# DEVOPS FOR AI

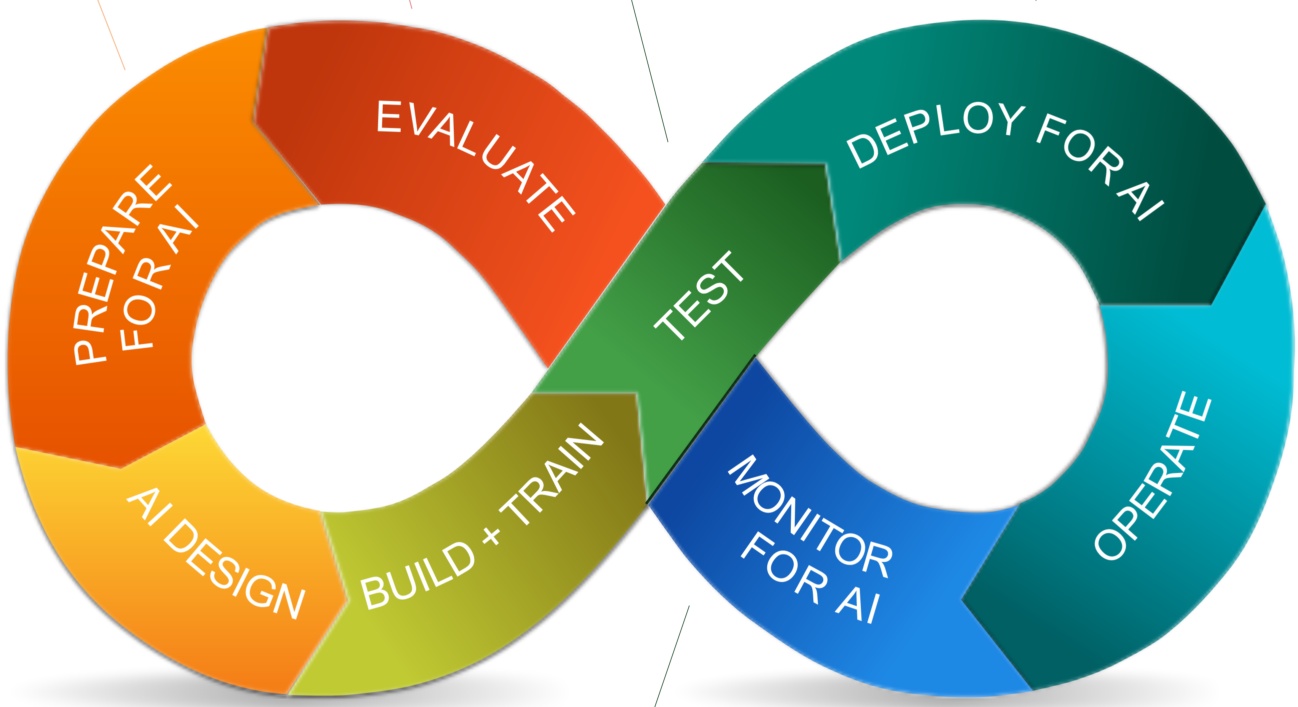
A framework for Artificial Intelligence & Machine Learning Solutions

### 8. Evaluate
- Validate business case
- Improve solution
- Improve process

### 5. Deploy for AI
- Model Management
- Deployment Pipeline for AI
- AI Data Storage

### 1. Prepare
- Idea generation
- Business case
- Validate idea

### 4. Test
- Model testing
- Wizard of Oz
- Prepare feedback loop




### 2. AI Design
- Architecture for AI
- AI/ML Model choice
- Specs for AI

### 7. Monitor for AI
- AI Metrics
- Model visualisation
- Business metrics

### 3. Build + Train
- AI/ML Model
- Data pre-processing
- Model tuning

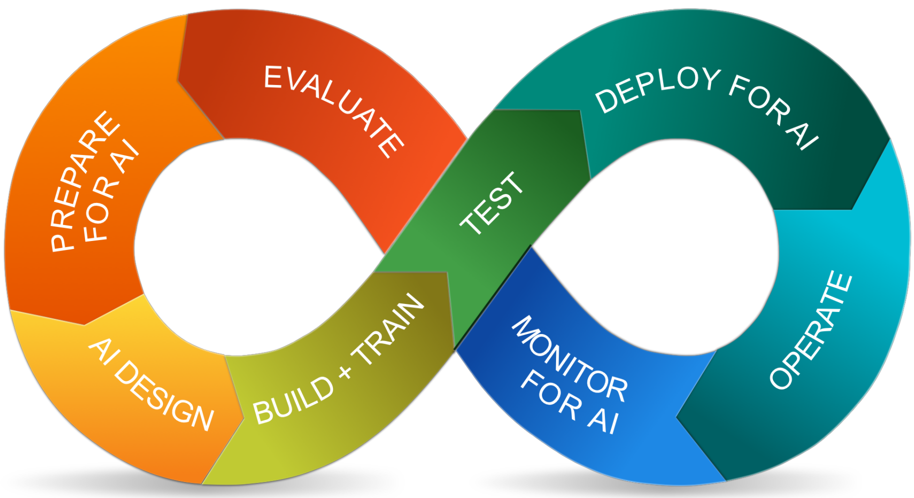
### 6. Operate
- Human in the loop
- Scaling & Performance
- Defect Analysis for AI

infoSupport
Solid Innovator

For more info: www.infosupport.com/ai

# DEVOPS FOR AI - AN INTRODUCTION

There isn’t an unique definition of DevOps, but Wikipedia defines it as this: *DevOps (a clipped compound of "development" and "operations") is a software development methodology that combines software development (Dev) with information technology operations (Ops). The goal of DevOps is to shorten the systems development life cycle while also delivering features, fixes, and updates frequently in close alignment with business objectives. The DevOps approach is to include automation and event monitoring at all steps of the software build.*




We can use this methodology for AI applications as well. Many activities needed to create an AI application involves traditional Software Engineering and these activities remain the same. In DevOps for AI we focus on the main activities that differ from the traditional Software Engineering DevOps.

