Fraunhofer

Julius Pfrommer | Niclas Renner | Lukas Rauh | Constanze Hasterok | Holger Kett | Janina Bierkandt | Jürgen Falkner | Damian Kutzias | Marco Huber

AI Beyond The Prototype

Requirements for Long-Term Operations of AI in Industry

KI Fortschrittszentrum
LERNENDE SYSTEME UND KOGNITIVE ROBOTIK

CC-KING
Competence Center
KI-Engineering

Opening Message

AI Creates Value in Long-Term Operations

The allure of Artificial Intelligence in industrial applications is undeniable. Many prototypes offer a vision where data-driven analytics and optimization bring a competitive advantage: Reducing the resource consumption, improving machine parameters, increasing the output quality, detecting anomalies and predicting failures, and so on.

Yet, amidst the excitement of initial experimentation, it is easy to overlook the critical imperative: AI-based systems in industry create their value in sustained long-term operations. And many AI prototypes, however successful, never make it into the operational phase. From this we conclude that the transition from a prototype to an industrialized solution is a difficult task and that the requirements for sustained operations go much beyond what a prototype can deliver.

Requirements Beyond the Prototype

In the age of agile software development methods, the explicit definition of requirements may fall to the wayside. However, the long-term productive use of AI in industry comes with challenges that only become apparent in the operational phase. Then it can be already too late to make fundamental changes. Hence, we advocate to anticipate the requirements for successful operations early on. That way, deliberate decisions can be made throughout the different phases of development.

Collaboration of AI and Domain Experts

Successful AI projects in industry need a tight collaboration between AI experts and domain experts. Especially when solutions from both sides get integrated into an overall AI-enabled system. As part of development, the common vision of the eventual solution gets increasingly more concrete. But without a common language (verbal, written or even visual) to express it, this vision can deviate substantially between stakeholders without them being aware. Stating high-level requirements starts the dialogue between AI experts and domain experts, ensures that misunderstandings can be clarified early on, and lays the basis for joint development.

Improving the Success Rate of AI Initiatives

This document aims to improve the success rate of AI initiatives in industry. To do so, it provides a framework for the definition of requirements beyond the prototype. This enables a dialogue with respect to development goals between technical specialists from industry and the AI domain, as well as the synchronization with decision makers and organizational leaders. Beyond this document, the initiatives of *KI-Engineering* and *AI Innovation Center* develop technical and organizational tools for AI in industry.

We wish our readers much success on their journey towards »AI beyond the prototype«.

Constanze Hasterok Marco Huber Thomas Renner

Requirements Beyond the Prototype

The step from a working prototype to a long-term solution comes with many difficulties and pitfalls. Especially the requirements change beyond the prototype. We make these differences explicit so the operational requirements can be better anticipated.

Operations Scenarios for AI in Industry

We distinguish four archetypes of operations scenarios shown in Figure 1. They are different along the axis of i) the criticality of operations (does the business or even human health directly depend on the functionality) and ii) whether development and operations are handled by the same organization or not.

Prototype A prototype is developed to demonstrate the feasibility of a solution approach. It is typically operated under close supervision of the original developers. For that, also substantial changes and improvements can be made to the system during operations.

Internal Deployment An internal deployment is different from the proof of concept as the system developers are no longer part of the daily operations. However, if the deployment remains within the same organization as the development, then it remains possible to share data between the development and operations teams.

Trial Operations Trial observations are non-critical like a proof of concept. Trials are however performed in an external environment. The difference is that trial operations are often limited in time and scope and are done not in order to validate a method, but to showcase performance before advancing into the external deployment stage.

External Deployment External deployment is both critical and operated by an organization that is not the original developer of the system. Here, the requirements for the technology-readiness level is the highest. Additional requirements are for documentation and training, as well as the agency of the operator to perform small maintenance tasks in-house.

Critical usage	Internal Deployment	External Deployment
Non-critical	Proof of Concept	Trial Operations
	Internal use	**External use**

Figure 1: AI Operations Scenarios.

Requirement Categories

We group the requirements for AI beyond the prototype into six broad categories. The remainder of this document discusses each category in a dedicated section.

1. Autonomy Level
2. Performance
3. Supervision and Maintenance
4. Integration and Deployment
5. Acceptability
6. Regulation Compliance

The relative importance of the requirement categories is typically very different between the operations scenarios. Figure 2 shows the result of an internal expert survey for the respective importance of each requirement category. As an example, the long-term availability of a solution can be neglected for the Proof of Concept, wheres an external deployment needs to cover this requirement so that the operating organization can rely on a technology that has become critical to its business.

To transfer the prototype into operations, the changes to the requirements can sometimes be so serious that development has to be restarted after the Proof of Concept. In other cases an iterative refinement of a prototype solution can reach a maturity level high enough for critical operations.

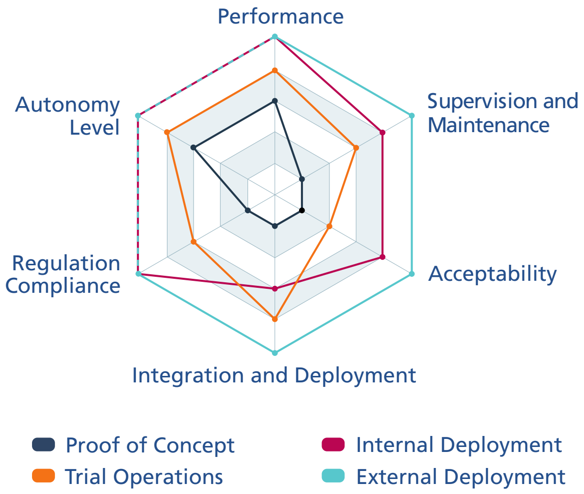


Figure 2: Importance of the requirements categories.

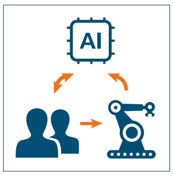
1. Autonomy Level

The autonomy level is a business requirement with a strong impact on all other requirement categories. For example concerning performance and robustness and the appropriate human-machine interfaces. And also repetitive steps for Machine Learning itself can be automated.

Autonomy Level of AI in Industry

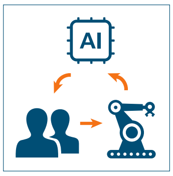
There exist several definitions for AI autonomy in industry, for example for Industrie 4.0 [18] and autonomous driving [19]. We aggregate and simplify them into four autonomy levels.

1. Assistance Functionality



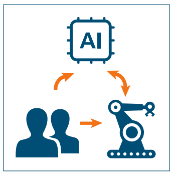
A human operator continues to make all decisions and bears the full responsibility. The AI functionality assists the operator for decision-making in challenging situations. For this the operator needs visibility into the ongoing operations with appropriate interfaces to access the assistance functionality and to interact with the underlying system.

2. Human in the Loop



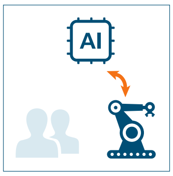
The AI functionality observes the behavior of the system at runtime and proposes courses of action, that are approved or overridden by the human operator. The responsibility still rests with the human operator. The regular human feedback (accepting or overriding a proposed course of action) can be used for the continuous improvement of the AI functionality.

3. Human Supervision



The AI functionality acts autonomously within defined limits. Human supervision and intervention is possible—in some cases the constant presence of a human operator can be required. When the system leaves its defined limits, control is returned to the human operator or the system switches into a known-safe operating mode.

4. Full Autonomy



In fully autonomous operations, human operators have no possibility to influence the system's behavior. This comes with high requirements for the AI performance and robustness. Full autonomy is typically only possible in tightly controlled environments that do not deviate from the assumptions made during development. On the upside, the requirements for human-machine interaction are reduced.

Machine Learning Automation

Most AI methods today make use of Machine Learning to generate models from data. A Proof-of-Concept development may include many manual steps to generate a model (see blue elements in Figure 3). However, during operation "beyond the prototype" the model has to be monitored for its validity and updated, either continuously or from time to time. In this case, automating the steps is advantageous in terms of reproducibility and saving time. The choice of which steps to automate depends on the individual circumstances of an application. Figure 3 indicates different possibilities for the automation of ML-based model generation and when to use them.

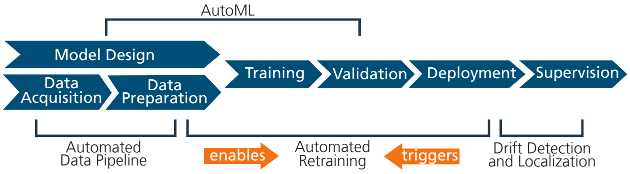


Figure 3: Automation of Machine Learning steps.

AutoML is used in the initial development phase to explore the solution space of possible models [10]. This involves the selection of the model type, optimization of hyper-parameters for the learning algorithms, neural architecture search, data augmentation, and so on.

Automated Data Pipelines are advisable when data continues to be aggregated for training after the initial deployment. Otherwise, manual data acquisition, integration and cleanup are big efforts that introduce friction and typically degrade the quality of the continuously aggregated data. Steps that are too difficult to automate can be made more efficient with appropriate user interfaces.

Automated Retraining allows more frequent updates when AI models continue to be improved after the initial deployment. Additional advantages are less human errors and the possibility to make model updates without the continuous presence of ML experts. Typically the model verification is automated also for a high confidence in the update model.

Drift Detection and Localization give an alert when a model no longer accurately reflects the current reality. Then automated retraining can be triggered besides cyclic model update intervals or human intervention. Localizing the drift in a larger system helps to update only those parts that have changed and hence minimize the data requirements.

2. Performance

AI performance refers to the accuracy, robustness and speed with which AI systems process information, make decisions, and execute tasks. This is important to any AI application. Yet it can be tricky to navigate the performance tradeoffs and to define how much performance is "good enough".

Model Performance

AI models can have many different purposes. For the discussion of model performance we however focus on predictive models. Predictive models come in one of two kinds: classification and regression. Classification produces nominal outputs (such as a country classification; Germany, France, Netherlands, and so on) and regression models predict numerical values (e.g. 21.5 degrees Celsius). The shaded box on the right shows examples for the most commonly used performance metrics for classification and regression.

