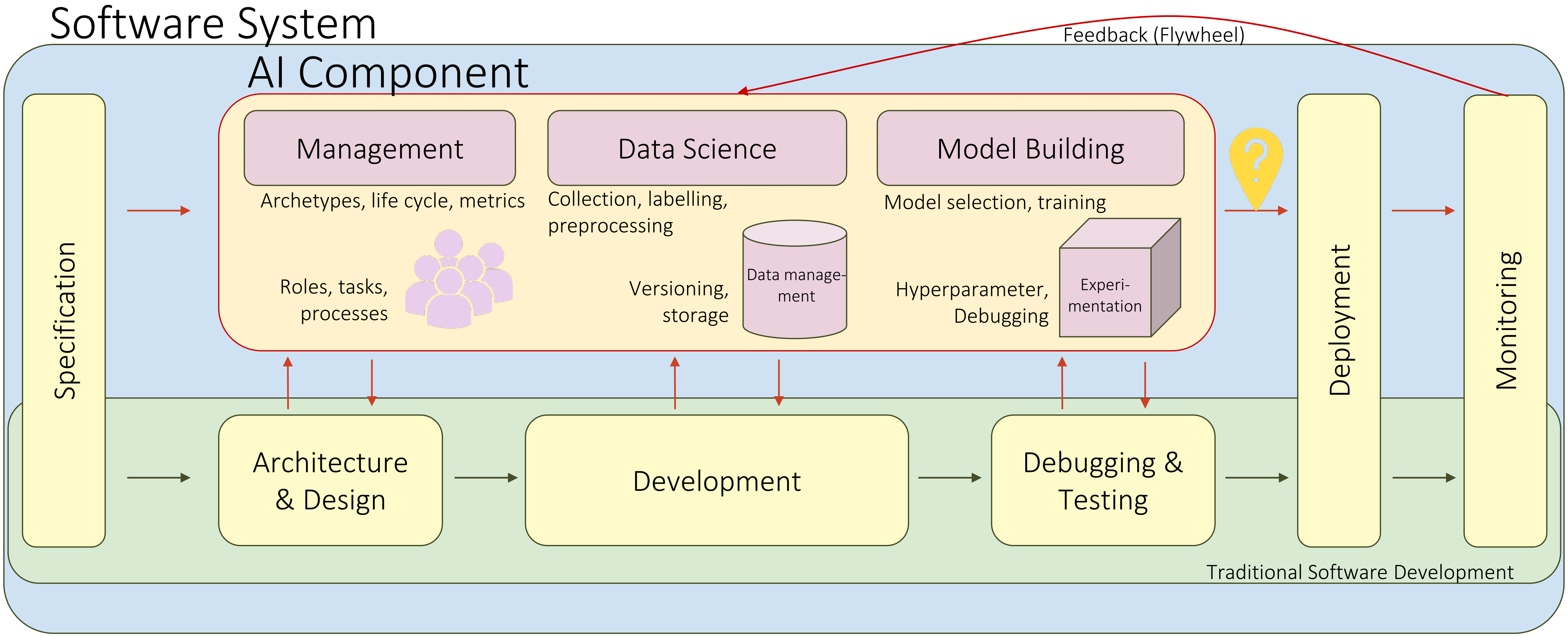
How to incorporate the development process of the AI component into the development process of the remaining software system?

- CI/CD for AI projects
- AIOps, MLOps, DataOps
- Technical debt of AI component



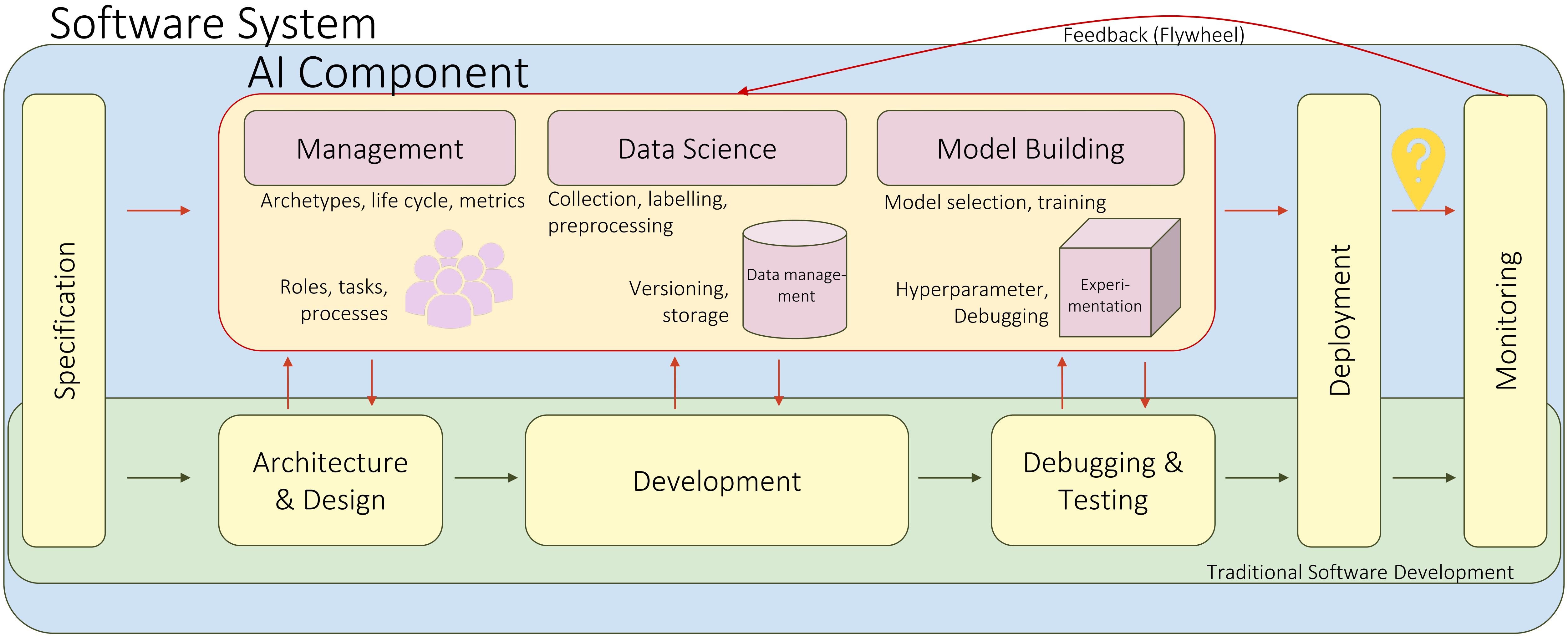
How to test an AI component? How to make conjunctive tests with the remaining software?

- What are appropriate metrics to test for?
- Testing your data vs. testing your ML code vs. testing your pipeline
- How to combine unit- / integration testing?



How to deploy AI models?

- Which tools and frameworks for to use for deployment?
- How to handle multiple versions of your model?
- How to test new versions?
- What target devices are possible for deployment and how do they differ?



Software System

AI Component

Feedback (Flywheel)