This guide shows the DevOps cycle for AI applications, but does not describe in any way of form how you have to perform a certain activity. It always depends on the context of your product or project. It does provide a high level overview of activities to perform DevOps for AI. With small articles we show examples and practical cases. As a reader you can judge if the example is usable and applicable for your project or product. If not we hope it gives inspiration to find a good way that works within your context.

The DevOps for AI cycle consists of the following phases:

1. Prepare for AI
2. AI Design
3. Build + Train
4. Test
5. Deploy for AI
6. Operate
7. Monitor for AI
8. Evaluate

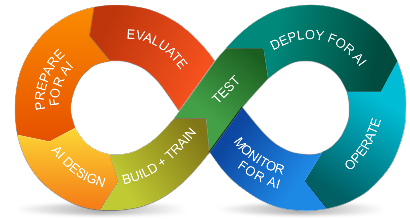
Each phase has a detailed model with the most important activities. The models are not designed to be complete and comprehensive. Any valuable additions are welcomed.

Happy reading and let's create some awesome AI.

# 1. PREPARE FOR AI

"Success is where preparation
and opportunity meet."

Bobby Unser





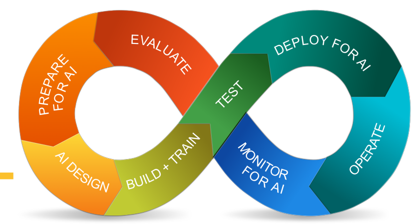
During 'Prepare for AI' we try to validate our idea as soon as possible. Is it possible to make significant impact on the business goals. To start validating we probably need some data and an idea about an initial AI/ML model. The proposed solution must be feasible within the boundaries of our company's data policy.

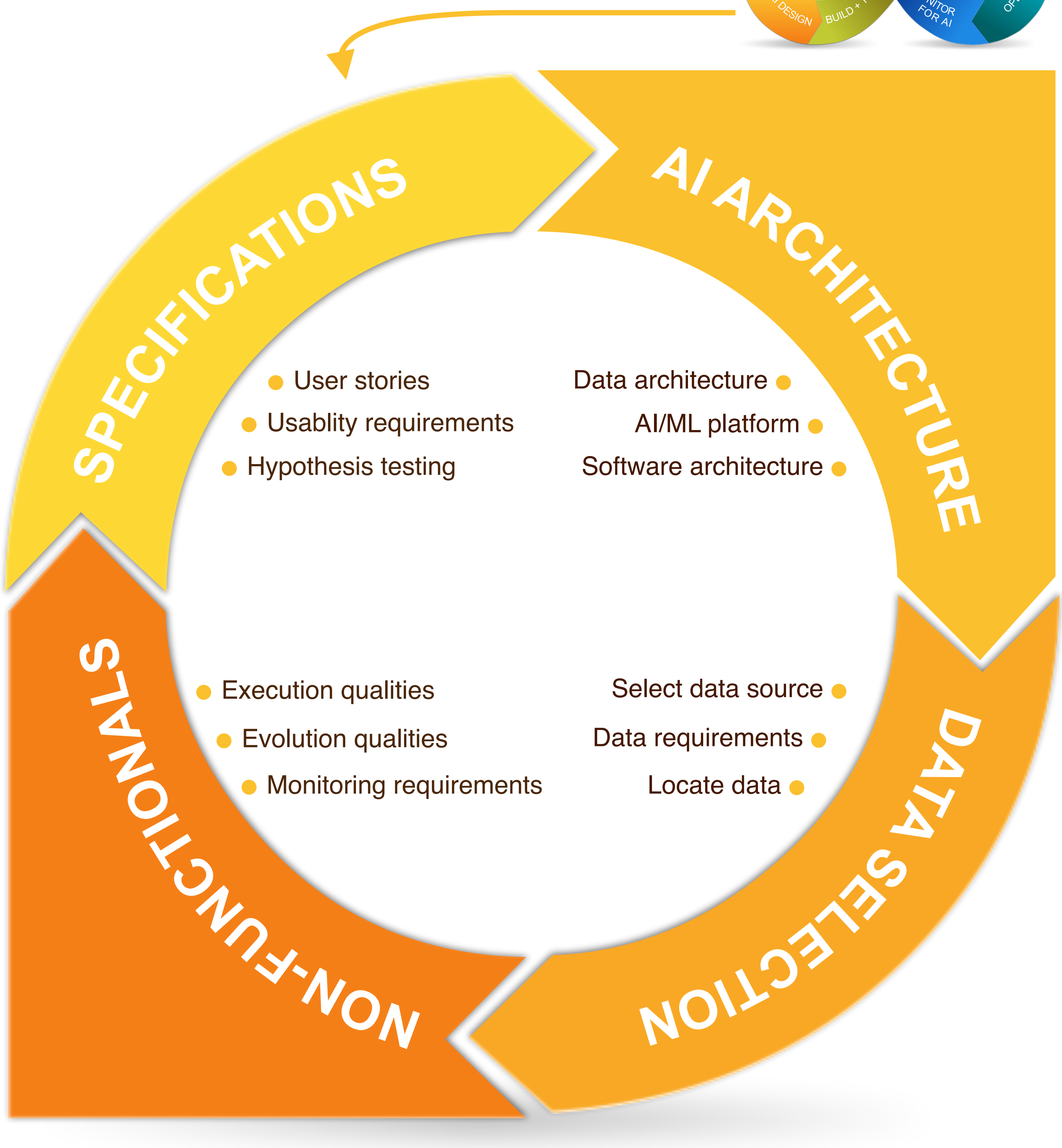
infoSupport
Solid Innovator
For more info: www.infosupport.com/ai

# 2. AI DESIGN

“Success is where preparation
and opportunity meet.”