The performance requirements for an AI application can come in the form of a threshold for a well-known metric. In addition, the performance requirements can state a trade-off between different goals. A typical case is the trade-off between false-positives and false-negatives. In the case of a medical equipment, where false-negatives lead to missed treatment opportunities, one way to account for this is to set the *weighing-factor* for the cost of false-negatives much higher than that of false-positives in the evaluation metric

The performance metric not only plays an important role for the evaluation of a model, but also for Machine Learning itself. The task of the learning algorithms is to reduce the prediction error on the examples in the training data (the technical term for this is *empirical risk minimization*). It is also possible to use different metrics for model training and evaluation. For example when additional *regularization terms* are added to the training metric to account for prior knowledge or to reduce overfitting when not enough training data is available.

Example Performance Metrics

Below are example performance metrics for classification (top) and for regression, i.e. prediction (bottom). The metrics are defined for an empirical dataset $(x, y) \in D$ with N input-output pairs. For classification, we consider binary classification with two possible outcomes (positive and negative). The outcome of a classifier is either true or false for the respective sample. The number of true-positive classifications on the empirical samples is tp, the number of false-negative classifications is fn, and so on. For the regression case, the output of a predictive model $y \approx m(x)$ is compared to the empirical data.

Metric	**Description**
Accuracy	The probability of the model classifying any of the samples correctly: $\frac{tp+tn}{tp+fp+tn+fn}$
Precision	Given a positive classification, with what probability is this correct: $\frac{tp}{tp+fp}$
Recall	The probability of a positive sample being correctly classified as positive: $\frac{tp}{tp+fn}$
Mean Squared Error	Average squared difference between the prediction and the empirical outcome: $\frac{1}{N} \sum_{(x,y) \in D} (m(x) - y)^2$
Mean Absolute Error	Average absolute difference between the prediction and the empirical outcome (less sensitive to outliers): $\frac{1}{N} \sum_{(x,y) \in D} \lvert m(x) - y \rvert$

Model Robustness

The robustness of AI systems refers to their ability to perform reliably and effectively even in unfavorable or different conditions compared to its training and validation setup. Tightly controlling the environment of an AI-based system in industry to remain exactly identical during long-term operations adds additional cost and hindrance. Hence changes can occur due to noise from stochastic processes and sensors, drifts in the external and internal operating conditions, adversarial attacks, as well as hardware and software failures. Furthermore, in a complex enough environment it is not possible to have all possible scenarios in the training data. So the system should be able to cope with new and not foreseen situations also. Overall, robustness to small changes is usually a strong requirement for long-term operations.

The degree of the required robustness varies across different applications. For critical use cases like autonomous driving, medical systems, or financial systems, robustness becomes vital. It becomes the shield that protects against accidents, misdiagnosis, or financial losses, thereby safeguarding human lives and preventing catastrophic consequences. On the other hand, even in cases like image recognition, where the stakes might not be as high, robustness remains crucial in order to ensure the trust in AI systems. It ensures accurate and reliable results, even when the input data is noisy or ambiguous.

AI systems are often referred to as black-box systems. The sheer complexity can make their internal workings opaque to human understanding. That makes it almost impossible to assess the robustness of an AI system by gaining insights into its inner workings. The robustness becomes apparent when exposing the system to (artificially created) challenging scenarios. But the way these are created and evaluated is highly application-specific.

We propose a few robustness dimensions along which requirements can be specified and verified.

Model Input Sensitivity In layman's terms, in a well-behaved AI model, small changes in the input result in small changes in the output. Similar to hysteresis (e.g., debouncing of a physical button) this reduces the amount of flicker in the output from small random changes. For classification models the same can be achieved by having large areas in the input-space lead to the same classification output.

Handling of New Situations If the data coverage of possible scenarios is not complete, the model performance should remain high are "degrade nicely" when exposed to scenarios that are out-of-distribution to what was available during development. This is captured already in the typical split of training- and test-data. But new situations can also arise that change assumptions used during development that are not even part of the data.

Self-Adaptation over Time Robustness over longer time-periods can also refer to the self-adaptation of the system to changes. This can be achieved by technical and also by organizational means. In relation to that, supervision and maintenance are discussed further in the next section.

Table 1 lists contributing factors to robustness, spanning the entire life cycle of an AI system, starting from data collection and extending to post-deployment monitoring. Paying careful attention to these factors is essential for building a robust AI system. In an ideal scenario, the required level of robustness is determined even before data collection, during the stage of defining the use case.

Factor	Description
Data collection	Scope, completeness and absence of bias of collected data.
Data labeling	Correctness and absence of ambiguity of labels.
Data splitting	Splitting data according to specific features like imbalanced data to prevent bias and leakage.
Preprocessing steps	Meaningfulness of preprocessing steps and independent application on data splits.
Model selection	Meaningfulness of model selection and prevention of unnecessary complex models.
Performance metrics	Relevance of performance metrics.
Hyper-parameter optimization	Prevention of underfitting and overfitting.
Model evaluation	Completeness of evaluation procedure including edge cases.
Model confidence	Calibrating of model confidence.
Testing in real-world setting	Extensive testing under real-world conditions.
Monitoring	Monitoring of model performance over time.
Drift	Detection of data and concept drifts.
Machine Learning Operations (MLOps)	Usage of established MLOps methods and frameworks.

Table 1: Contributing factors to AI robustness (from [2]).

Processing Time

The processing time of training and execution of AI algorithms can be problematic when either the waiting time for results becomes too long or if the processing time cannot be covered by the energy envelope (cost or availability of energy). For machine learning, the processing time for inference (execution of the model after training) is usually considerably faster than the training itself. So the two cases need to be considered separately. Overall, the processing time depends primarily on three contributing factors: data, algorithms, and hardware.

Data The data used for training and at inference affects processing time in various ways. The number of data points directly influences processing time during training and inference. Equally important is the type of data. A data point can be a row in a data table containing a few bytes, an image can comprise just a few KB or even MB of information, etc.

Algorithms An appropriate algorithm has to be chosen depending on the available data and the use case. Typically, more complex data structures (e.g., images, text, or time series) require more complex algorithms than data tables. Many modern algorithms are structured to allow for massive parallel computations on specialized hardware.

Hardware Suitable hardware must be chosen depending on the algorithm and underlying data. GPUs with specialized tensor units allow for massive parallelization, which speeds up the training of modern neural networks immensely. However, more traditional algorithms like support vector machines and decision trees often do not require specialized hardware. Different hardware stacks are used for training and inference due to the lower computational cost at inference. Also, latency is not a concern for training, but it may be for inference. Thus, it is important to consider whether cloud resources or edge devices are suitable for the given use case.

3. Supervision and Maintenance

Supervision and maintenance of AI-based systems is crucial for ensuring long-term operations. Crucially, AI models in industry may require maintenance and updates after changes in their operational environment. These changes can be caused by wear and tear, component replacement, different input material properties, and so on.

The majority of industrial machine learning use cases operates on data streams whose structures and distributions can change over time. This non-stationarity makes the supervision and maintenance of ML models a crucial requirement in order to ensure their long-term operation. In practice, non-stationarity of data streams in machine learning use cases results in a cycle between operation and maintenance (see Figure 4). When the monitoring of the ML application shows a degradation of performance or unexpected results, maintenance takes place where new data is acquired and based on that, the ML model is adapted. In an ideal world, all steps within this cycle are automatized. In practice, however, many steps are manually executed and supervised by humans in order to gain insights into the behavior of the system.

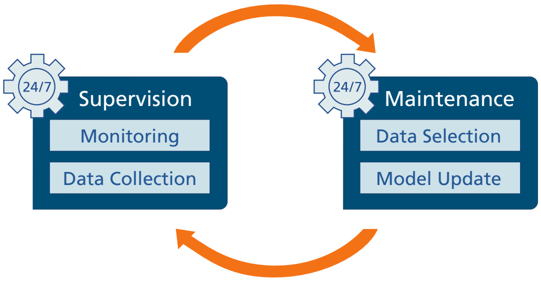


Figure 4: AI operations and maintenance.

There are various reasons behind the non-stationary statistical characteristics of data streams, leading to a decline in model performance. These reasons encompass a range of factors such as adjustments in machine settings, alterations in sensor parameters (including noise, resolution, calibration, and aging effects), deterioration in the quality of equipment materials, seasonal variations, fluctuations in operator preferences and behaviors, adversarial actions, and hardware or software malfunctions. We commonly refer to alterations in data streams as *drifts*.

Drift Detection

To effectively use machine learning models with data streams prone to drift, continuous monitoring of both the models and incoming data is essential. Neglecting this requirement will result in models that become less precise and less reliable over time. In most real-world scenarios, drifts occur unexpectedly and are difficult to predict.

To formalize our understanding of drifts we construct a learning problem with a set of features X and a set of targets Y. The challenge is to find a model $M : X \rightarrow Y$ by training on historical data.

We distinguish between two types of drifts: data drift and concept drift. Data drift refers to a change in the feature distribution $P(X)$ with the feature space X, while concept drift refers to a change in the dependency of the targets Y on the features, i.e., in conditional distribution $P(Y|X)$.

Figure 5 shows a visualization for a data drift. A model trained on the data that is shown in the left canvas will have difficulties to predict the target y for the three green data points in the right figure since the model was not trained within that feature space region. The green line shows the ground truth concept from which the data originates and the red line shows the feature distribution $P(X)$.

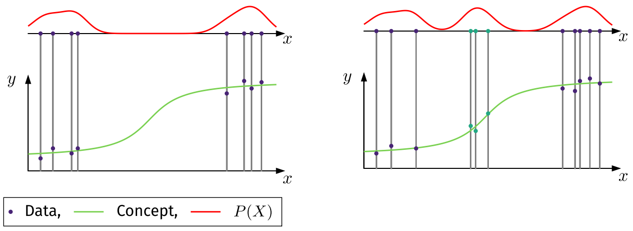


Figure 5: Example for a data drift that reduces the performance of ML models. Figure from [8].

Figure 6 shows a concept drift. The ground truth concept (green line) is shifted upwards in the right image compared to the left. A model trained on the data shown on the left will not be able to properly predict the target y given the X-values of the data points in the right canvas.

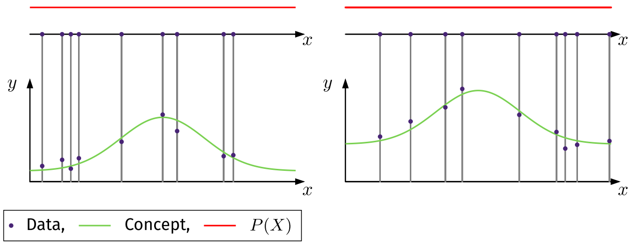


Figure 6: Example for a concept drift, i.e., a drift in $P(Y|X)$. Figure from [8].