Specification

Management

Archetypes, life cycle, metrics

Roles, tasks, processes

Data Science

Collection, labelling, preprocessing

Versioning, storage

Data management

Model Building

Model selection, training

Hyperparameter, Debugging

Experimentation

Deployment

Monitoring

Architecture & Design

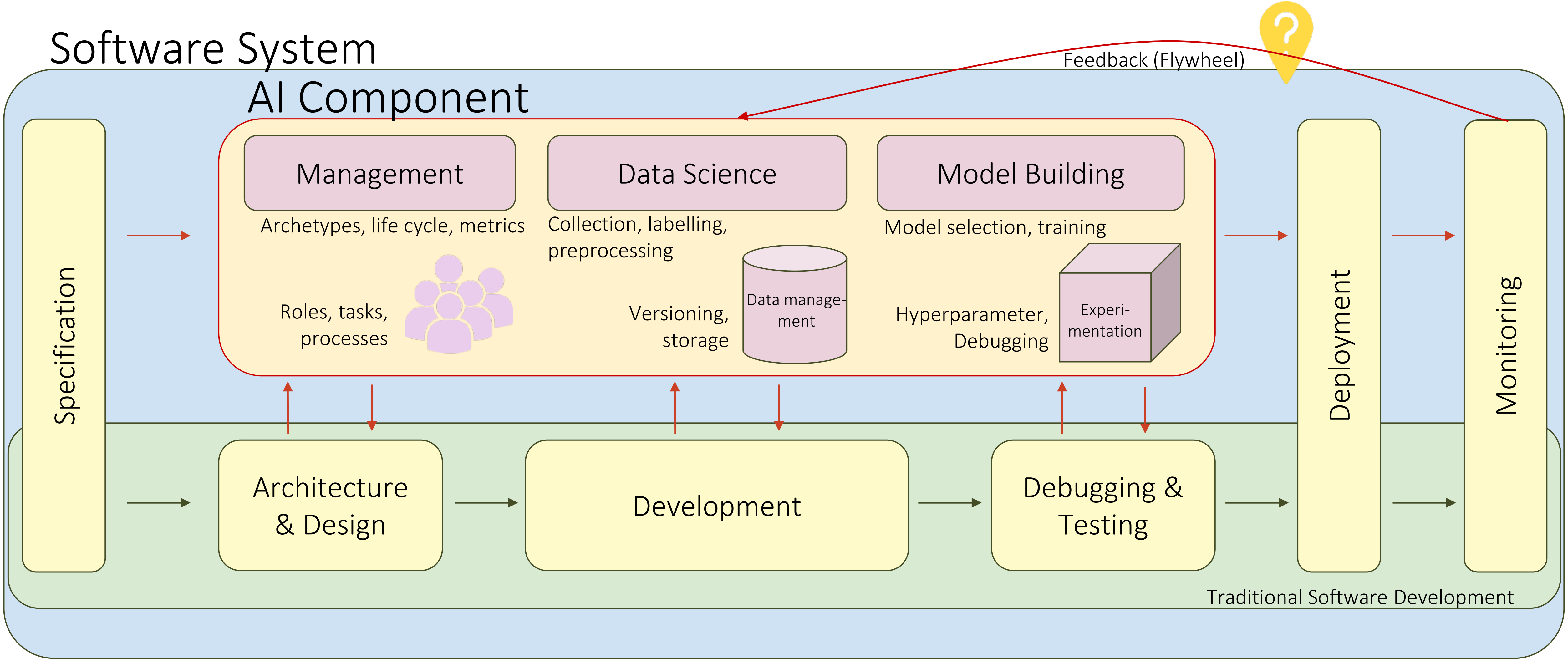
Development

Debugging & Testing

Traditional Software Development

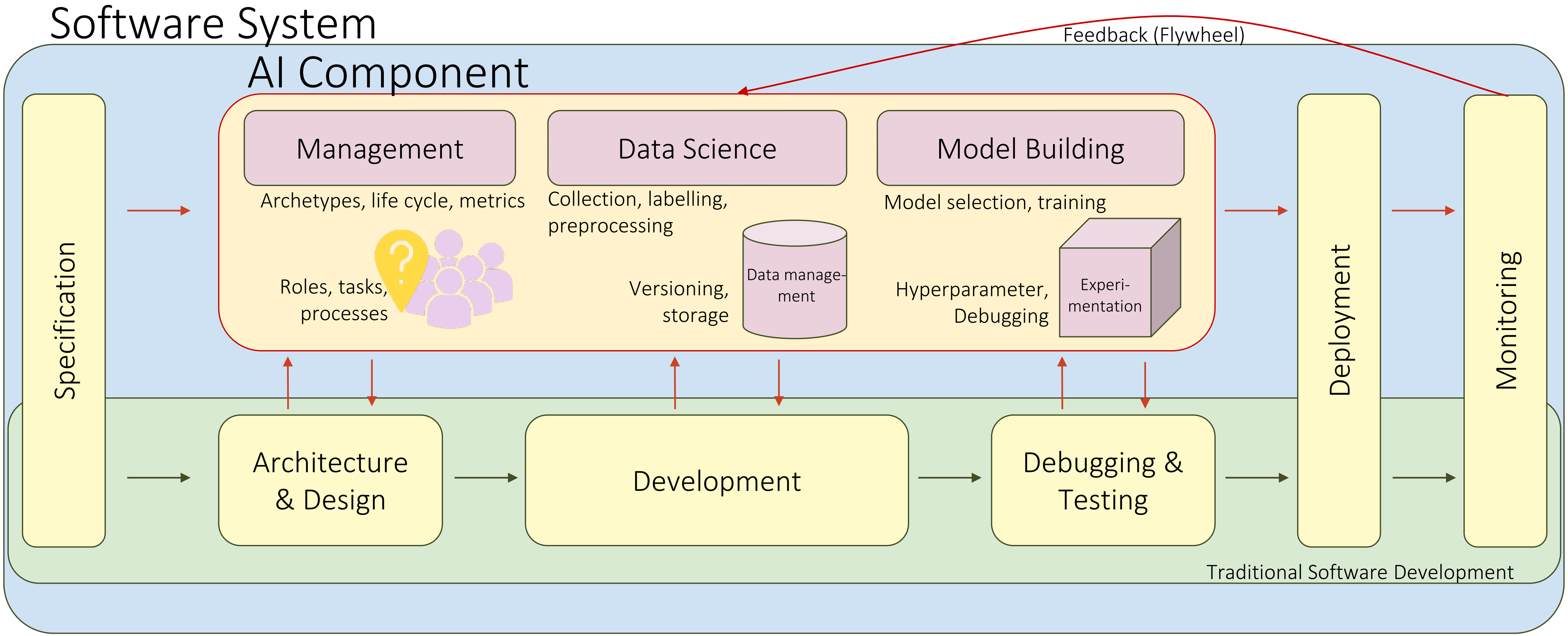
How to monitor AI decisions in practice?

- How to observe data distribution to detect data and model shifts?
- How to log the AI output?
- How to prevent or inform fatal AI errors?
- How to implement fall back measures?
- What frameworks and tools exist to incorporate monitoring?



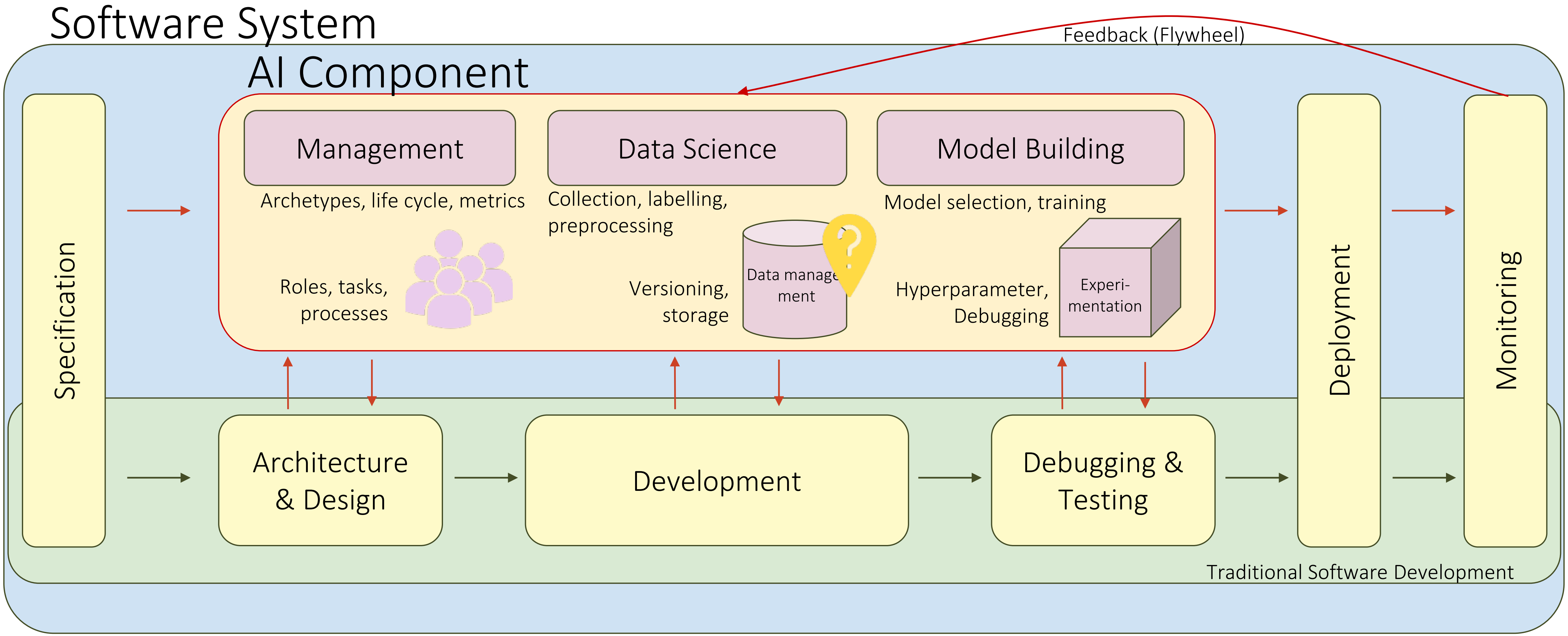
How to realize the feedback loop (Flywheel)?

- How to build labelling / feedback mechanisms into the application?
- What consequences does this have on UI/UX design?
- How to implement the feedback mechanism in an automated pipeline?
- How and when to trigger re-learning of the ML model?



Project management

- What AI project archetypes and ML model types exist to realize a business goal?
- How to start an AI/ML project based on requirements?
- How to select suitable metrics to optimize for?
- How to build teams and organize your company/project to productionize the AI/ML model?



