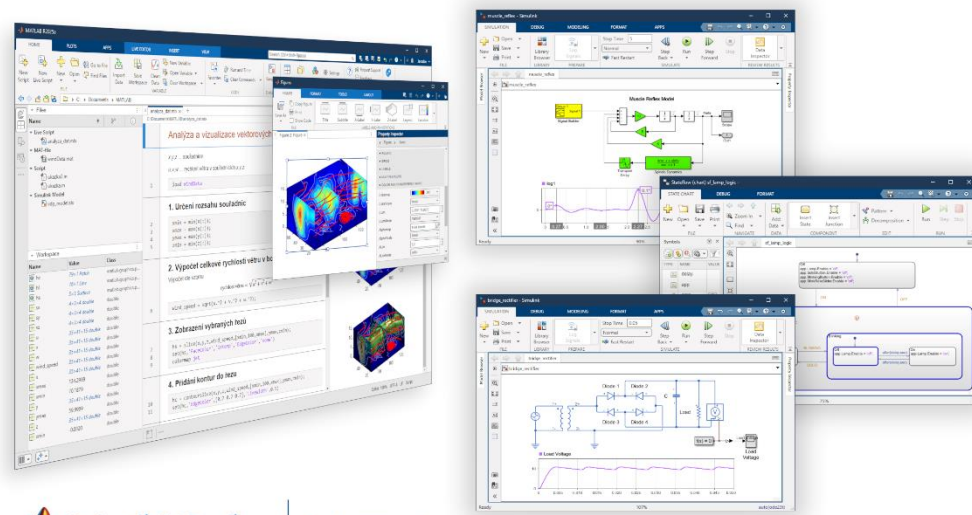


11.9.2025 Technical Computing Camp 2025

# Verifikace, komprese a implementace AI modelů na embedded systémy



Jaroslav Jirkovský  
jirkovsky@humusoft.cz

[www.humusoft.cz](http://www.humusoft.cz)  
[info@humusoft.cz](mailto:info@humusoft.cz)

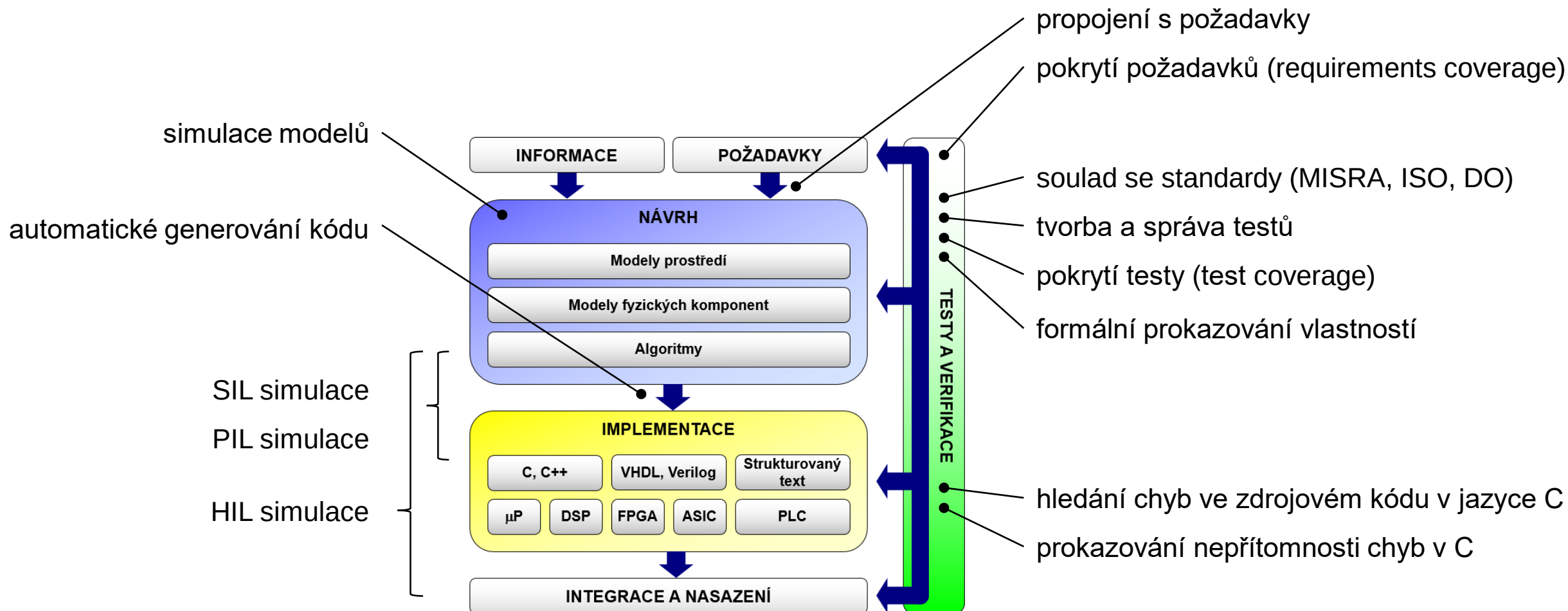
[www.mathworks.com](http://www.mathworks.com)