The effect of such drifts is shown in Figure 7, which compares the absolute error (AE) of two ML models on a simulated data stream over time. At time step 23 a sudden drift occurs. The AE increases for both ML models as a result of the drift. A classical

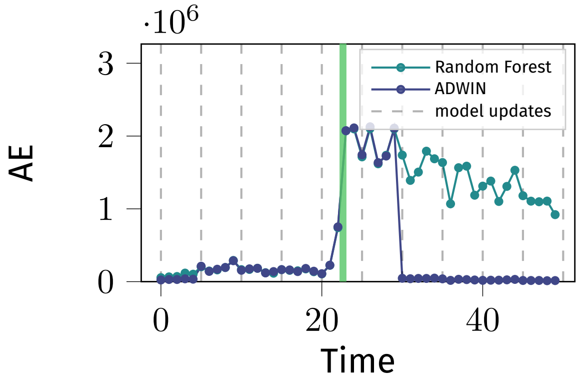


Figure 7: Absolute error (AE) of two ML models (random forest regressor (RFR) and RFR combined with ADWIN) applied to a data stream. ML models are evaluated on each new data point and are retrained on the historical data combined with the new data batch. Figure from [8].

Random Forest model is not able to adapt to the new state and hence, after the drift the AE only decreases linearly with increasing amount of data over time. In contrast, the adaptive method ADWIN [1] monitors statistical changes in the data stream (features and targets), detects the drift, aggregates data from the new state and triggers a re-training of the ML model based on the new data. Hence, a short time after the drift occurred, the performance is at its original level. Consequently, the only way to detect drifts is by closely examining the data and the model performance.

Drift methods commonly incorporate the following functions: data aggregation, change detection, and model adaption. Each algorithm has a different way how these functions are implemented. ADWIN, for example, uses a windowing technique to aggregate batches of data from the stream. The change detector compares two windows consisting of multiple batches with each other based on the mean of the distributions within these windows. If a change is detected, the model is re-trained based on the latest data window. If no change is detected new data batches are added to the window. Other data aggregation techniques are the aggregation of the full data stream since it's beginning, abstraction of all data into a model representing the data distribution etc. Change detectors can be based on statistical metrics either of the features (and targets if available) or of the model performance if targets are available. Common statistical metrics are the mean, variance, Wasserstein distance, etc. The functionality of model adaption is not part of all drift methods. Some only output an alarm if a drift is detected, some automatically trigger a re-training. (see section .

The mentioned methods for realizing data aggregation, change detection and model adaption are only examples. Developing a consistent strategy how to derive them for different drift characteristics is an active field of research.

Retraining for Maintenance

For machine learning systems, the identification of a drift necessitates a corrective response to restore prediction performance to satisfactory levels. When identified, the drift initiates a retraining process—a process to update the machine learning system with a new model version—similar to a maintenance task for physical systems.

Data Selection for Retraining Data selection for retraining is a crucial aspect of maintaining the effectiveness and relevance of the resulting model. Some drift detection methods automatically trigger a re-training of the model based on a selection of data. One option is to use the latest batch, window or data point and forget everything before. Another one is to use a discounting factor that assigns lower weights to older data samples than newer data samples. The golden batch strategy focuses on selecting data that best reflects the evolving landscape and allows the model to effectively adapt to new patterns while mitigating the risk of overfitting or bias. This careful selection of data ensures that the retraining process includes relevant information and improves the adaptability and robustness of the model.

Retraining Streaming data offers the potential to enhance machine learning models over time by incorporating new incoming data to enrich the data pool used for training and thus integrating new information into the model. In this way, machine learning models are able to adapt to feature or domain drifts in the data and deliver consistent prediction performance. However, depending on the specific machine learning problem, this flexibility in terms of the adaptability of the model is associated with varying costs.

Primarily, retraining costs depend on the type of machine learning model used in the solution. The cost dynamics are shaped by the methodology involved in model training, whether it necessitates manual labeling of new data (supervised learning) or can undergo automated training utilizing raw data supplemented by automated data preparation steps (unsupervised learning). Clearly, the involvement of manual tasks requires more resources in the form of data management and labeling tools, potential infrastructure, and human resources. The frequency of retraining additionally impacts resource allocation and therefore requires a balance between optimizing performance and operational expenses. Distinct methodologies delineate suitable retraining intervals. One approach involves reacting to identified data or model drift, while another entails regular retraining based on the accumulation of pertinent new training data or temporal intervals, utilizing the entirety of available data at each iteration.

Automated Retraining Even further, automated retraining drives the optimization of the continuous operation of machine learning systems to the extreme. Automated retraining, when suitable for the type of machine learning model, is a proactive approach to streamline the retraining process by automating model updates in response to new data or identified performance degradation caused by drifts. By leveraging automated triggers, robust pipelines for data, and model retraining systems, adapt models to changing dynamics with minimal or even without manual intervention with the goal of achieving consistent model performance. Therefore, automated retraining incorporates comprehensive monitoring mechanisms to firstly detect data drifts, model degradation, or operating conditions and secondly track automated changes and their impact on the machine learning system. The introduction of automatic retraining not only increases the adaptability of the model but also minimizes operational downtime and strengthens the reliability of the model in dynamic real-world scenarios.

Redeployment Enabling uninterrupted model updates within live applications necessitates an integrated pipeline automation process. The introduction of a new model mandates achieving, at minimum, comparable prediction performance as the preceding models on the expanded dataset. Automated testing and integration via a continuous integration pipeline validate this performance continuity. Thereafter, a continuous delivery pipeline facilitates live updates seamlessly. Tailored deployment strategies such as A/B testing or canary deployments, contingent on the application, mitigate downtime and customer disruption, ensuring a smooth transition.

When updating models, it is of utmost importance that the relevant model metrics and data streams are closely monitored. This is particularly important in highly automated pipelines, such as following the active learning paradigm, where the integration of new training data is predominantly automatic. Ensuring data quality through vigilant monitoring is essential in such scenarios to maintain the integrity of the model's performance and accuracy.

Modeling Strategies with Additional Characteristics Within the spectrum of modeling techniques, transfer learning, active learning, and federated learning present distinct strategies, each influencing the retraining and maintenance of AI models. Transfer learning harnesses pre-existing knowledge from a source domain to expedite learning in a target domain, notably reducing the need for extensive retraining. On the other hand, active learning optimizes retraining efforts by iteratively selecting data points for annotation and refining models incrementally with minimal human intervention. Federated learning innovates the retraining process in total by enabling model training across distributed environments while preserving data privacy, allowing continuous model enhancement without centralizing data. Comparatively, transfer learning expedites retraining by leveraging prior knowledge, while active learning strategically reduces annotation efforts.

Historizing and Versioning

While updating the machine learning system, with the machine learning model in its core, and iterating through revisions of data, models, and application code, the challenge of keeping track of revisions, their dependencies, and compatible counterparts arises. Maintainability of data, model, and the target system (embedding the machine learning system) extends the common approach of tracking application code known from traditional software development practices and drives reproducible machine learning strategies.

Reproducible ML Reproducible machine learning relies significantly on robust versioning to track model versions and runtime environments. Versioning aids in quickly identifying errors by linking specific predictions to their respective model versions, enhancing debugging and system reliability. Additionally, clear versioning ensures transparency and compliance, documenting the exact models used for predictions, which is essential for accountability and regulatory adherence. Lastly, versioning minimizes discrepancies between development and production environments by maintaining consistency across deployment environments. Therefore, the versioning process for reproducible machine learning systems revolves around three main artifact types: data, models, and the target system.

- **Data Versioning** is essential for maintaining integrity and evolution during model development. Tracking relevant information when data points are added, removed, or modified preserves a historical record of iterations and enables collaborative efforts among diverse teams. Versioning data is specifically important for understanding performance drifts and accelerating potential compliance requirements.
- **Model Versioning and Lifecycle Management** are important for clarity and accountability of machine learning systems. The lifecycle management of versioned models necessitates constant tracking of iterations, maintaining a comprehensive model history, and ensuring reproducibility of model outcomes. Essential details like configurations, hyperparameters, and involved data must be captured.
- **Target System and Environment Versioning** is crucial for comprehensive impact analysis and system stability. The integration of ML (and its runtime environment) into production systems results in complex integrated systems with a large number of hardware, software, and ML components. Documenting changes in these systems is vital, particularly for data-driven setups where performance inconsistencies often require contextual insights for resolution. Starting documentation could include details on environment configurations, software versions, and dependencies, helping quickly pinpoint issues and resolve them effectively.

4. Integration and Deployment

AI-based systems need a technical environment for computation, storage and communication. The different parts of the AI solution can have very different requirements (e.g., for training vs. operations) and span heterogeneous IT and OT environments.

An AI-based system in industry typically decomposes into system components that each have different requirements for their integration and deployment environment.

Decomposition into System Components

The decomposition is highly specific and different for every use case. But there are functions that typically get grouped together in a system architecture. We discuss these typical groupings as a guideline for the system decomposition. The relevant requirements for the integration and deployment environment are discussed afterwards.

Data Collection and Storage The data collection and storage can be thought of as a "data collection pipeline" in its own right. This includes the data collection, filtering and preprocessing, data integration (for example to synchronize timestamps) and storage.

We typically differentiate between different types of data. While a relational database (think SQL) could handle all of these in principle, there are specialized technologies for higher efficiency and convenience.

Time Series Data

- Example Use Case: Continuous sensor measurements
- Example Technology: Time Series Database (e.g., InfluxDB, ClickHouse)

Discrete Event Data

- Example Use Case: Alarm notifications
- Example Technology: Relational database with an index over the timestamp and alarm source

Unstructured Object / Blob Data

- Example Use Case: Camera images
- Example Technology: Object store using the S3 API

Meta-Data and Context Information

- Example Use Case: Product recipe information
- Example Technology: Relational database, Ontology-based triplestore

Infrequent steps in the data collection process can be handled manually. Each manual step however increases frictions when the data collection continues during operations. Then it becomes more difficult to maintain a current, high-quality and large dataset for retraining after the initial deployment.

Model Development, Training and Validation In a machine learning context, the model development builds upon the collected data. The data is used both for development and training (the model learns and gets optimized from the data) and for validation (empirical performance estimation of the resulting model). For that, the environment for the model development is typically catered towards the needs of developers—less so for the needs of an operational deployment.