Software System

AI Component

Feedback (Flywheel)

Specification

Management

Archetypes, life cycle, metrics

Roles, tasks, processes

Data Science

Collection, labelling, preprocessing

Versioning, storage

Data manage ment

Model Building

Model selection, training

Hyperparameter, Debugging

Experi-mentation

Deployment

Monitoring

Architecture & Design

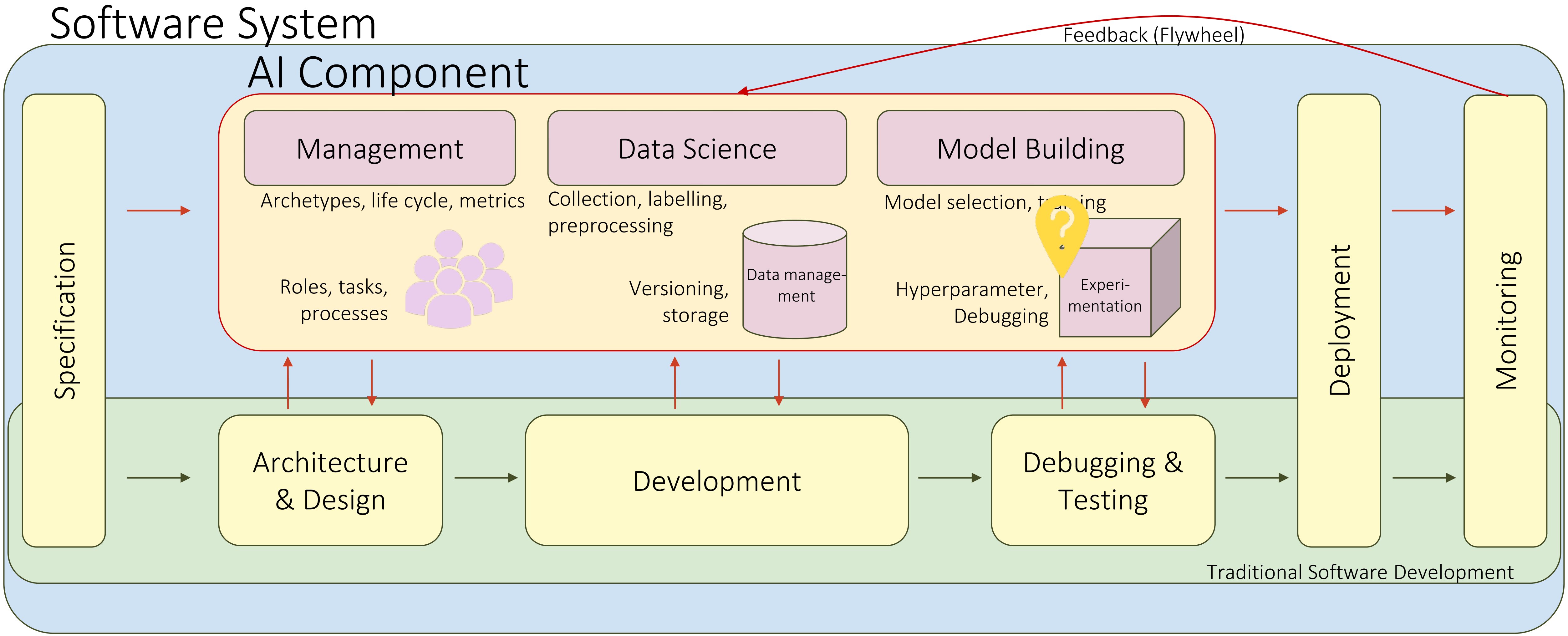
Development

Debugging & Testing

Traditional Software Development

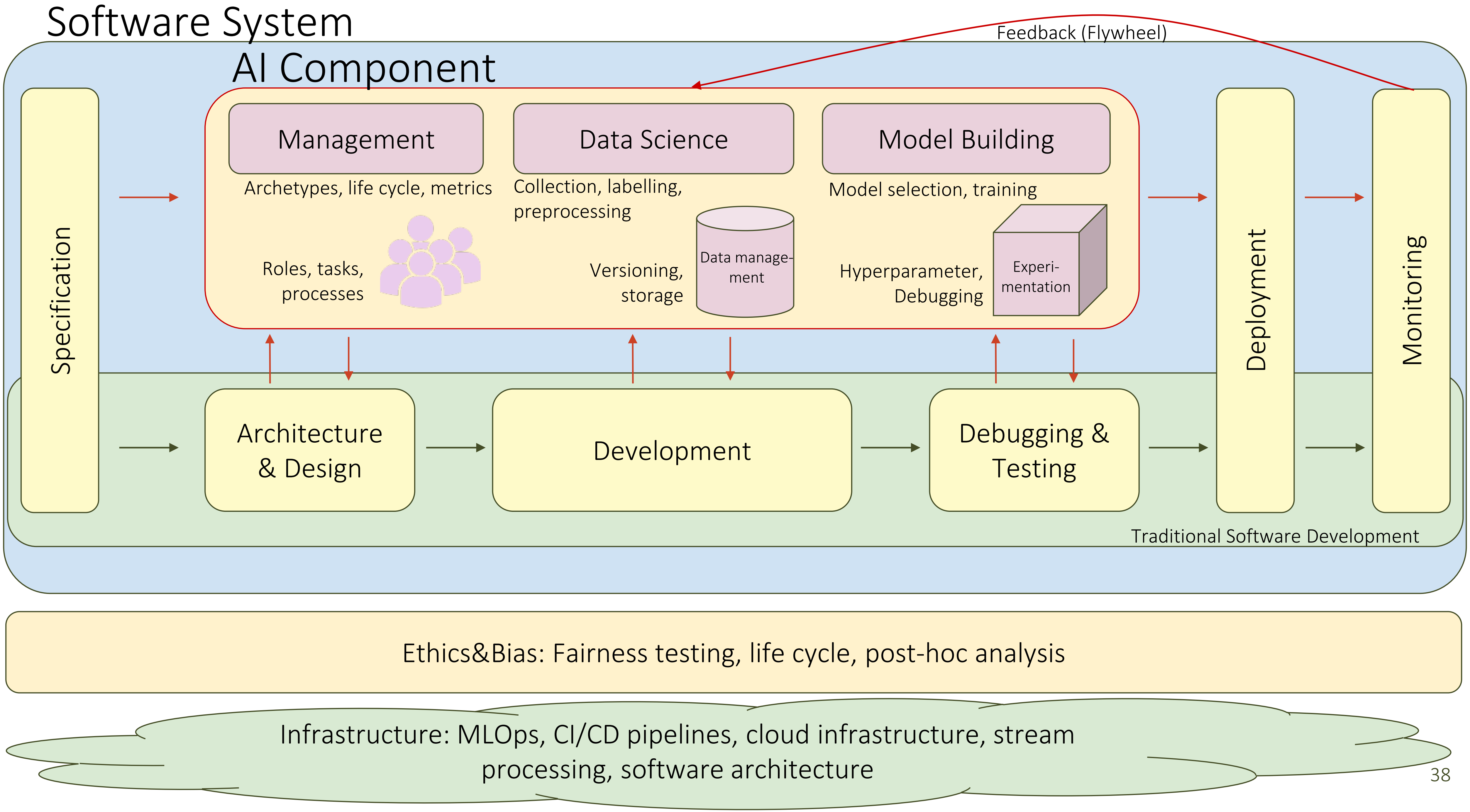
Data management

- How to store and version data?
- How to use different storage approaches?
- How to collect, label, and preprocess data?



Experimentation

- What tools and frameworks exist to automatically track and store experimentation runs?
- How to make experiments reproducible and how to automate the experimentation runs?
- How to ensure validity and perform hyperparmeter tuning of the ML model?



Software System

AI Component

Feedback (Flywheel)

Specification

Management

Archetypes, life cycle, metrics

Roles, tasks, processes

Data Science

Collection, labelling, preprocessing

Versioning, storage

Data management

Model Building

Model selection, training

Hyperparameter, Debugging

Experimentation

Deployment

Monitoring

Architecture & Design

Development

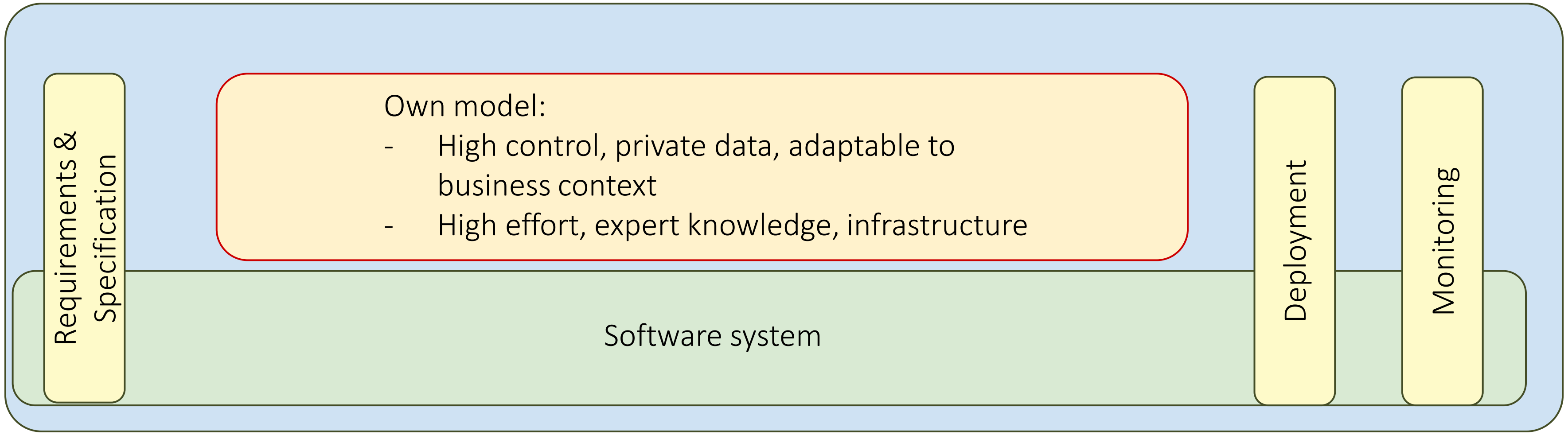
Debugging & Testing

Traditional Software Development

Ethics&Bias: Fairness testing, life cycle, post-hoc analysis

Infrastructure: MLOps, CI/CD pipelines, cloud infrastructure, stream processing, software architecture