Bobby Unser



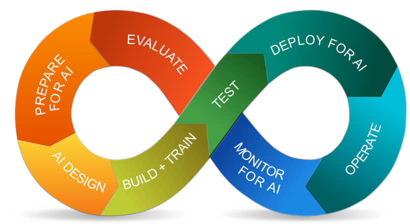



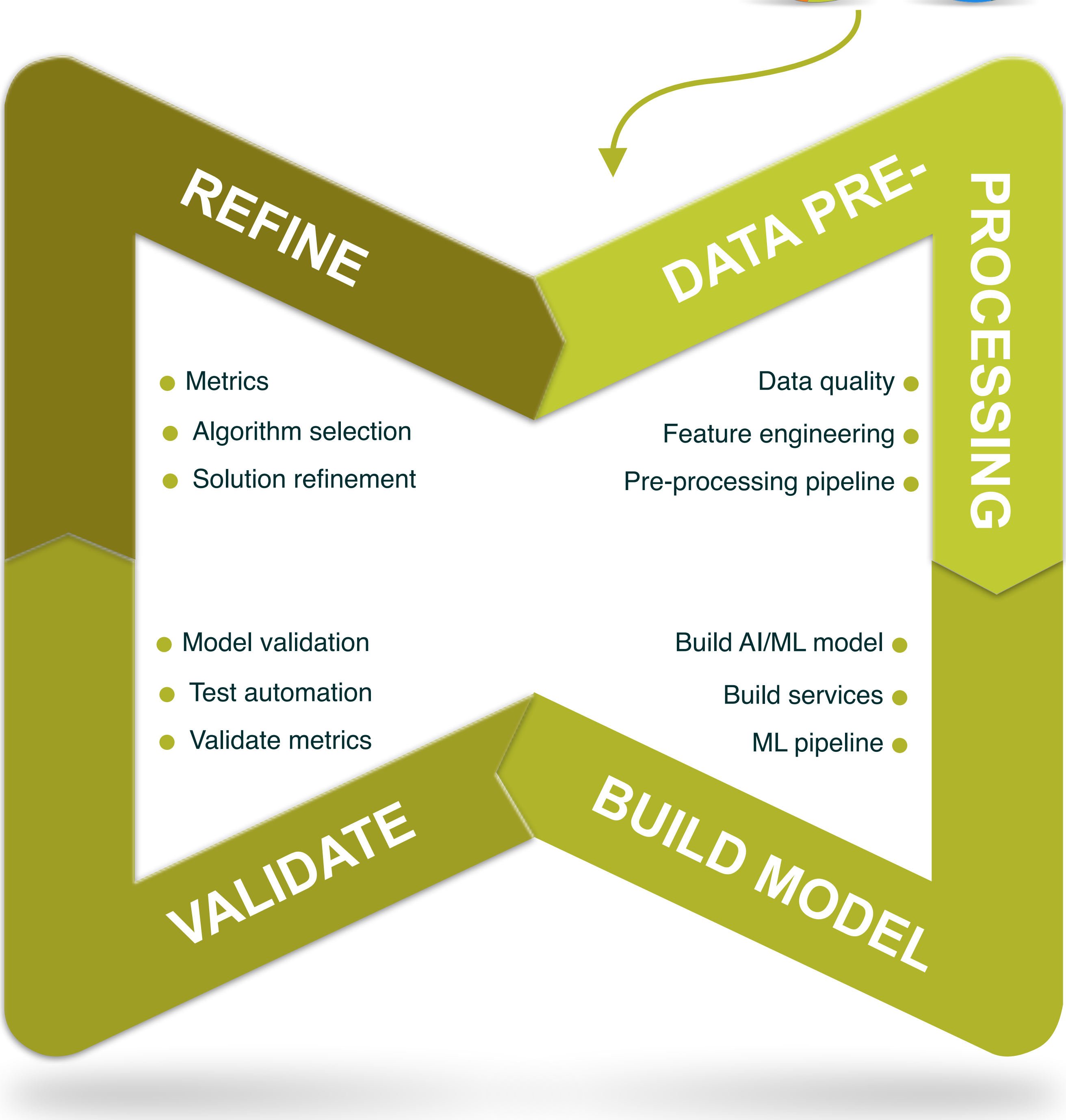

In the phase ‘AI Design’ we draw the first contours for our solutions. We give answers to the questions like which architectures are feasible, which data is needed and where is the data located? The functional and non-functional requirements sets a framework for the solution.