Model Serving Model serving describes the operational application of a model after training. The long-term deployment typically happens in a different environment from the training. It can even happen that different programming languages are used for the model development and model serving. Then only the learned model parameters are shared between the two.

The application of a machine learning model is typically much more resource-efficient than their training. So the model serving can happen in a more resource-constrained deployment environment.

User Interfaces User interfaces (UIs) are essential for making AI systems accessible and user-friendly. User interfaces can be graphical layouts on computer screens. But interfaces can also be physical (e.g., buttons), voice-controlled, and so on. See Chapter 5 for details.

Human operators require interfaces to interact with an AI-based system as an assistance functionality, in a human-in-the-loop approach, or to supervise autonomous operations (cf. Section 1 on the autonomy levels). In addition, there can be interfaces where the operator needs to provide information to the system for its current operations and for the ongoing data collection. For example when he has manually changed machine settings or to annotate the reason for an unplanned machine downtime.

Supervision and Maintenance Supervision and Maintenance are described in Section 3. Depending on the level of access, supervision and maintenance also require dedicated user interfaces. These are different from the operator interfaces geared towards the daily interaction. For example the supervision might benchmark the performance across different factories and over a longer time-period. Then the information is not only required locally.

Integration and Deployment Requirements

The following categories for the integration and deployment requirements can be mapped to all components resulting from the system decomposition.

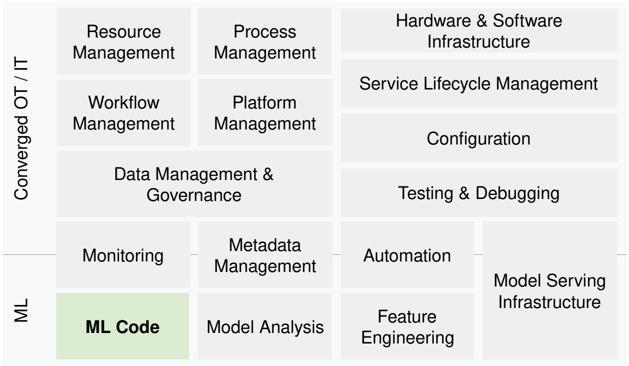


Figure 8: Exemplary illustration of the involved system components besides the AI components.

Centralized vs. Distributed Facing the challenges of data privacy, bandwidth, and availability of AI-based applications, the decentralization of AI-based applications has brought up new learning approaches with unique advantages but in turn other challenges and extended requirements.

Distributed and federated learning are pivotal in the context of decentralization, as they allow multiple AI models to learn from decentralized data sources without sharing the data itself. Distributed learning involves partitioning the learning process across multiple computational nodes, each processing a subset of the overall data. This approach is especially beneficial in environments where data is voluminous and geographically dispersed. By distributing the computational load, systems can handle larger datasets more efficiently, speeding up the training process while reducing the bandwidth needed for data transfer. In industrial applications, distributed learning can train AI models on data from multiple production facilities, enabling optimizations tailored to local conditions without requiring voluminous data movements to train one central model. Optionally, model updates can be returned to the central model to improve it over time without sending potentially critical and voluminous data.

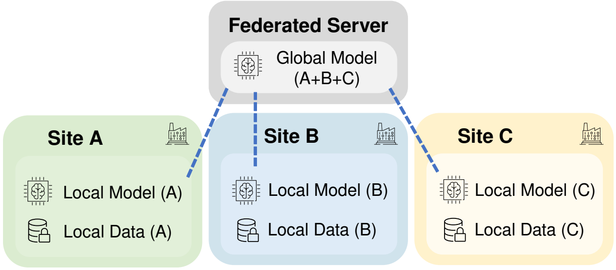


Figure 9: Illustration of federated learning with three sites.

Federated learning takes the concept of distributed learning even further beyond single companies. Each node in the federated training network trains a local model on its own data (as in distributed learning), but model updates are shared and then combined centrally. This is illustrated in Figure 9. This method not only preserves privacy but also minimizes the risk of data breaches. Federated learning is ideal for scenarios where data cannot be shared due to regulatory or competitive reasons, such as in healthcare or financial services. By leveraging federated learning, organizations can benefit from collective improvements in AI models while ensuring that each participant's data remains within its control.

Access to local resources Some function cannot be readily moved to a different environment because they need access to local resources. For the discussion, take the examples of access to OT interfaces for industrial field devices and GPU resources.

Field devices in an industrial setting have special requirements not commonly found in office IT. Operational technology (OT) encompasses tools and technologies for the operation of industrial applications, such as Programmable Logic Controllers (PLCs). Because of technological differences, OT solutions are often separate from enterprise IT systems. They further often use dedicated communication channels (such as fieldbus protocols or OPC UA in a dedicated TCP/IP network) that are not forwarded into the office-IT environment.

GPUs, the specialized hardware commonly used for machine learning, are typically deployed in specialized compute clusters. Typically large data volumes are first transferred to a storage that is close to the GPU. Then random data access patterns can be readily used on the local GPU resource for training.

Robustness Development typically happens with a developer present. And for the training of a fixed model, the training pipeline needs to work "just once". So the robustness of the development environment is usually not an issue. The model serving on the other hand has much higher robustness requirements. This includes physical toughness (e.g., in environments with vibration or high temperatures) and also the overall availability (e.g., no Windows Updates forcing a restart in the middle of operations).

Scalability and Bandwidth Requirements The volume of data and the available bandwidth are crucial in determining whether the data can be transferred to the cloud for processing or if local preprocessing is needed. This involves assessing the amount of data and ensuring that the bandwidth is sufficient for transferring data to the cloud. Additionally, the speed at which data needs to be processed plays into whether latency issues might necessitate local data processing, making it imperative to evaluate the infrastructure's ability to meet these requirements.

Scalability is crucial for AI systems to efficiently handle varying workloads depending on the lifecycle phase. It involves the system's ability to expand its processing capacity in response to increased demands. Key considerations include choosing between horizontal scaling (adding more machines) and vertical scaling (upgrading existing hardware), emphasising flexibility, cost-effectiveness, and the ability to manage data volume growth. Effective scalability ensures that AI applications remain responsive and cost-efficient under different operational loads.

Redundancy, Availability, and IT Operations Capabilities The need for redundancies in data, applications, and network access highlights the importance of computing infrastructure availability. This directly influences the decision on whether to manage IT operations in-house or to opt for outsourcing or managed solutions, depending on the available competencies and capacities.

New data continues to be aggregated from both production sites and is stored in a cloud environment. Data selection, preprocessing, and retraining are performed in the cloud environment where GPU resources can be dynamically added.

Access Control and Governance Identifying the stakeholders who need access to the primary data and processed outcomes, as well as determining the systems that require data access at various processing stages, are crucial for defining access control measures and the requirements for API management. Additionally, the necessity for different environments for development, staging, and production underscores automation requirements, emphasizing the importance of a well-structured system integration and deployment strategy.

Implementing decentralized learning methods requires careful consideration of data access rights and governance structures. The data generated by industrial applications can be highly sensitive in a competitive market environment, often coming with stringent access restrictions. For instance, a machine builder might supply an AI model where the machines are operated by a different organization that does not want to share operational data. In such cases, retraining of the AI models after deployment must be confined to on-site activities only, which significantly influences the overall integration and deployment architecture. Clear policies must therefore be established to define who can contribute to the model, how data is accessed, and how contributions are aggregated in decentralized learning approaches. These considerations ensure that all participants' data remains within their control, enhancing trust and facilitating compliance with regulatory requirements.

Company-Wide Deployment Strategy

There is no one-size-fits-all solution for a company-wide and overarching strategy for the deployment environment and integration architecture of AI-based applications. As an example for the environments close to the data source, while one company might prefer to set up its own on-premise data centers, another might opt for a specific cloud provider and its edge computing solutions. A number of trade-offs in terms of integration and deployment requirements affect organizational strategy decisions.

Cloud vs On-Site The choice between cloud and on-site deployments is pivotal in shaping a company's AI application strategy. Cloud-based solutions offer scalability, flexibility, and reduced upfront costs, as resources can be adjusted based on demand, and companies can leverage the latest technologies without significant investments in physical infrastructure. Furthermore, opting for a specific cloud provider can streamline operations through integrated services and support, potentially offering competitive advantages in agility and innovation.

On the other hand, on-site deployments provide companies with greater control over their data and systems, which can be crucial for meeting stringent data security and privacy requirements. This approach involves significant upfront investment in physical infrastructure and ongoing costs for maintenance and upgrades. However, it allows for customized solutions that are tightly integrated with existing processes and systems, potentially offering better performance for certain applications due to reduced latency and direct control over the hardware environment.

Technology Harmonization / Virtualization Virtualization is a key enabler for the ongoing convergence of OT and IT, affecting deployment strategies. Organizations can leverage the benefits of virtual machines, containers, and software-defined infrastructure by virtualizing OT systems. This allows the deployment of IT technologies and applications within the OT environment, enabling greater flexibility, scalability, and interoperability. In a converging infrastructure, the connectivity between OT and IT components is increasing. More IT capabilities are being introduced into the OT domain through this connectivity. This includes the integration of data analytics, machine learning algorithms, and advanced visualization tools into OT systems. The increased connectivity and virtualization of OT systems allow for improved data collection, analysis, and decision-making, leading to enhanced operational efficiency and productivity.

IT-Security Securing AI systems requires a company-wide IT-Security framework. Key measures include robust access controls to prevent unauthorized access to models and sensitive data and encryption protocols to maintain confidentiality. Regular vulnerability assessments and penetration tests are vital for detecting and addressing security weaknesses in these complex systems. Secure development practices, such as input validation and sanitation, are critical to prevent adversarial manipulation. Continuous monitoring is crucial for detecting system anomalies and responding swiftly to security incidents. The secure deployment of AI applications should ensure that interfaces and connections comply with industry-standard security protocols, with regular updates and patches to address new threats.

Furthermore, compliance with the international standard IEC 62433 is crucial for organizations in the industrial sector. This standard provides guidelines to enhance IT security and protect critical infrastructure from cyber threats. It emphasizes the importance of robust access controls, secure communication protocols, cryptographic protections, continuous risk assessments, and regular security audits. Compliance with IEC 62433 helps organizations proactively manage and mitigate cyber risks, essential for safeguarding industrial systems. By adhering to the standard and further security requirements as part of the company-wide deployment strategy, organizations can establish a robust defense against cyber threats targeting AI systems.