Building everything



Requirements & Specification

Own model:

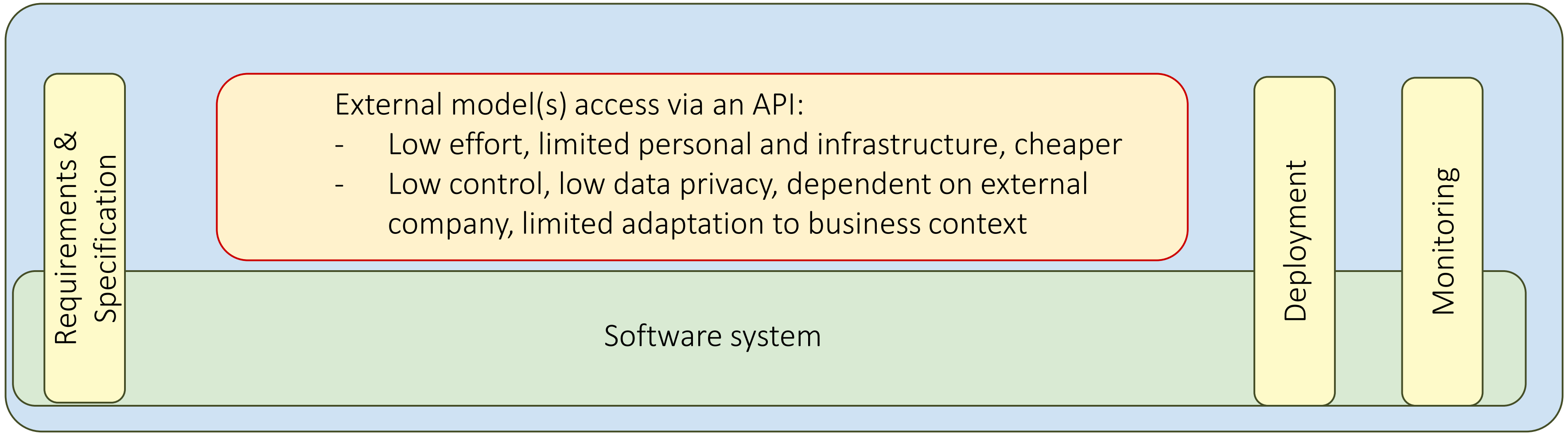
- High control, private data, adaptable to business context
- High effort, expert knowledge, infrastructure

Deployment

Monitoring

Software system

Building the software system



Requirements & Specification

External model(s) access via an API:

- Low effort, limited personal and infrastructure, cheaper
- Low control, low data privacy, dependent on external company, limited adaptation to business context

Deployment

Monitoring

Software system

Why this course?

TL;DR:

- ML modelling is just a tiny part of the process of bringing an AI to production
- Software engineering (SE) methods are a key success factor for AI/ML projects
- SE and ML projects differ, but need to be arranged to integrate both worlds
- Up to 80% of AI/ML projects fail, mostly because there is a lack of proper project management and planning, all known since decades in SE

Why Project Management?

Venturebeat reported: 87% of ML projects fail[*1]

Gartner reported: 53% of ML projects do not make it from prototype to production[*2]

> Gartner research shows only 53% of projects make it from artificial intelligence (AI) prototypes to production. CIOs and IT leaders find it hard to scale AI projects because they lack the tools to create and manage a production-grade AI pipeline. The road to AI production means turning to AI engineering, a discipline focused on the governance and life cycle management of a wide range of operationalized AI and decision models, such as machine learning or knowledge graphs.

Causes:

- Unclear project goal
- Misaligned ML and project goal
- Separate and uncoordinated work between data scientists and software engineers
- Unclear interfaces in terms of team responsibilities, documentation, and technical level
- Never deployed to production
- Technically infeasible

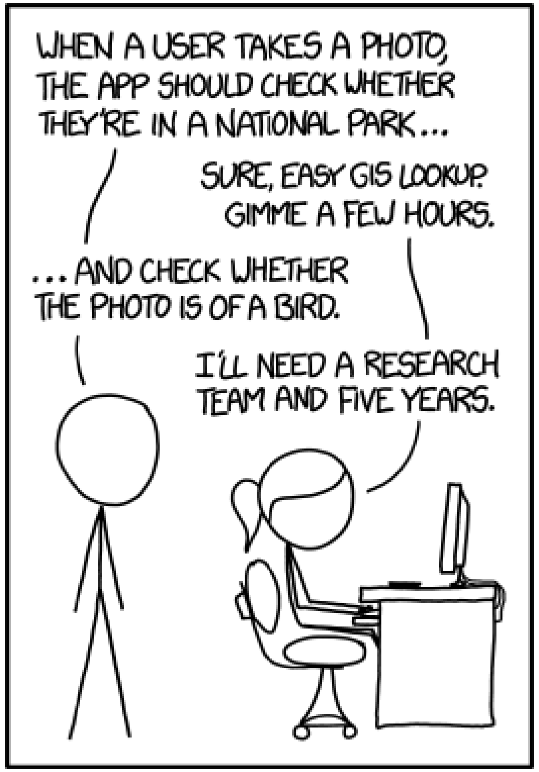
[*1]:https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production/

[*2]:https://www.gartner.de/de/newsroom/pressemitteilungen/2020-10-19-gartner-identifiziert-die-wichtigsten-strategischen-technologietrends-fuer-2021

Problem: Easy vs. Impossible Tasks

Similar looking things may be inherently different in complexity.

Project management may be a solution to reduce risk.

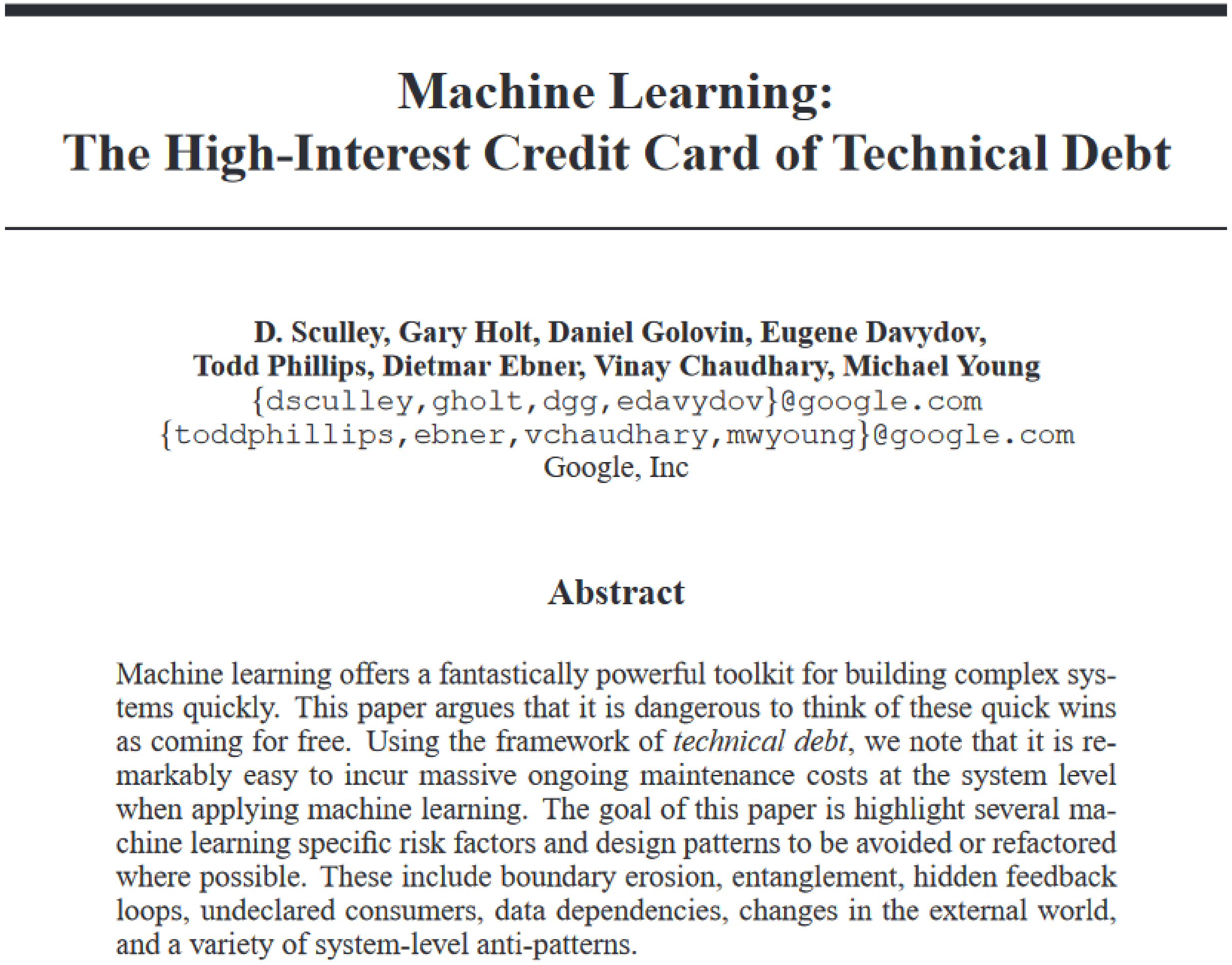

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Problem: Too much focus on ML Modelling