# 3. BUILD + TRAIN

“Those who can image anything, can create the impossible”

Alan Turing



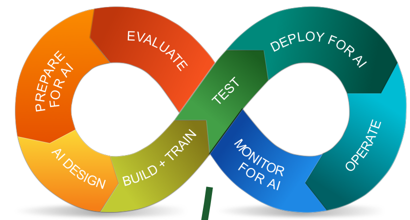



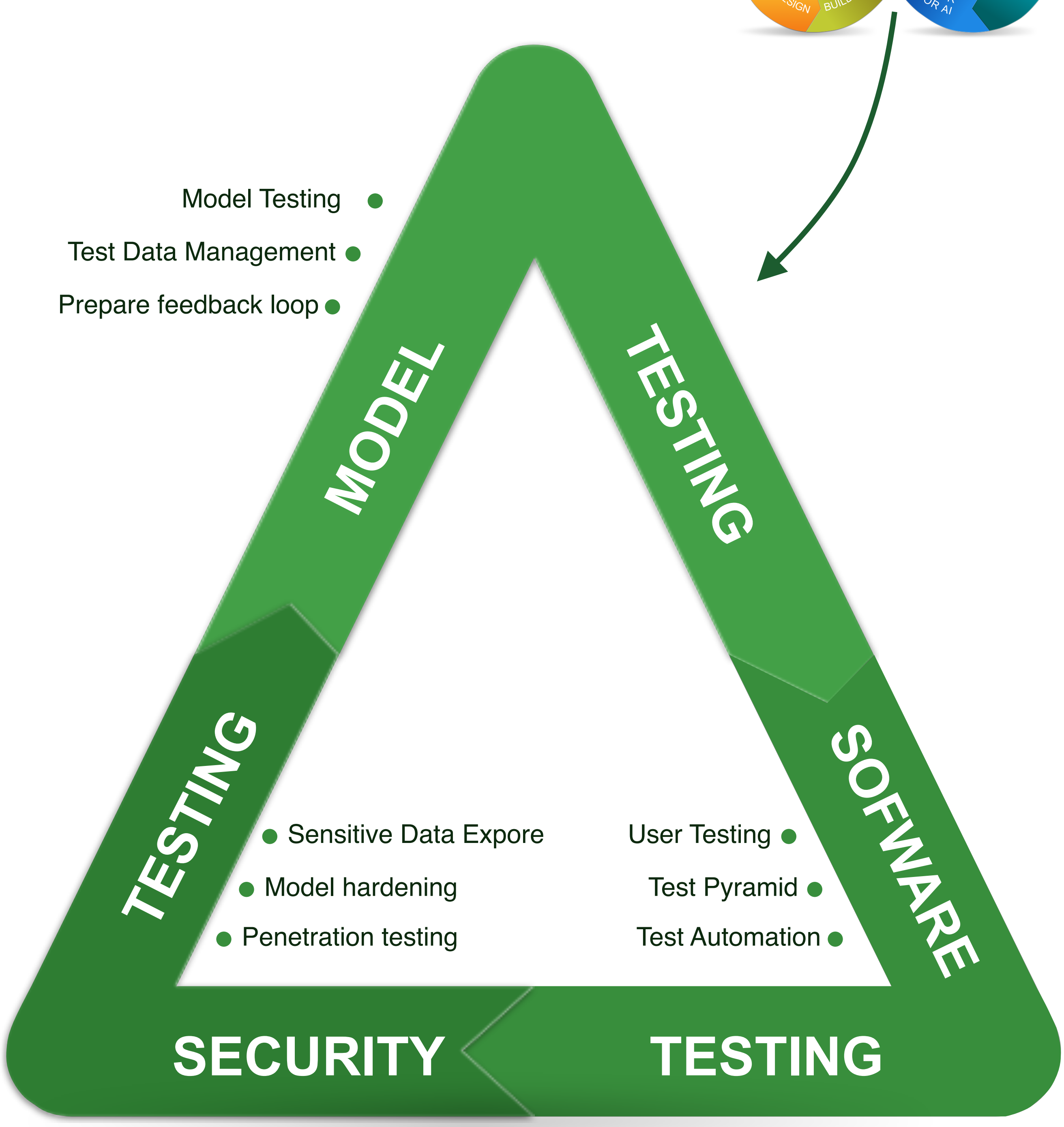

During ‘Build + Train’ we create our models and the software to serve the models and choose the metrics to optimize our models on. For continuous delivery purposes we instantiate data pre-processing and machine learning pipelines next to our regular build.

# 4. TEST

“The bitterness of poor quality remains long after the sweetness of low price is forgotten.”

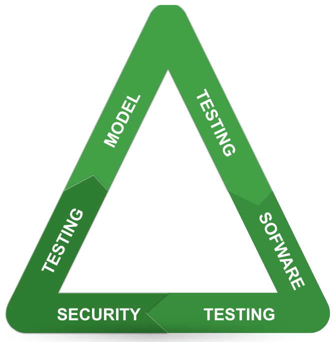
Benjamin Franklin




During ‘Test’ we test our models for production and make sure the whole solution works as aspected. This means we perform traditional DevOps software testing with extra focus on security testing. We don’t want to expose senstive data and protect our AI/ML models against abusive use.

# CASE STUDY: USER TESTING AI MODEL FOR ABN AMRO BANK




Most of the times when we think of testing our AI or Machine Learning model it's about accuracy or maybe even precision and recall. We presume that these measures tell us the correctness of our product. But the correctness of the model is not the same as a satisfied user. Let me tell you a short story of how we created a great model for tagging knowledge which resulted in poor results after the first user tests. And by the way, this story has a happy end.

## Smart product

We developed a smart product full of artificial intelligence and machine learning for ABN AMRO, a large bank in The Netherlands. Bank employees can create Standard Operational Procedures (SOPs) and share them with co-workers. Co-workers can search for the right SOPs, but the system recommends the right SOPs at the right moment as well.

## Better, more tags and faster

One of the key elements for our model is the use of tags. A tag is a keyword or term assigned to a piece of information. This kind of metadata helps describe an item. The search, recommendation and profiling algorithms are leaning heavily on these tags. Long story short, they're important for the system, so it's important the users add the right tags to the information.

*After some test rounds we found the perfect solution for this problem.*

To improve the processing of tagging, we created a tag recommendation model. With the help machine learning algorithms and with Natural Language Processing we extract all possible tags out of the description and the procedure. The recommendation model classifies the tags as ‘relevant’ or ‘not relevant'. This has to result in better tags, more tags and faster tagging.