5. Acceptability

Human acceptance is key to the long-term success of AI-based systems. Acceptability needs to be achieved for the operating organization and the individual operators interacting with a solution.

Agency of the Operating Organization

The autonomy level describes the agency of the human operator. In addition, there is an agency on the organizational level. Often times the team developing an AI-based application is different from the operational team. In addition, the development team might come from an entirely different company, such as engineering service providers, consultants and research partners.

In our experience, operational organizations want to retain agency. They try to limit a possible loss of control if business-critical functionality is handled by outside parties. Hence, the agency goes beyond the daily operations and also encompasses supervision, maintenance and continued development.

Access to Data In addition to considerations of autonomy and agency, access to data is a critical aspect of operating AI-based systems in industry. The operational organization must have unrestricted access to relevant data sources to facilitate effective system performance and decision-making. Access to comprehensive and high-quality data enables the AI system to learn, adapt, and generate valuable insights that drive operational efficiencies and competitive advantage.

Continued Development Continued development of AI-based systems is essential for innovation and competitiveness in the industrial landscape. This involves introducing product- or system-level changes to enhance functionality, efficiency, and user experience. However, the process of continued development also raises questions about access control and intellectual property rights, particularly regarding code-level access and collaboration with external partners.

To remain agile and responsive to market demands, organizations must be proactive in introducing product- or system-level changes to their AI-based systems. This may involve implementing new features, optimizing algorithms, or integrating feedback from end-users and stakeholders.

Granting code-level access to external parties raises concerns regarding data security, confidentiality, and intellectual property protection. Organizations must establish clear protocols and agreements to govern access rights, responsibilities, and ownership of code assets.

User Acceptance

The operator agency describes the access levels and degrees of freedom for the human operator that interacts daily with the system. This is important as the success of many AI-based systems hinges on the collaboration with the human operator. Of course, this not only applies to AI-based systems, but to the introduction of new technologies in general. There are, therefore, already several approaches and models for categorizing and addressing the challenges of acceptance. According to the Technology Acceptance Model (TAM) [5], the two key factors for acceptance are the »perceived usefulness« and the »perceived ease of use«. These models have recently been expanded to include AI-specific aspects . A key factor here is usually »trust in the AI system«. There are various approaches to building and maintaining this trust.

Human-Centered quality The quality of the AI-based system also plays a decisive role in acceptance. In particular, the aspect of »perceived usefulness« is strongly influenced by this. However, quality is also a key factor for trust, as trust in the system's predictions decreases with every misjudgment.

The aspect of **»Human-Centered quality«** is particularly important for the quality of interfaces to users. ISO standard 9241-220 [7] is based on four dimensions:

- Freedom of harm from use
- Accessibility
- Usability
- User Experience

The interfaces should therefore be designed in such a way that errors are avoided, and users have barrier-free access. It is also about effectiveness and efficiency during use (usability). If the use also creates a positive feeling (user experience), all dimensions of »Human-Centered quality« are fulfilled.

Acceptance research has shown that these dimensions also contribute to trust. This means, for example, that people are much more likely to forgive errors in systems that they like to use than in systems they don't like. To achieve this effect, the user experience must therefore be high. This in turn requires the fulfillment of needs. One example is the need for competence; users want to see and experience themselves as competent at all times. With AI-based systems, attention must therefore be paid to how exactly the collaboration between humans and AI is designed so that users do not feel left out or less competent, see also the section Human in the Loop.

For the specific design of the interfaces, for example what should be displayed when and how in order to increase acceptance, please refer to the sister study: »Design of AI systems« [12]. It is based on a case study in which different design variants were compared with each other and evaluated by users.

User Documentation and Training User documentation supports users in learning and using systems. For onboarding, additional training is often offered in which the handling of the system is demonstrated with example use cases and users can practice. If the content of these formats is well designed, it can increase user acceptance.

Ideally, the content is adapted to the user's previous knowledge, skills, tasks, and roles. In practice, explanations of the meaning and purpose of individual functions and processes are considered particularly valuable by users. Understanding the context and background often increases trust in the system and therefore acceptance.

For AI-based systems, formulations that focus on the type and nature of support for the user are suitable. This clearly describes the added value and directly shows where the user's help is necessary, which aspects they are responsible for and what options they have. A nice side effect of training is the opportunity to learn how users behave and which aspects may cause difficulties. Similar insights can also be gained by analyzing which sections of user documentation are accessed and how often. AI-based systems may be able to use the data obtained in this way directly.

The extent to which it should be explained in the documentation or the system itself that AI is being used and how it works depends on various parameters. People often state that they want to know whether it is an AI-based system. From an ethical point of view and for reasons of transparency, it therefore makes sense to point this out. Explaining how the system works can increase acceptance (see the following section). However, the level of detail of the explanation must also be adapted to the individual users or at least the various roles.

Explainability Explainability refers to the ability to understand and interpret the decisions made by machine learning models. It allows to comprehend the reasons behind the predictions or classifications produced by an AI model. Explainable AI aims to develop models and techniques that provide transparent and interpretable results. However, there is often a trade-off between explainability and algorithmic performance. Highly complex models such as deep neural networks may achieve remarkable accuracy but lack interpretability. On the other hand, simpler models like decision trees or linear regression models are more interpretable but may sacrifice performance in certain scenarios. Balancing the trade-off between explainability and performance is a critical consideration when operationalizing AI solutions. It depends on the specific use case, the level of interpretability required, and the impact of potential errors or biases. Organizations must carefully assess the needs of their stakeholders and regulatory requirements to determine the optimal level of explainability in their AI systems.

Change Management When AI-systems are deployed in practice, change management may become relevant. Some AI systems just work in background without changes for the actual work and the corresponding humans. Others come with more or less intense changes for the work. This may result in the necessity to improve the skills of the affected employees or even completely change the job they do. This may need intense training which takes not only the time for training, but also for planning, at least when other departments or even other enterprises are involved.

In addition to the training time, the willingness of the affected employees is of relevance. Humans are creatures of habits. Learned and repeated working processes and steps are optimized and automatized by the subconsciousness, which is both advantageous (in terms of performance) and disadvantageous (for changes). Humans tend to reject changes of already learned working processes and steps, therefore some kind of change management should be used, especially, when the processes to be changed should work uninterrupted.

Many different variants for change management exist, whereas the most important factor is that the affected employees get enough time for a proper change and possible boycotts are prevented. A simple and classical variant is the 3-phase change management variant of Kurt Lewin, i. e. 1) Unfreezing, 2) Change, and 3) Refreezing.

User Feedback By involving users in the design of systems, their requirements and needs can be considered at an early stage. The human-centered design process [6] describes how this can look like in the various phases. This participatory design has also been proven to increase user acceptance.

It is also important to analyze user feedback after a system has been implemented. This involves both active feedback—the users speak up about something—and passive feedback—the system or certain functions are not used or are only used by certain groups of people. This step is particularly important for AI-based systems, as the systems and their functionality may change again.

Ideally, possible feedback channels for users should already be planned during the design phase and set up accordingly. It should also be clarified how passive feedback, e.g., the non-use of certain functions, can be collected during runtime.

Long-Term Availability

Industrial processes often have a long-term perspective on their capital investment. It is not uncommon for them to go even beyond 10-year usage scenarios. Looking at the fast progress in AI technology, can a state-of-the-art ML model from today still be run 10 years into the future? Is the hardware (e.g., GPU) still available and supported? The same questions also arise for the availability of human experts.

Availability of Technology In the rapidly evolving landscape of technology, both software and hardware can quickly become outdated. This poses challenges for AI solutions that rely on specific software frameworks, hardware architectures, or proprietary technologies. To address this issue, building on (Open) standards is increasingly becoming a viable approach, and it is expected to become more common in the future.

Standards such as OPC UA for machinery communication, As-set Administration Shell (AAS) for semantic information modeling of digital twins, and the Open Neural Network Exchange (ONNX) format for model artifacts provide specifications for interoperability and compatibility among different components and systems. By adopting these standards, solution providers can future-proof their AI solutions and mitigate the risks associated with technology obsolescence. Implementations of these standards may vary, as different vendors and developers may offer their interpretations or extensions. However, adhering to a standard ensures that even if a particular solution provider discontinues their products or services, the AI solution can still function seamlessly with alternative compatible implementations.

Building on (Open) Standards offers several advantages. It promotes vendor independence, allowing organizations to switch solution providers or integrate new technologies without significant disruptions. It also fosters collaboration and innovation by enabling the exchange of models, data, and tools between different systems. Moreover, standards facilitate the development of ecosystems and marketplaces where solution providers can offer compatible products and services. However, open standards and open-source software need to be maintained. Many companies thus rely on third parties for support of open-source packages.

Preservation of Development Artifacts After deployment, industrial applications that are operational without regular maintenance effort tend to get out of focus. Years after the initial development, changes to the overall system may require to access development artifacts are now lost to time and organizational changes. Re-engineering of industrial control units to overcome lost development sources is not uncommon. But in case of AI-based systems the re-engineering can be much more costly. Up to repeating the entire development cycle including the generation and curation of dedicated training data. For that we recommend the preservation of development artifacts across the full development cycle using versioning approaches for data, models and target systems (see section 3) and by documenting the development process.

6. Regulation Compliance

Many industries are subject to legal regulations and standards. In addition, general regulations for AI methods have appeared. This needs early consideration in a development project, because compliance of AI-based systems can be a large effort and not all AI methods are suited for it.

Industry-Specific Regulation

AI certification has received increasing attention in recent years. There are several institutions that push their own certificates on the market. These certificates do not meet any regulatory or legal requirements but refer to a confirmation of the compliance with certain requirements published by the issuing organization. [13] provide a detailed overview of the topic.

There are several norms, that are applicable for AI systems. The most important ones are the emerging AI norms which are provided by the committee ISO/IEC JTC 1/SC 42. Those norms deal with various aspects including risk management, data, transparency, bias, testing of AI systems and many more. At the time of this publication there are 25 Published ISO standards and 32 ISO standards under development by the committee.

Non-AI-specific norms worth mentioning are IEC 61508—a standard for functional safety of electrical, electronic and programmable electronic safety-related systems—and ISO 21448, addressing the safety of advanced driver-assistance systems and autonomous vehicles focusing on the potential hazards arising from the insufficiencies of the intended functionality.