Limited awareness and education on processes and tasks of the whole production pipeline

Reports from Google show where the true costs for bringing an ML model to production lies, including maintenance and error handling



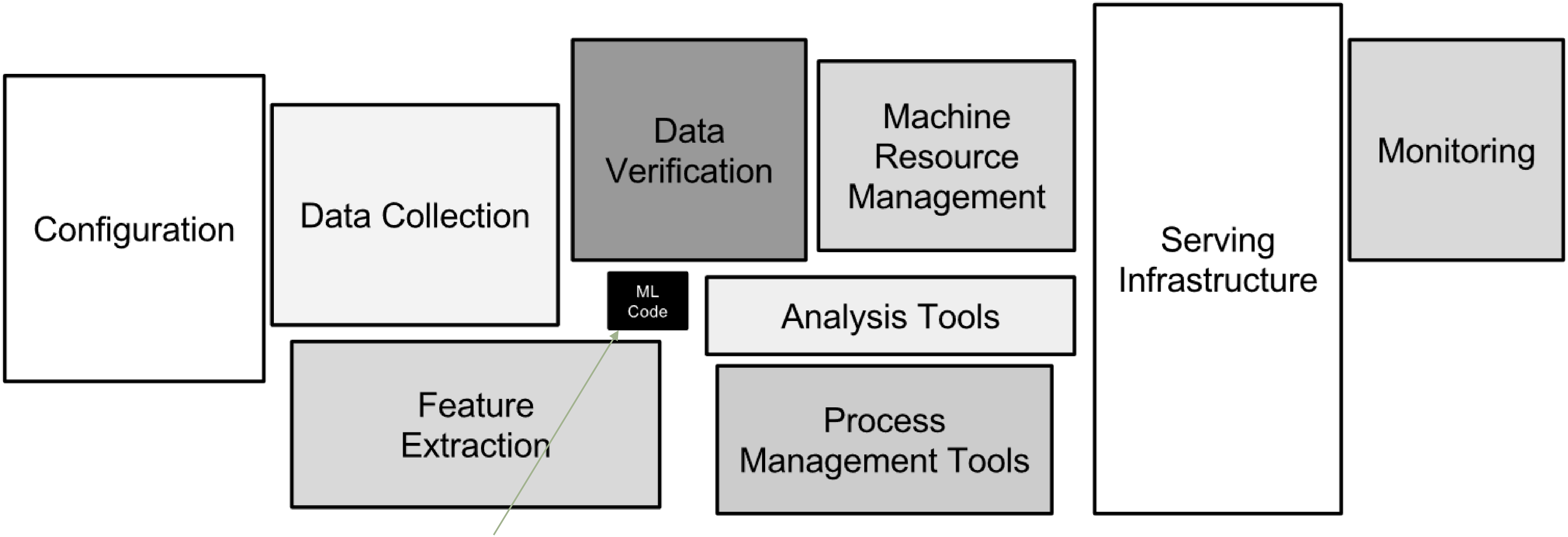
Machine Learning: The High-Interest Credit Card of Technical Debt

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov,
Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young
{dsculley,gholt,dgg,edavydov}@google.com
{toddphillips,ebner,vchaudhary,mwyoung}@google.com
Google, Inc

Abstract

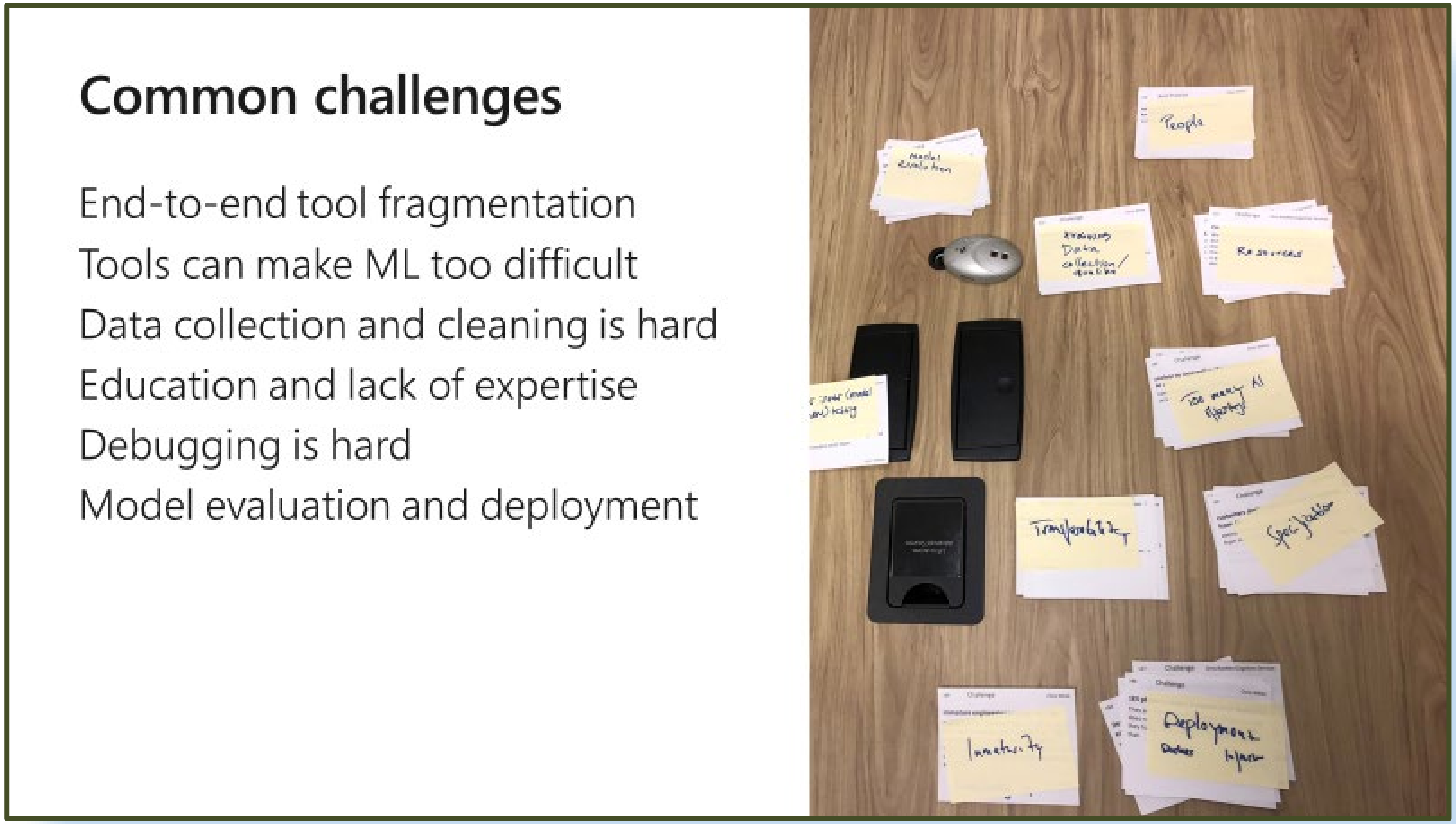
Machine learning offers a fantastically powerful toolkit for building complex systems quickly. This paper argues that it is dangerous to think of these quick wins as coming for free. Using the framework of *technical debt*, we note that it is remarkably easy to incur massive ongoing maintenance costs at the system level when applying machine learning. The goal of this paper is highlight several machine learning specific risk factors and design patterns to be avoided or refactored where possible. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, changes in the external world, and a variety of system-level anti-patterns.

Hidden Technical Debt



Education focuses here! But, why?