## User testing

We were as confident as we could be. This was a neat model. All the measures were looking good. Precision high, pretty good recall. The user interface was slick and during typing the system recommended tag after tag after tag. Let the user testing begin. Nothing can go wrong, right?

Wrong! Our usability testing expert came back with the first results and they were disappointing. Yes, we collected more tags. Yes, the tags were more relevant, so better. Yes, the users liked the idea of tag recommendations. But no, it wasn't faster. Even worse it took way more time for each user in the test to complete the tagging.

## Problem and solution

The problem was that we showed all relevant tags and sometimes this could add up to over 10 or 15 tags. The

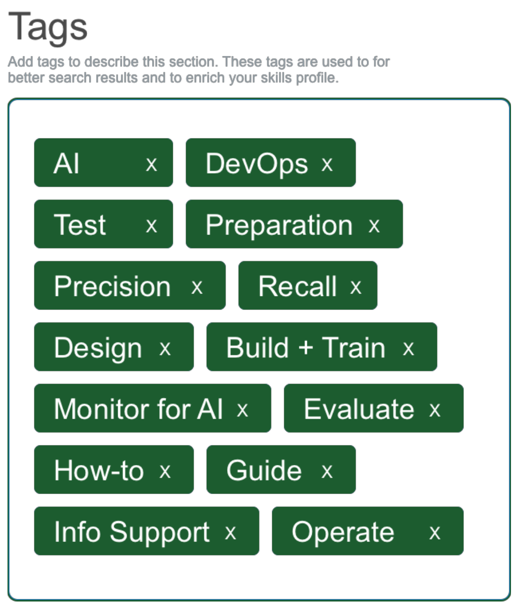



*Figure 1: First UI design*

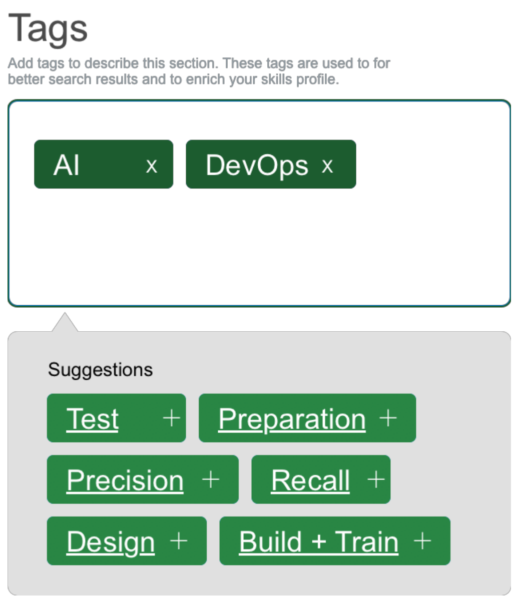



*Figure 2: Final UI design*

*“Our usability testing expert came back with the first results and they were disappointing”*

users read all the tags and had to think for each tag if it was the right tag. So it slowed them down.

After some test rounds we found the perfect solution for this problem. We split the selected tags and the suggested tags and showed less of the tags. It took some testing to find the sweet spot for the number of tags to show. It wasn't a game of just picking a number. The sweet spot is different for each piece of information. After round 4 we came up with the solution in Figure 2 and now we have better tags, more tags and the tagging is faster and easier than ever.