AI-Specific Regulation

EU AI Act The EU AI Act [4] sets requirements on AI systems based on the risk of the use case. AI systems are categorized into four risk classes as shown in Figure 10: unacceptable risk, high risk, limited risk, and minimal risk. AI systems with unacceptable risk are outright banned. AI systems with minimal risk are not subject to any special requirements. AI systems with limited risk are subject to transparency obligations. AI systems with high risk need close examination regarding the AI Act. These systems may be implemented but are subject to a conformity assessment and must meet the requirements formulated in Articles 9-15 of the AI Act. Table 2 gives an overview of these requirements and links corresponding chapters of this whitepaper. For General purpose AI (GPAI) models the AI Act distinguishes between GPAI model provider and GPAI system provider. GPAI model providers have obligations formulated in Articles 53, 55 and Annex XI of the AI Act, and GPAI system providers must conduct the risk assessment and comply with the corresponding obligations.

US Executive Order and AI Bill of Rights The Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence issued on October 30, 2023 by the Biden administration covers principles of safety and security, privacy, civil rights, consumer and worker protections, innovation and competition, and national security. It directs the creation of best practices for development and deployment and regulatory guidance or requirements for critical AI uses. The Blueprint for the US AI Bill of rights (2022) contains five principles, regarding the design, use, and deployment of AI systems: Safe and Effective Systems, Algorithmic Discrimination Protections, Data Privacy, Notice and Explanation, Alternative Options. The principles are not-binding but intended as guideline.

General Regulation Requirements for AI

Traceability Traceability is an essential aspect of operationalizing machine learning systems. It involves documenting various aspects over the lifespan of an AI solution, including data, development, and operations. To ensure traceability it starts with documenting the journey of the data in the machine learning process. This includes information about the data sources, data preprocessing steps, and any data transformations applied. Documenting the development process involves recording the choices made during model selection, feature engineering, and parameter tuning. Additionally, documenting operations entails keeping track of the deployment environment, system configurations, and any changes made during the operational phase. A detailed method for documenting an AI solution, taking into account the requirements of the EU AI Act, is given by [2].

Documentation plays a vital role in maintaining transparency and accountability throughout the lifecycle of an AI solution. It enables stakeholders to understand the decision-making

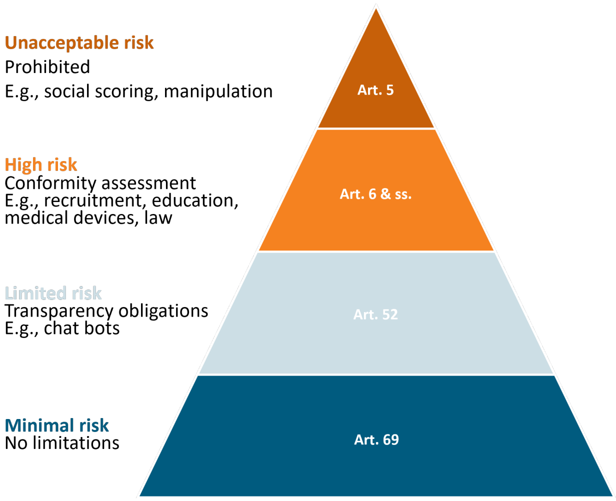


Figure 10: Overview of EU AI Act risk classes.

AI Act requirement (Art. 9-15)	Explanation	Corresponding chapters
Risk management system	Measures to minimize and deal with risks during the entire AI systems lifecycle	Not addressed
Data and data governance	Focusing on the quality of data and the absence of bias	Absence of Discrimination
Technical documentation	Comprehensive technical documentation	Traceability
Record-keeping	Logging of events during operations	Historizing and Versioning
Transparency and provision of information to users	AI systems need to be transparent and provide information of its usability and documentation	User Acceptance, Human-Centered quality, User Documentation and Training
Human oversight	Enabling of human interaction and provision of human-machine interface tools	Autonomy Level
Accuracy, robustness and cybersecurity	AI systems must achieve an appropriate level of accuracy, robustness and cybersecurity and perform consistently in those respects throughout their lifecycle	Model Performance, Model Robustness, IT-Security

Table 2: Overview of EU AI Act requirements, explanation and relevant chapters in this document that address the requirement.

process, identify potential biases, and address concerns related to fairness, ethics, and compliance.

Absence of Discrimination Bias-free AI is essential to ensure fairness, equality, and ethical decision making. Unwanted bias in AI systems can lead to inaccurate or unreliable results, discriminatory outcomes, and perpetuate existing social biases or inequalities and can lead to a lack of trust in the AI system and AI in general. It is necessary to consider the motivation of the creator and users and the potential harm of bias.

An compelling example are medical applications. Research has shown, that AI systems can outperform physicians in specific tasks, e.g., in detecting breast cancer through mammography screening [11]. While the potential of AI is huge, it can easily lead to undesirable outcomes if a bias is present.

Consider an AI system, that is being installed to decide whether a patient is suitable for an artificial hip. Making an informed decision requires comprehensive knowledge of a patient's medical record and current health status needs to be considered. To get the necessary information the data will likely be collected from multiple sources like physicians, health history and current health status. We want to end up make the best decision for the patient. Therefore, we must carefully consider the data sources and their potential biases. Since the health insurance is interested in paying the least amount of money possible, their data of past hip replacements may contain a bias that leads to the decision that an elderly patient is not suitable for a hip replacement, even though there is no medical reason to supports this statement.

In summary, avoiding bias in AI systems requires careful data collection, unbiased model development, and ongoing monitoring and evaluation of the AI system to identify and correct any biases that may arise.

AI Model Cards

AI model cards are a structured description of a model's technical properties and additional background information such as usage rights. Model cards are commonly used for the evaluation and selection of models originating from a different developer group, such as reusable foundation models.

A detailed method for using model cards to document an AI solution and approaching regulatory compliance, especially considering the requirements of the EU AI Act, is given by [2]. The method considers the entire lifecycle of AI systems using four cards (use case, data, model, operation).

Data Protection If personal data is used for an AI system, the organisation must comply with the General Data Protection Regulation (GDPR). The GDPR sets rules on how personal data is collected, processed, and stored. Core principles include obtaining consent, providing transparent information, ensuring data security, and respecting individual rights. Therefore, before an AI project begins, it should be verified whether personal data will be processed and, if so, whether the GDPR applies; consultation with the works council and other stakeholders is advisable.

Privacy-by-design principles should guide every stage of development and deployment, incorporating data minimisation, anonymisation, and pseudonymisation wherever feasible to safeguard individuals' privacy.

Finally, documentation of all processing activities is essential for demonstrating compliance, including the legal basis, data categories, and retention periods. Organisations must also be ready to meet data-subject requests such as access, rectification, or deletion without delay.

Solution Recommendations

Different from ad-hoc prototype development, we make recommendations for the development of AI-based systems for operational use in industry: i) define explicit requirements for the long-term operational use ii) use a visual representation of the overall system using a data-pipeline viewpoint to communicate with stakeholders, iii) follow a defined process for the project management that includes the management of data availability, and iv) build solutions on top of an established technical platform.

Explicit Requirements

The requirements encountered during long-term operations need to be anticipated during development. Agile development practices aim at creating a usable first version early on and to iterate based on stakeholder feedback. In the development of AI-based solutions for industry, it may however happen that a system can only go operational in the target environment when development is already in an advanced stage. This can limit the possibility for direct user feedback. This situation has the increased risk that some requirements are encountered at a stage where fundamental changes to the overall system have become very costly.

> **Our recommendation** *is to use the requirement categories from this document to make the requirements for the long-term operational use explicit early on during development.*

The following are common examples of missed requirements that hinder long-term operational use. First, consider that an AI system is integrated into a cyber-physical system, such as an automated production line. The system environment however changes its behavior over time. For example by wear and tear, component replacement and parameter tuning of the system operators. It might become necessary to update AI-based models to the new reality which is not reflected in the original training data. This results in requirements, such as the continued availability of data and access to maintenance interfaces of the AI-based solution. Second, the system operator personnel might not trust in the performance of an AI-based solution or even perceive it as hostile to their personal status as operations experts. Having an AI that is able to explain its decisions (and therefore allow human overrides when the operator has additional context) can increase the acceptability. This has a large impact on the maintenance and upkeep of the AI-based system – which is often directly influenced by the motivation of the operators.

Visualize Data Flows: From Concept to Code

An AI pipeline concept which takes into account all data sources and requirements on the application, enables reliable project planning. An example of such a pipeline is shown in Figure 11. The aim is to draw the data processing pipeline of the whole system in order to create awareness about the necessary modules and stakeholders that have to be involved. An excerpt of important questions that should be considered in the pipeline concept are listed in the box above.

> **Our recommendation** *is to use a visual representation of the data flows in the overall AI system to effectively communicate with stakeholders.*

Answering those questions in the first phase of the project will probably raise many more questions but will help to keep an eye on all the important aspects. It is advisable to follow an established AI process model which structures and coordinates the development process and already incorporates many of these aspects.

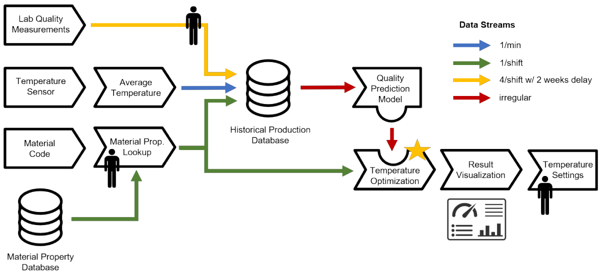


Figure 11: Example machine learning pipeline diagram [17].

During development of the AI system, the AI pipeline concept is extended by many details and realized on a technical level. MLOps frameworks will help to implement AI pipelines, manage and operate different versions of AI models that will be trained during the system's lifecycle.

Project Management Process

Process models guide through certain processes such as managing a project. There are various existing process models, especially for general project management. A fundamental characteristic is the focus of process models: organizational vs. content-based. Organizational process models focus on the labor organization, including topics such as communication, meetings and the fundamental type of project organization (plan-driven vs. iterative vs. agile). Content-focused process models guide through the process with recommendations on the contentual topics and their interdependencies. AI projects are a specific type of IT projects with a higher level of uncertainty and thus, the usage of an agile or at least iterative organizational approach is recommended. This does not mean, that milestones and deadlines should be avoided, but that planning should include possibly relevant iterations with the accompanying resource requirements. Besides that, several new aspects come with AI projects, which can be covered by content-focused process models, integrating known project steps with adaptions and supplements from the domain of AI.