Challenges Reported by Microsoft Research



Common challenges

End-to-end tool fragmentation

Tools can make ML too difficult

Data collection and cleaning is hard

Education and lack of expertise

Debugging is hard

Model evaluation and deployment

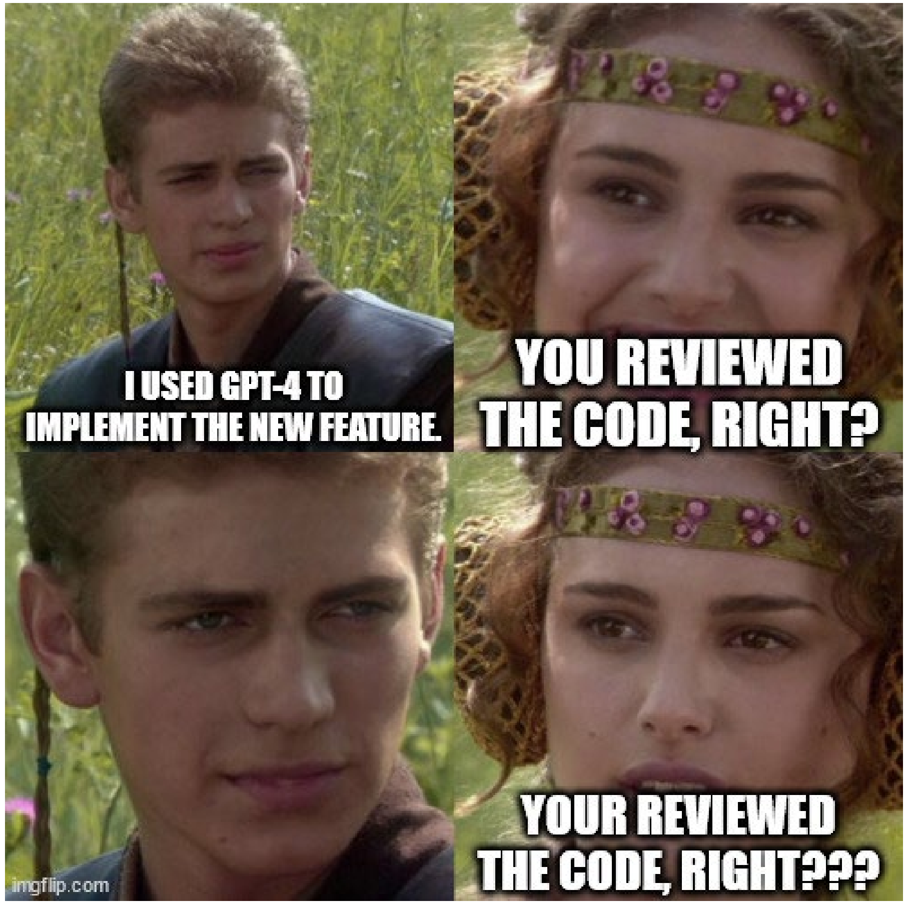
People
Model Evaluation
Data collection
Resources
Too many AI options
Transferability
Specification
Immaturity
Deployment

Software 2.0: A Software Engineering Perspective

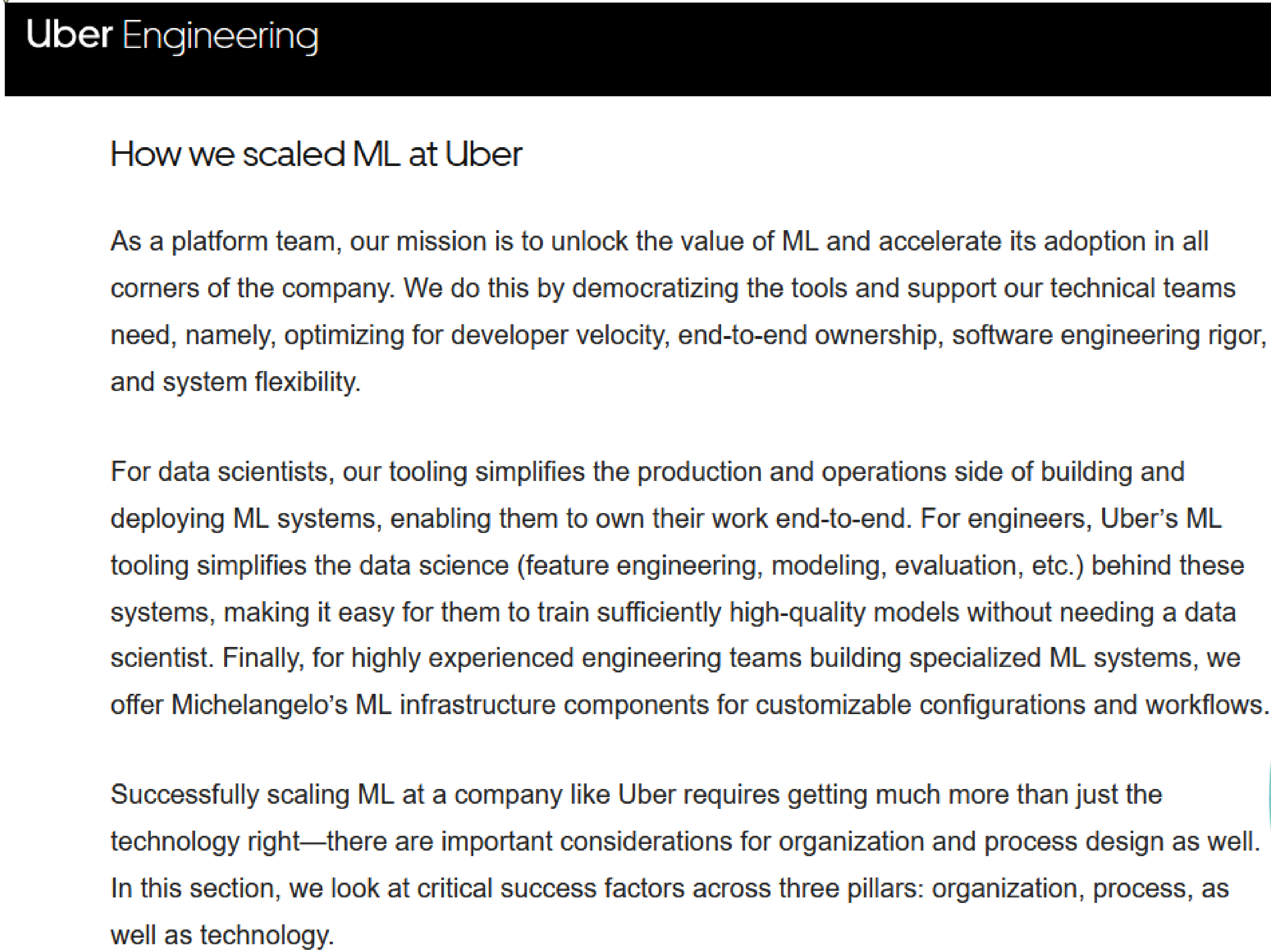
Karpathy Blog

ML perspective not aligning to 40 years of SE experience

What is with maintenance? What is with debugging? What is with code understanding? What is with specifications? -> best specification is code itself


I USED GPT-4 TO IMPLEMENT THE NEW FEATURE.
YOU REVIEWED THE CODE, RIGHT?
YOUR REVIEWED THE CODE, RIGHT???
imgflip.com

Solution



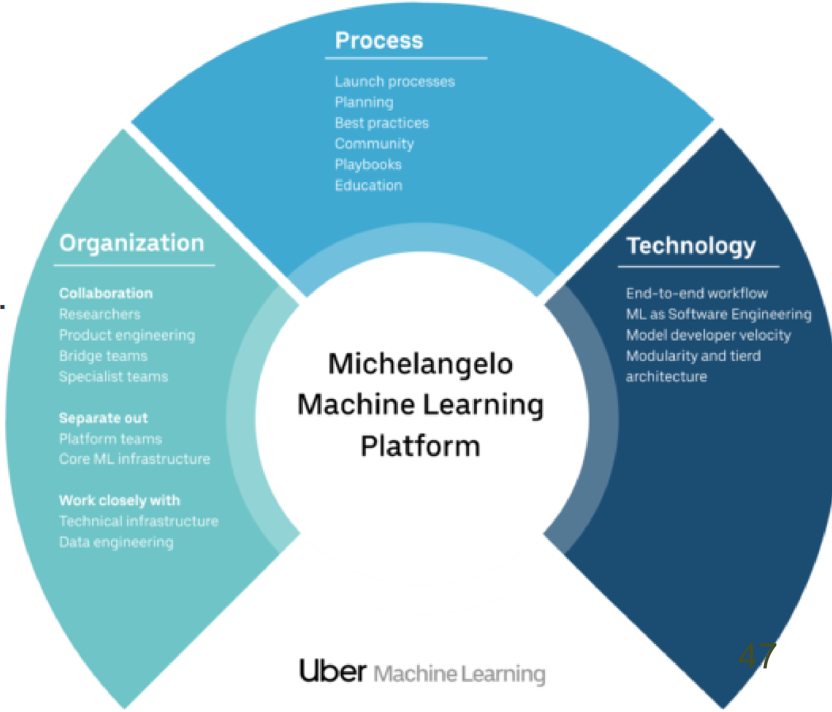
Uber Engineering

How we scaled ML at Uber

As a platform team, our mission is to unlock the value of ML and accelerate its adoption in all corners of the company. We do this by democratizing the tools and support our technical teams need, namely, optimizing for developer velocity, end-to-end ownership, software engineering rigor, and system flexibility.