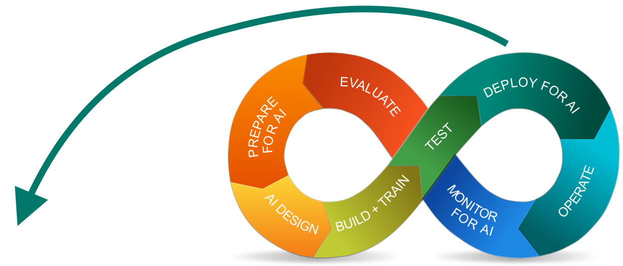
## Conclusion

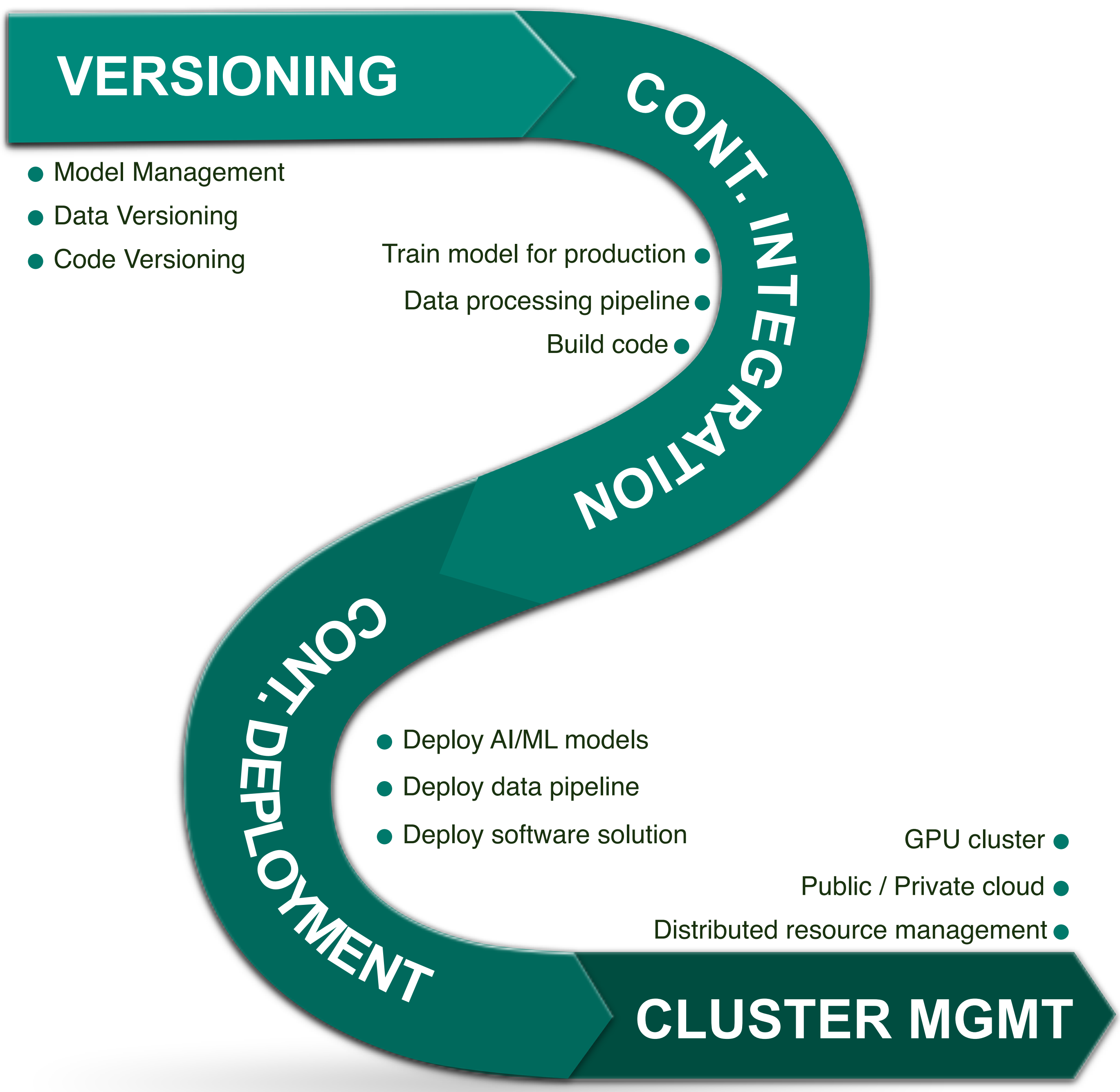
So although the recommendation model performed well, it didn't satisfy the user. We had to adjust the model to create the best solution for the user. We often think we know what our model does, but we never know how users react to it, unless we test it.

# 5. DEPLOYMENT FOR AI

“There are really three parts to the creative process. First there is inspiration, then there is the execution, and finally there is the release.”