> **Our recommendation** *is to use an explicit and continuous process model for AI project management. If possible, a model with concrete tool recommendations for the (new) challenges of AI should be used.*

When deciding on a process model to use, there are several meta-characteristics that should be taken into account [14]. The continuity is one of the most important of such characteristics. It means that a process model starts with the beginning of a project (including idea development and project setup) and goes up to the deployment in practice, including affected humans, processes, and the technology. Many process models only take into account completely new content, such as exploratory data analysis and model building. This is not wrong, but bears the risk of forgetting relevant aspects of a project, effectively reducing the success chances. A common problem according to our experience is, that topics like system architecture, data architecture and integration are often forgotten, which may lead to silo systems. Such systems usually sum up technical and organizational debts, which result in long-term costs. Another very important characteristic is the presence of concrete tool recommendations. It is nice to know which problems and tasks might arise, but even better when a way to handle them is provided. The focus on pragmatic aspects of real-world problems is important, too, since theoretical optimums tend to be far too extensive and costly for application in practice.

A recent survey of data science process models has investigated available models with a focus on continuity and the presence of tool recommendations [16]. The results show that all observed models have major gaps, especially in the early and late phases of project implementation. This includes the current industry standard, the CRoss Industry Standard Process for Data Mining (CRISP-DM) [3], as well as several newer models.

AI projects are interdisciplinary and rely on clearly defined roles across all phases of a project. A combination of data scientists, technical expertise, and many more are usually part of AI projects. Early planning should assign specific tasks or roles to named individuals, teams, or partners to avoid delays, extra costs, or failure. AI process models should include all key roles to ensure nothing is overlooked, especially the AI-specific ones like data officers and AI experts.

Two recent models address the discussed challenges and are briefly described in the following in addition to CRISP-DM in its role as the current industry standard.

CRISP-DM The CRISP-DM was published in 2000 for data mining projects. It focuses primarily on understanding, exploring, and modeling data (a detailed gap analysis can be seen in [16]). The subsequent evaluation of data is aligned with the business understanding, and the outcomes determine whether to initiate the modeling cycle again or deploy the model. CRISP-DM assumes the presence of data, but does not take into account data sources, such as system components that can influence both the data and the model. Consequently, the process model lacks the capability to articulate the interaction between data quality and the functionalities of the ML component.

DSPG The Data Science Project Guide (DSPG) is a model developed based on the previously mentioned survey to close the gaps in continuity and tool recommendations. The whitepaper is published under the non-restrictive license CC-BY-SA [15] and it is concisely written for the application in practice. It structures AI projects in four phases as a reference structure: 1) Goals and Requirements, 2) Structured Project Setup, 3) Concepts and Implementation, and 4) Utilization of the Results. These four phases come with 21 project steps which represent the contents of a data science or AI project. The contents are provided with short descriptions, key questions and concrete tool recommendations. By this design, known shortcomings of other models such as the CRISP-DM are addressed by the DSPG. The model is meant as a reference model—in practice, there will always be small changes such as a reordering of project steps or skipping some.

PAISE® The Process Model for AI Systems Engineering, known as PAISE®, tackles challenges of developing AI-based systems by integrating strategies from systems engineering, software development, and data science. It consists of seven phases with the fifth phase, the development cycle, as its core. The fundamental methodology involves incrementally developing the entire system, consisting of hardware, software and AI components. Additionally, the role of datasets, being vital for the development of AI components, is explicitly addressed. Datasets are treated as own components with individual requirements and own development processes. The decomposition of the overall system facilitates the parallelization of domain-specific development processes. Concurrently, interdisciplinary checkpoints are employed to test component dependencies, leading to the refinement of component specifications and solution approaches.

The selection of a suitable process model for data science and AI projects may vary based on requirements and viewpoints which are usually dependent on the use case, the enterprise and the level of knowledge of the project members. Table 3 shows some relevant aspects for the decision about a process model to be used. A content-focused perspective can additionally be helpful and can be found in form of a gap analysis in [16].

	CRISP-DM [3]	DSPG [15]	PAISE® [9]
Data acquisition considered	✗	✓	✓
Explicit elicitation of requirements	✓	✓	✓
System development besides model training	✗	✓	✓
Planned interaction with other engineering disciplines	✓	✓	✓
Change management and stakeholder considerations	✗	✓	✗
Differences between prototype and operational development	✓	✓	✓
Operation and maintenance after deployment	✓	✓	✓
Iterative (I) or linear (L) process models	I	I	I / L

Table 3: Comparison of AI Development Process Models.

MLOps Platforms

AI systems and the pipelines involved usually get very complex on a technical level. In navigating the complexities of keeping machine learning systems continuously operational, practical experience reveals the difficulty of organizing seamless collaboration between data scientists, DevOps teams, and other stakeholders. Addressing this challenge is crucial to maintaining a consistent development and operation pipeline for machine learning models and their integration into machine learning target systems.

> **Our recommendation** *is to use established MLOps platforms as the basis for AI development and deployment in productive use.*

MLOps platforms have emerged as a key tool for managing the end-to-end lifecycle of machine learning models. These platforms enable seamless interdisciplinary collaboration between data scientists, developers, and operations teams, facilitating the integration of machine learning into business processes.

MLOps platforms offer a range of key features and benefits. They provide provide control, streamlined deployment blueprints, automatically deployed real-time monitoring with drift detection, elastic scalability, and centralized, traceable management that integrates directly with brownfield services like storage and CI/CD services.

When deciding on a MLOps platform, the following categorization gives a starting point for the consideration of what type of platform meets the requirements:

- **All-in-one suites**: comprehensive hyperscaler solutions, as MLOps emerged from the IT domain, that accelerate time-to-value but introduce ecosystem lock-in.
- **Topic-specific tools**: best-of-breed components offering deep capabilities in deployment, version control, or orchestration, yet demanding custom integration.
- **Open-source stacks**: highly flexible, community-driven toolkits adaptable to any requirement, but require detailed tool knowledge for setup and ongoing maintenance.
- **Custom solutions:**: tailored platforms delivering pinpoint functionality and meet unique organizational needs at the cost of dedicated in-house development and slower feature evolution.

MLOps – A Process Model?

The Machine Learning Operations (MLOps) discipline, often referred to as ”DevOps for ML,” offers a process model and methodological framework for effectively managing the lifecycle of machine learning solutions. Therefore MLOps treats ML models with the same rigour as software artifacts: models, data and code are all version-controlled, tested and released through CI/CD pipelines. Coined in 2015 to curb the technical debt that stalls many prototypes, MLOps frames a maturity ladder (popularised by Google) that moves organisations from ad-hoc scripts and model development to fully automated, end-to-end pipelines accompanied by matching organisational change. By covering every stage in the lifecycle of AI projects, from data preparation, model training, deployment, real-time monitoring and continuous improvement, MLOps turns prototypes into scalable, reliable ML products with predictable operating cost.

Beyond this core and basic type of decision, further considerations affect a platform decision. Select a platform that scales with data and compute growth, is easy to use and extend, protects sensitive data and models, supports multiple languages and frameworks, and comes with strong community backing, reputable vendors, and transparent pricing.

Real-world case studies and success stories demonstrate the effectiveness of MLOps platforms in improving the operationalization of AI solutions. These examples showcase how organizations have overcome challenges and achieved improved efficiency, reliability, and time-to-market by implementing MLOps platforms. Currently, the market is nascent and lacks standardized frameworks. Therefore, an established, comprehensive platform serves as a pragmatic starting point. As expertise grows and practical needs become clearer, a shift to more tailored or custom solutions might prove more cost-effective than investing entirely in an end-to-end platform. However, planning for a future shift warrants thoughtful platform selection due to the considerable effort involved in transitioning without interoperable data models and standards in place.

References

[1] Albert Bifet and Ricard Gavaldà. “Learning from Time-Changing Data with Adaptive Windowing”. In: *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, pp. 443–448.

[2] Danilo Brajovic et al. “Model Reporting for Certifiable AI: A Proposal from Merging EU Regulation into AI Development”. In: *arXiv preprint arXiv:2307.11525* (2023).

[3] Pete Chapman et al. *CRISP-DM 1.0: Step-by-step data mining guide*. 2000.

[4] European Commission. *Proposal For A Regulation Of The European Parliament And Of The Council Laying Down Harmonised Rules On Artificial Intelligence (Artificial Intelligence Act) And Amending Certain Union Legislative Acts – COM/2021/206 Final*. 2021.

[5] F. Davis. “A technology acceptance model for empirically testing new end-user information systems - theory and results”. PhD thesis. Massachusetts Inst. of Technology., 1985.

[6] *Ergonomics of human-system interaction. Part 210: Human-centred design for interactive systems*. Standard 9241-210. International Organization for Standardization, 2019.

[7] *Ergonomics of human-system interaction. Part 220: Processes for enabling, executing and assessing human-centred design within organizations*. Standard 9241-220. International Organization for Standardization, 2019.

[8] Constanze Hasterok, Jan Hermes, and Benedikt Stratmann. “SiD2Re - A novel simulation framework for drifting regression data”. In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. 2023, pp. 1–8.

[9] Constanze Hasterok and Janina Stompe. In: *at - Automatisierungstechnik* 70.9 (2022), pp. 777–786.

[10] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[11] David Killock. “AI outperforms radiologists in mammographic screening”. In: *Nature Reviews Clinical Oncology* 17.3 (2020), pp. 134–134.

[12] Christian Knecht, Christian Saba, and Bastian Pokorni. *Gestaltung von KI-Systemen - Akzeptanzförderliche Mensch-KI-Zusammenarbeit am Beispiel eines Assistenzsystem*. 2024.

[13] Janika Kutz et al. “KI-Zertifizierung und Absicherung im Kontext des EU AI Act”. In: *Fraunhofer-Publica* (2023).

[14] Damian Kutzias, Claudia Dukino, and Holger Kett. “Towards a Continuous Process Model for Data Science Projects”. In: *Advances in the Human Side of Service Engineering*. Vol. 266. Springer International Publishing, 2021, pp. 204–210.

[15] Damian Kutzias, Claudia Dukino, and Jan-Paul Leuteritz. *Leitfaden zur Durchführung von KI-Projekten: Menschenzentrierung von der Idee bis zur Anwendung*. Ed. by Oliver Riedel et al. Stuttgart, 2023. DOI: `10.24406/publica-1637`.