For data scientists, our tooling simplifies the production and operations side of building and deploying ML systems, enabling them to own their work end-to-end. For engineers, Uber's ML tooling simplifies the data science (feature engineering, modeling, evaluation, etc.) behind these systems, making it easy for them to train sufficiently high-quality models without needing a data scientist. Finally, for highly experienced engineering teams building specialized ML systems, we offer Michelangelo's ML infrastructure components for customizable configurations and workflows.

Successfully scaling ML at a company like Uber requires getting much more than just the technology right—there are important considerations for organization and process design as well. In this section, we look at critical success factors across three pillars: organization, process, as well as technology.



Solution

Motivation and Topics

Uber Machine Learning

Scaling Machine Learning at Uber with Michelangelo

Cite from Jeremy Hermann and Mike Del Balso from Uber:

“

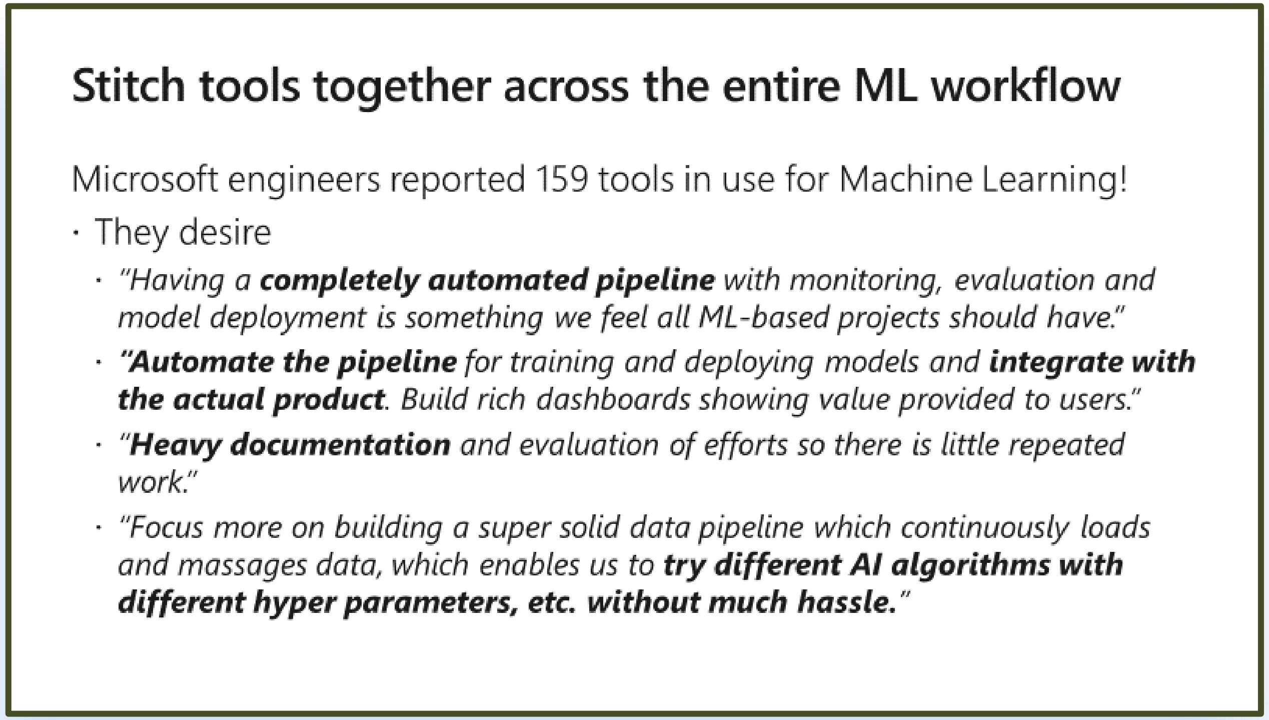
- ○ **End-to-end workflow**: ML is more than just training models; you need support for the whole ML workflow: manage data, train models, evaluate models, deploy models and make predictions, and monitor predictions.
- ○ **ML as software engineering**: We have found it valuable to draw analogies between ML development and software development, and then apply patterns from software development tools and methodologies back to our approach to ML.
- ○ **Model developer velocity:** Machine learning model development is a very iterative process—innovation and high-quality models come from lots and lots of experiments. Because of this, model developer velocity is critically important.
- ○ **Modularity and tiered architecture**: Providing end-to-end workflows is important for handling the most common ML use cases, but to address the less common and more specialized cases, it’s important to have primitive components that can be assembled in targeted ways.

”

https://eng.uber.com/scaling-michelangelo/

Solution

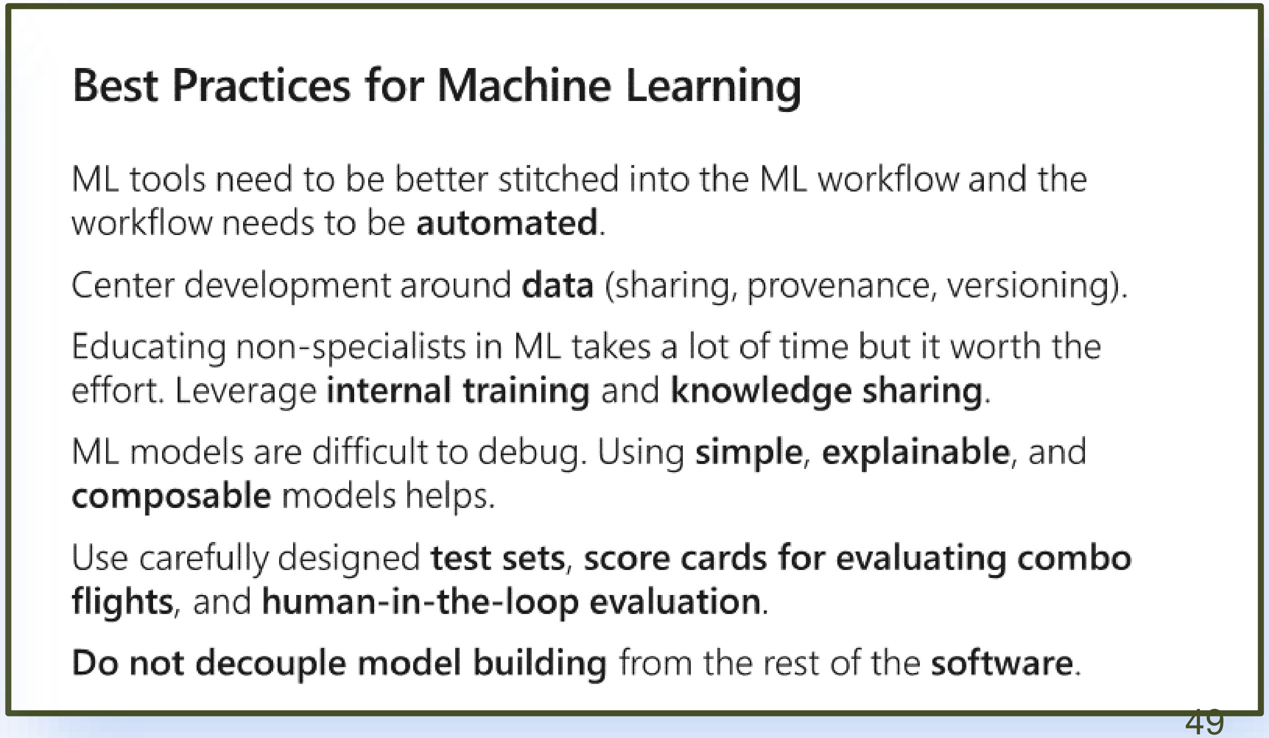
Microsoft Best Practices Motivate This Course



Stitch tools together across the entire ML workflow

Microsoft engineers reported 159 tools in use for Machine Learning!

- They desire
  - *"Having a **completely automated pipeline** with monitoring, evaluation and model deployment is something we feel all ML-based projects should have."*
  - ***"Automate the pipeline** for training and deploying models and **integrate with the actual product**. Build rich dashboards showing value provided to users."*
  - *"**Heavy documentation** and evaluation of efforts so there is little repeated work."*
  - *"Focus more on building a super solid data pipeline which continuously loads and massages data, which enables us to **try different AI algorithms with different hyper parameters, etc. without much hassle.**"*



Best Practices for Machine Learning

ML tools need to be better stitched into the ML workflow and the workflow needs to be **automated**.

Center development around **data** (sharing, provenance, versioning).

Educating non-specialists in ML takes a lot of time but it worth the effort. Leverage **internal training** and **knowledge sharing**.