Eddie van Halen



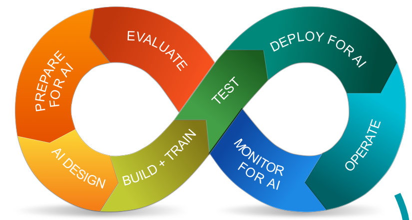



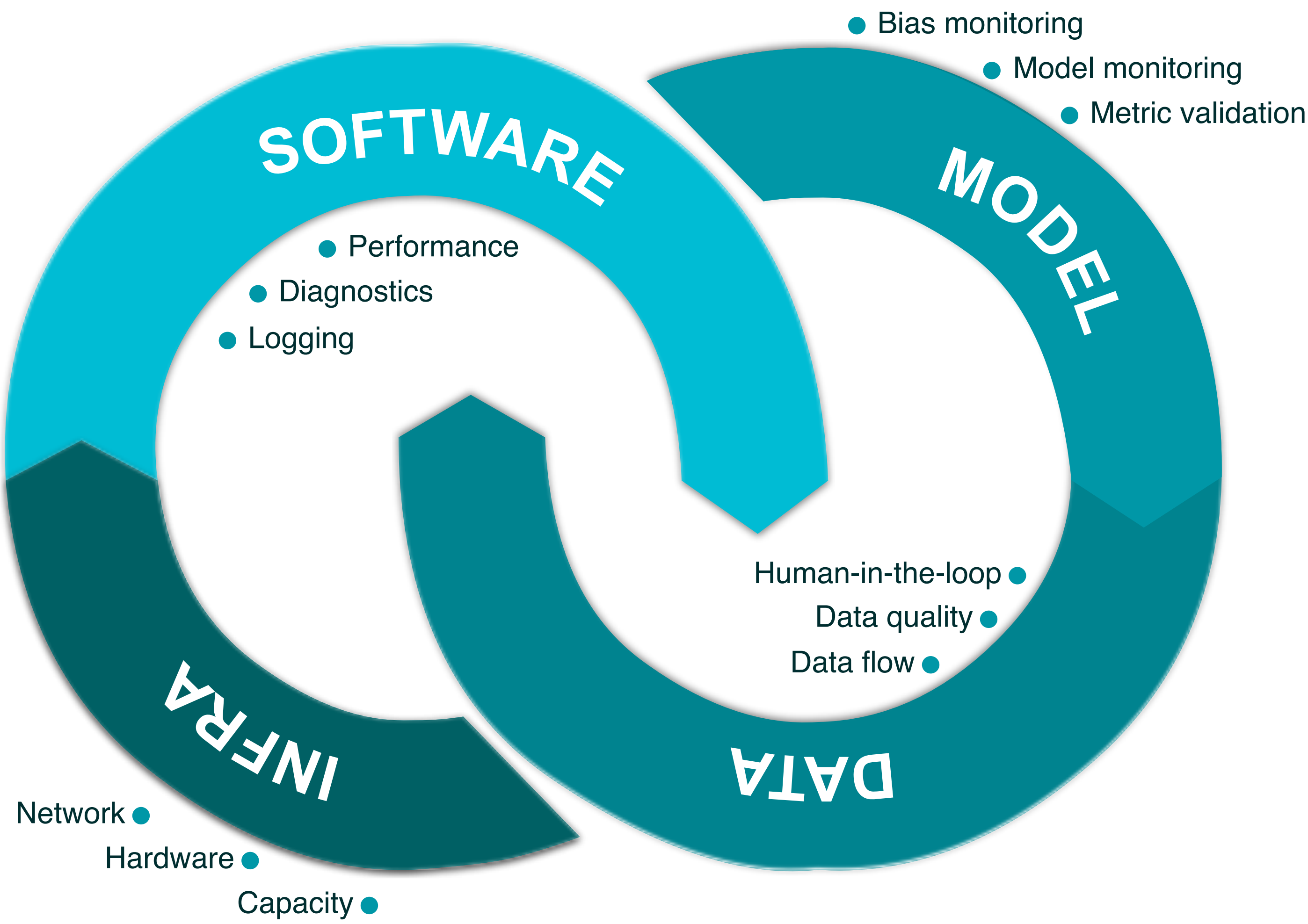

During ‘Deployment for AI’ we train our models for production and deploy them to the production environment. This can be a public or private cloud with a GPU cluster. An important part of the process is to version the AI/ML models along with the data and the code.

# 6. OPERATE

“You cannot always control what goes on outside. But you can always control what goes on inside.”

Wayne Dyer



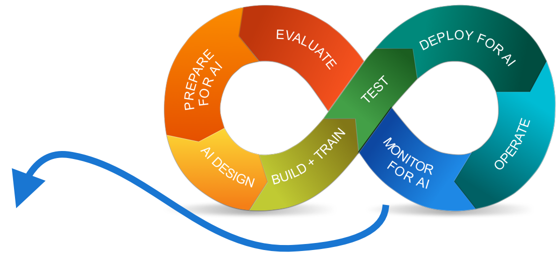



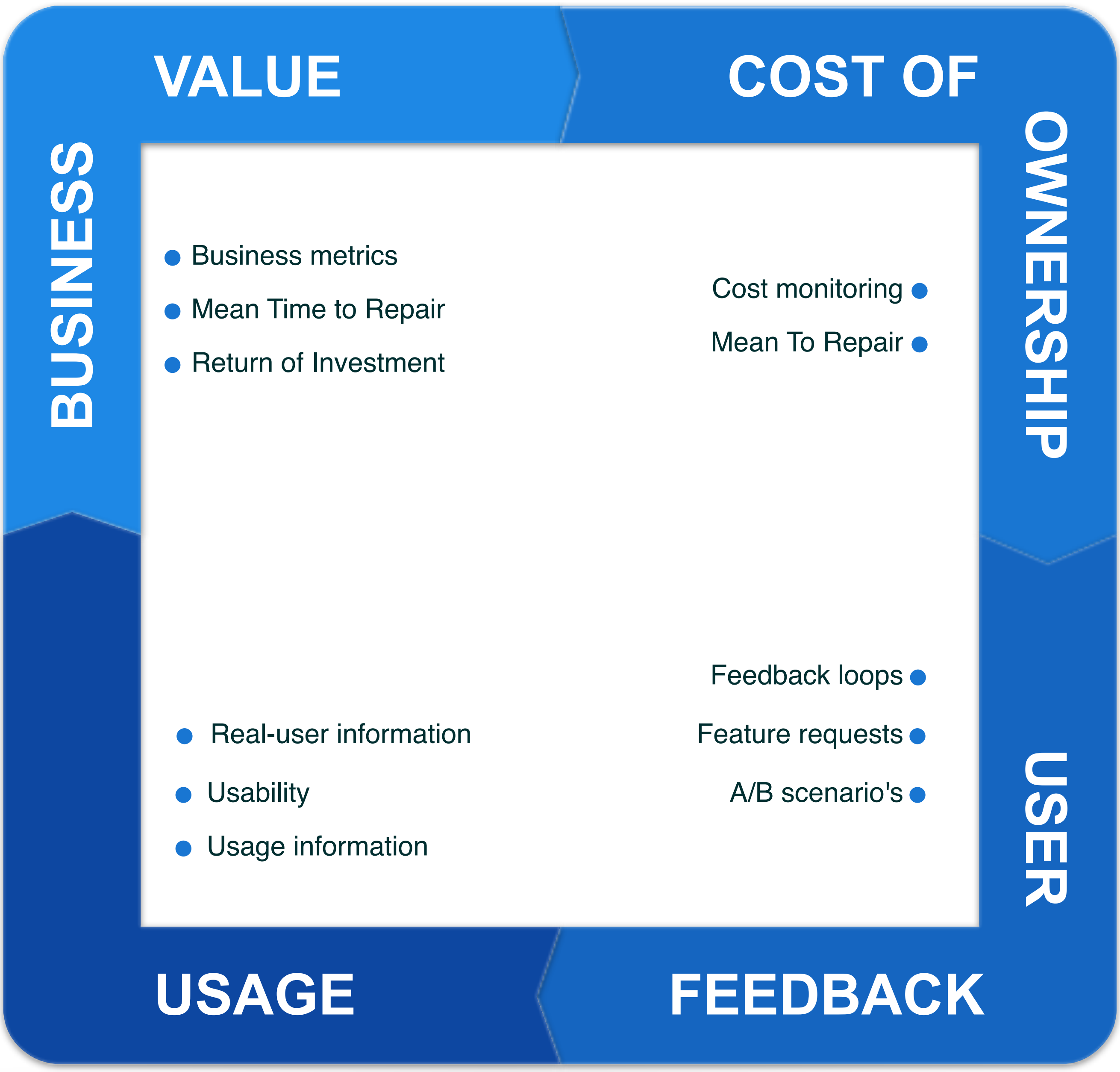

During ‘Operate’ we make sure the system keeps running the way as defined. We operate the software, the infrastructure, but also the model and the data for the AI part of the solution. We make sure the model keeps performing as specified even when it keeps learning and evolving.