[16] Damian Kutzias et al. “Comparative Analysis of Process Models for Data Science Projects”. In: *Proceedings of the 15th International Conference on Agents and Artificial Intelligence* 3.1 (2023).

[17] Julius Pfrommer et al. *ML4P-Vorgehensmodell Machine Learning for Production*. Tech. rep. Fraunhofer, 2022. URL: `https://publica.fraunhofer.de/entities/publication/e2e0d3d0-fd26-4bfd-98ba-e6d465cb7d6c`.

[18] Plattform Industrie 4.0. *Technologieszenario „Künstliche Intelligenz in der Industrie 4.0“*. 2019. URL: `https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/KI-industrie-40.pdf`.

[19] SAE International. *SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2021.

About

AI Innovation Center

The AI Innovation Center "Learning Systems and Cognitive Robotics" supports companies in exploiting the economic opportunities offered by artificial intelligence and machine learning. In application-oriented research projects and direct cooperation with industrial companies, the Fraunhofer Institutes for Manufacturing Engineering and Automation IPA and for Industrial Engineering IAO in Stuttgart are working towards bringing technologies from cutting-edge AI research into widespread use in the manufacturing industry and the service sector. The Institute of Human Factors and Technology Management IAT at the University of Stuttgart supports them. The center receives financial funding from the Baden-Württemberg Ministry of Economic Affairs, Labour, and Tourism.

Mission

The AI Innovation Center is the application-oriented branch of Cyber Valley, Europe's largest research collaboration in the field of artificial intelligence. It is also part of S-TEC, the Stuttgart Technology and Innovation Campus: www.s-tec.de

It bridges the gap between state-of-the-art AI research and small and medium-sized enterprises, making AI technologies usable for the economy in Baden-Württemberg and beyond. As a leading innovation partner for small and medium-sized enterprises, the center works on topics that are of central importance for the use of AI and robotics across industries, such as autonomy, efficiency, sustainability, human-machine interaction, and trust.

The AI Innovation Center informs companies about technology trends as well as their potential applications and provides them with low-threshold, needs-based support in the development and implementation of ambitious AI innovations so that they can make even better use of the economic opportunities offered by AI in the future.

Vision

The AI Innovation Center is a beacon for successful technology transfer to small and medium-sized enterprises and enables companies to use artificial intelligence and robotics economically and responsibly for business success and individual and social benefit.

Study Series "Learning Systems and Cognitive Robotics"

The study series "Learning Systems and Cognitive Robotics" provides insight into the potential and practical applications of AI. Further information and the latest versions of the studies can be found at the website of the AI Innovation Center: https://www.ki-fortschrittszentrum.de/studien.

CC-KING Competence Center

The Competence Center Karlsruhe for AI Systems Engineering (CC-KING, https://www.ki-engineering.eu) is a collaborative initiative spearheaded by three premier research institutions: the Fraunhofer Institute for Optronics, System Technologies and Image Exploitation (IOSB), the FZI Research Center for Information Technology, and the Karlsruhe Institute of Technology (KIT). Funded by the Ministry of Economics, Labour and Tourism of Baden-Württemberg, CC-KING bridges cutting-edge AI and ML research with established engineering disciplines to streamline the deployment of intelligent systems in real-world industrial and mobility contexts.

Mission and Methodological Foundations

KI-Engineering emphasizes AI Systems Engineering, a systematic framework for developing and operating AI-based solutions embedded within larger, complex systems. Unlike traditional engineering—where component behavior is predictable and well-defined—AI systems introduce runtime-dependent behaviors shaped by data and learning. CC-KING addresses this challenge by formulating methodologies that ensure predictability, functional safety, explainability, and certification readiness of AI and ML components from the design phase onward.

Applied Research, Tools, and Industry Transfer

At its core, CC-KING combines rigorous research with practical transfer mechanisms. It develops tools and process models such as the PAISE (Process Model for AI Systems Engineering) and the PAISE Toolkit—which integrate linear, system-level planning with agile, subsystem-oriented AI development. Real-world testing occurs in authentic lab environments: the Karlsruhe Research Factory for AI-integrated production and the Test Area Autonomous Driving Baden-Württemberg. CC-KING also actively supports SMEs via QuickChecks, TransferChecks, workshops, and an AI Systems Engineering learning lab, making sophisticated AI-engineering practices accessible to more organizations.

AI Beyond the Prototype – Requirements One Pager

1. Autonomy Level

Requirement	*Requirement Priority*	*Requirement Description*
Autonomy Level	☐ High ☐ Mid ☐ Low	☐ Assistance Functionality ☐ Human in the Loop ☐ Human Supervision ☐ Full Autonomy

2. Performance

Requirement	*Requirement Priority*	*Requirement Description*
Model Performance	☐ High ☐ Mid ☐ Low	
Model Robustness (Noise, Outliers, Drift)	☐ High ☐ Mid ☐ Low	
Processing Time	☐ High ☐ Mid ☐ Low	☐ Milliseconds ☐ Seconds ☐ Minutes ☐ Hours ☐ Days

3. Supervision and Maintenance

Requirement	*Requirement Priority*	*Requirement Description*
Drift Detection	☐ High ☐ Mid ☐ Low	
Retraining	☐ High ☐ Mid ☐ Low	
Model and Data Versioning	☐ High ☐ Mid ☐ Low	

4. Integration and Deployment

Requirement	*Requirement Priority*	*Requirement Description*
Low Latency	☐ High ☐ Mid ☐ Low	
Integration with OT / automation equipment	☐ High ☐ Mid ☐ Low	
Compliance with company-wide cloud / deployment strategy	☐ High ☐ Mid ☐ Low	
Special hardware (Storage, GPU)	☐ High ☐ Mid ☐ Low	
Aggregate data from multiple deployments	☐ High ☐ Mid ☐ Low	
Remote access	☐ High ☐ Mid ☐ Low	
Data Protection, IT-Security	☐ High ☐ Mid ☐ Low	

5. Acceptability

Requirement	*Requirement Priority*	*Requirement Description*
Agency of the Operational Organization	☐ High ☐ Mid ☐ Low	
Usability	☐ High ☐ Mid ☐ Low	
Documentation and Training	☐ High ☐ Mid ☐ Low	
Explainability	☐ High ☐ Mid ☐ Low	
User Feedback	☐ High ☐ Mid ☐ Low	
Change Management	☐ High ☐ Mid ☐ Low	
Long-Term Availability of Technology	☐ High ☐ Mid ☐ Low	
Long-Term Availability of Human Experts	☐ High ☐ Mid ☐ Low	

6. Regulation Compliance

Requirement	*Requirement Priority*	*Requirement Description*
General AI System Regulation	EU AI-Act Risk Classification: ☐ High Risk ☐ Limited Risk ☐ Minimal Risk	
Functional Safety	☐ High ☐ Mid ☐ Low	
Traceability	☐ High ☐ Mid ☐ Low	
Absence of Bias	☐ High ☐ Mid ☐ Low	

Fraunhofer

Imprint

Fraunhofer Institute of Optronics, System Technologies and
Image Exploitation IOSB
Fraunhoferstraße 1, 76131 Karlsruhe

Fraunhofer Institute for Industrial Engineering IAO
Nobelstraße 12, 70569 Stuttgart

Fraunhofer Institute for Manufacturing Engineering and
Automation IPA
Nobelstr. 12, 70569 Stuttgart

Constanze Hasterok
constanze.hasterok@iosb.fraunhofer.de

Niclas Renner
niclas.renner@iao.fraunhofer.de

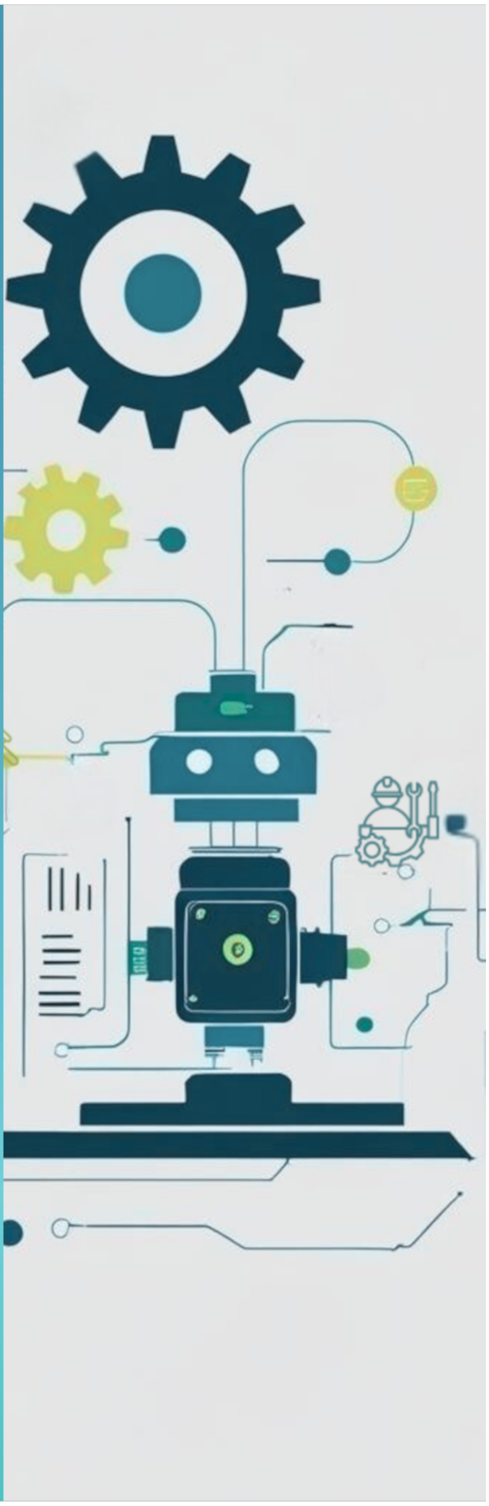
Lukas Rauh
lukas.rauh@ipa.fraunhofer.de

Fraunhofer-Publica
http://dx.doi.org/10.24406/publica-4825

Cover picture
Fraunhofer IAO

Funded by the Ministry of Economic Affairs,
Labour and Tourism Baden-Württemberg

CC-BY-NC-ND-License



Gefördert durch
Baden-Württemberg
Ministerium für Wirtschaft,
Arbeit und Tourismus

Cyber Valley

S·TEC

Universität Stuttgart
Institut für Arbeitswissenschaft und
Technologiemanagement IAT