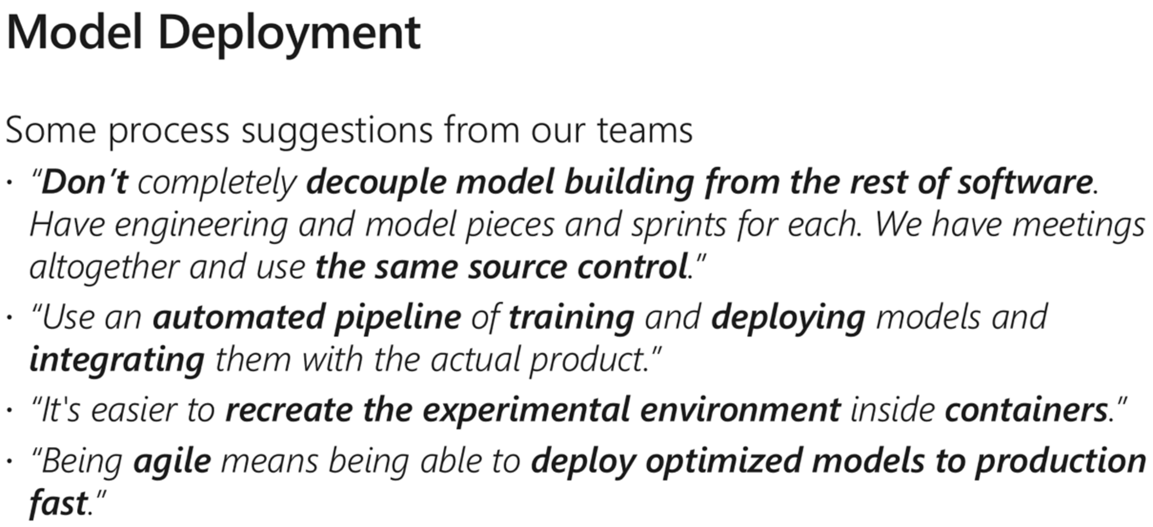
ML models are difficult to debug. Using **simple**, **explainable**, and **composable** models helps.

Use carefully designed **test sets**, **score cards for evaluating combo flights**, and **human-in-the-loop evaluation**.

Do not decouple model building from the rest of the **software**.

Solution

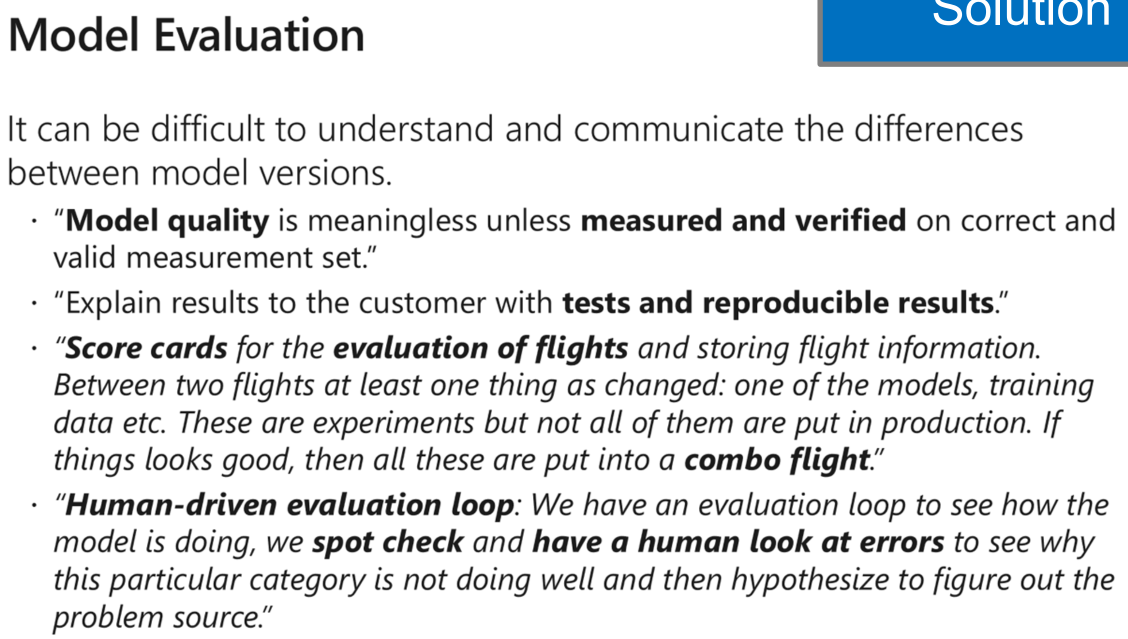
Further Best Practices



Model Deployment

Some process suggestions from our teams

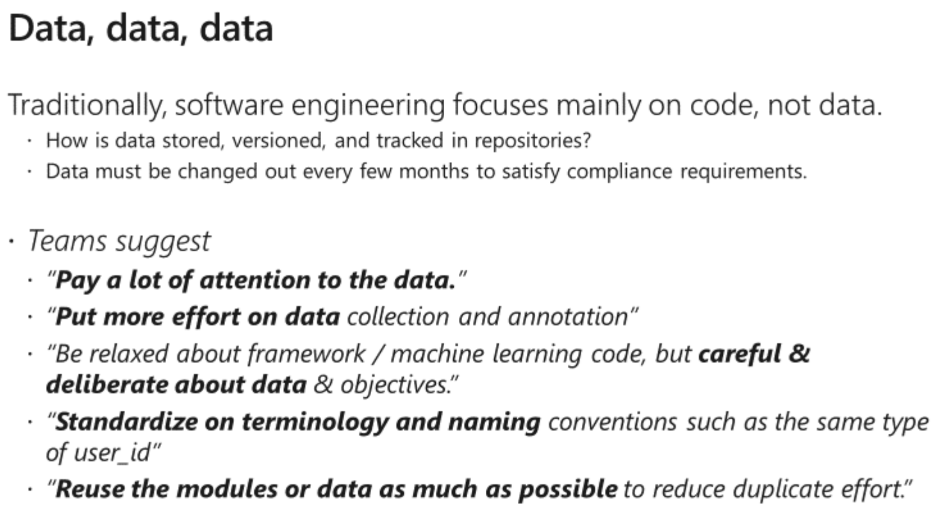
- *"**Don't** completely **decouple model building from the rest of software**. Have engineering and model pieces and sprints for each. We have meetings altogether and use **the same source control**."*
- *"Use an **automated pipeline** of **training** and **deploying** models and **integrating** them with the actual product."*
- *"It's easier to **recreate the experimental environment** inside **containers**."*
- *"Being **agile** means being able to **deploy optimized models to production fast**."*



Model Evaluation

It can be difficult to understand and communicate the differences between model versions.

- "**Model quality** is meaningless unless **measured and verified** on correct and valid measurement set."
- "Explain results to the customer with **tests and reproducible results**."
- *"**Score cards** for the **evaluation of flights** and storing flight information. Between two flights at least one thing as changed: one of the models, training data etc. These are experiments but not all of them are put in production. If things looks good, then all these are put into a **combo flight**."*
- *"**Human-driven evaluation loop**: We have an evaluation loop to see how the model is doing, we **spot check** and **have a human look at errors** to see why this particular category is not doing well and then hypothesize to figure out the problem source."*

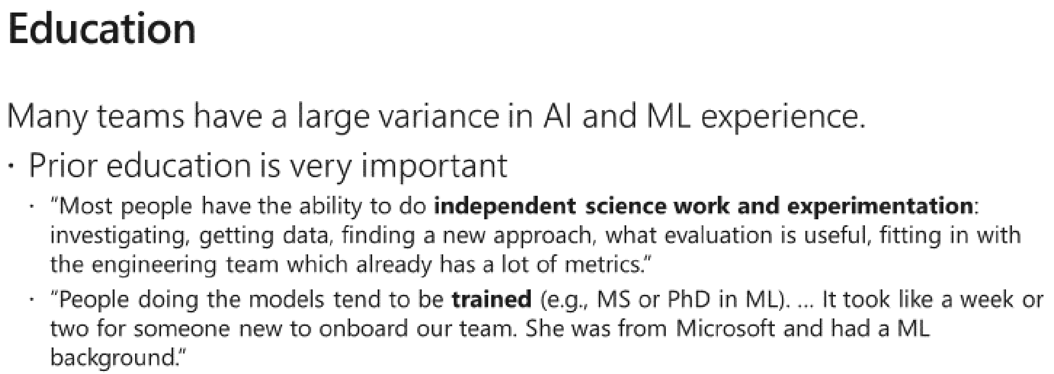


Data, data, data

Traditionally, software engineering focuses mainly on code, not data.

- How is data stored, versioned, and tracked in repositories?
- Data must be changed out every few months to satisfy compliance requirements.

- *Teams suggest*
  - *"**Pay a lot of attention to the data.**"*
  - *"**Put more effort on data** collection and annotation"*
  - *"Be relaxed about framework / machine learning code, but **careful & deliberate about data** & objectives."*
  - *"**Standardize on terminology and naming** conventions such as the same type of user_id"*
  - *"**Reuse the modules or data as much as possible** to reduce duplicate effort."*



Education

Many teams have a large variance in AI and ML experience.

- Prior education is very important
  - "Most people have the ability to do **independent science work and experimentation**: investigating, getting data, finding a new approach, what evaluation is useful, fitting in with the engineering team which already has a lot of metrics."
  - "People doing the models tend to be **trained** (e.g., MS or PhD in ML). … It took like a week or two for someone new to onboard our team. She was from Microsoft and had a ML background."

Questions

How hard is it to take out a feature from production?

- Example: Spam filter based on IP blocks now entire subnets of the company because spam emails are bouncing back

How much time does a prediction cost in production?

- Depends on HW, model, feature computation, data allocation, etc.