# 7. MONITOR FOR AI

"I'm still passionately interested in what my fellow humans are up to. For me, a day spent monitoring the passing parade is a day well-spent."

Garry Trudeau

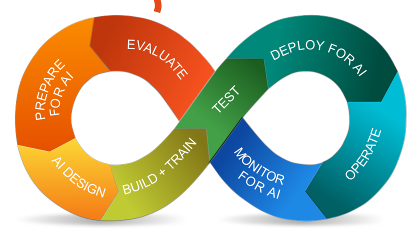



During 'Monitor for AI' we monitor our process as well as our product. We monitor if we meet the expected business impact and challenge this against the running cost. We also monitor the usage of our product and collect user feedback to find improvements.

# 8. EVALUATE

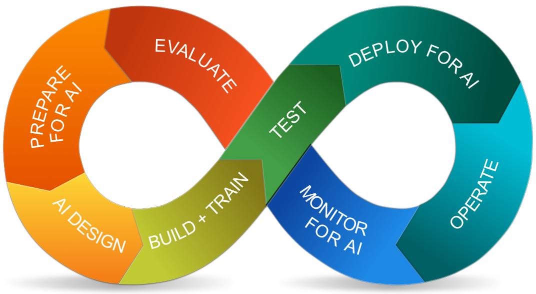
"You cannot always control what goes on outside. But you can always control what goes on inside."
Wayne Dyer





During 'Evaluate' we validate our business case based on our learning and outcomes of the product and process. We determine what to improve and pivot accordingly. We can pivot our business strategy, product or process but we stay grounded in our vision and learned facts.

# DEVOPS CAN PREVENT AI EXPERIMENTS




AI technology is so new and promising that we actually have no idea what we can do with it. This combination of characteristics makes AI as fascinating as it is complicated to deal with.

Although more and more time and budget is being made available within companies and institutions to investigate the possibilities of AI, the actual application is often still in an experimental phase. After that phase, the moment should follow when a new AI tool is actually taken into use by the business; and that is exactly where it still lingers.

The reasons why AI experiments still often end up in sight of the finish are not new - they are phenomena IT has been dealing with for years. Fortunately, there has also been a solution for these obstacles for years, namely DevOps teams. This way of working not only ensures that an AI team has the right composition (a combination of developers and end users); it is also in the mindset that belongs to DevOps, in which you always try to achieve a better result in small steps.

DevOps therefore has the potential to overcome the main pitfalls of AI experiments (and in particular that translation into practice). And that's in three success factors.

## The use of qualitative data

The success of an AI solution depends on the data that you have at your disposal and the quality of it. Ideally you as a development team want to have access to many different data sources within the organisation to test and develop. In practice things go wrong there soon. The owners of databases are often spread over the organisation, so that a lot of time and energy can be spent collecting data. Teams

AI technology is so new and promising that we actually have no idea what we can do with it. This combination of characteristics makes AI as fascinating as it is complicated to deal with.

Although more and more time and budget is being made available within companies and institutions to investigate the possibilities of AI, the actual application is often still in an experimental phase. After that phase, the moment should follow when a new AI tool is actually taken into use by the business; and that is exactly where it still lingers.

The reasons why AI experiments still often end up in sight of the finish are not new - they are phenomena IT has been dealing with for years. Fortunately, there has also been a solution for these obstacles for years, namely DevOps teams. This way of working not only ensures that an AI team has the right composition (a combination of developers and end users); it is also in the mindset that belongs to DevOps, in which you always try to achieve a better result in small steps.

DevOps therefore has the potential to overcome the main pitfalls of AI experiments (and in particular that translation into practice). And that's in three success factors.

## The use of qualitative data

The success of an AI solution depends on the data that you have at your disposal and the quality of it. Ideally you as a development team want to have access to many different data sources within the organisation to test and develop.
In practice things go wrong there soon. The owners of databases are often spread over the organisation, so that a lot of time and energy can be spent collecting data. Teams engaged in AI experiments therefore have a tendency to do so with the (limited) data that are easily and quickly available. And then we have not even talked about checking the data quality, where you as a team will really have to pass IT - voilà, an extra barrier.

A DevOps team not only involves developers and end users, but also team members who are responsible for the data strategy and quality.

## Learning to live with a margin of error

IT is, by definition, quite an exact field and that makes it difficult for us if results are not 100% accurate. This desire for perfection that typifies the average IT person can slow down or even prevent the putting into practice of a new AI solution.

An example: when a team will develop an AI chat bot that is used to handle customer inquiries, the team usually wants a solution that always comes up with the right answer to questions. If the accuracy then turns out to be 'only' 80 percent, then one conclusion may be that the tool is not working properly. What is actually forgotten is that people are not flawless either; a customer service representative does not always immediately know the right answer to a question. In these cases you have quickly made a business case if the accuracy of an AI chatbot can be improved, for example from 72 to 80 percent. That in 20 per cent of cases a customer service representative is needed, that does not have to be a problem; if the investment only pays for itself in the improvement you have made.

This too is a typical example of a DevOps approach; an incremental improvement or 'minimal viable product' also has a value, especially in the experiment phase. Learn to live with small steps; several small steps eventually lead to a big impact.

## Involving the end user

But perhaps the main advantage of working in DevOps teams is to involve