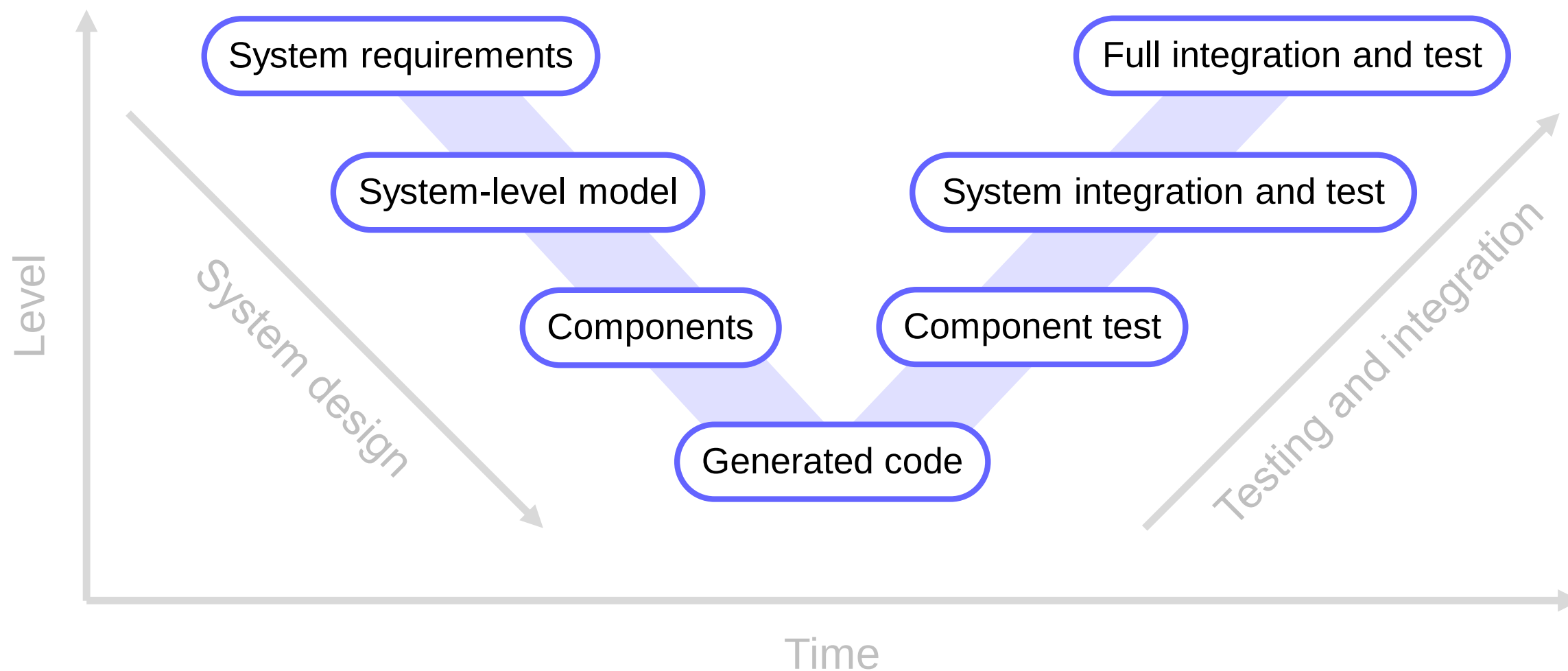
## Nasazení

# Vývoj metodou Model-Based Design

- Systematické využití simulačních modelů napříč vývojovým procesem

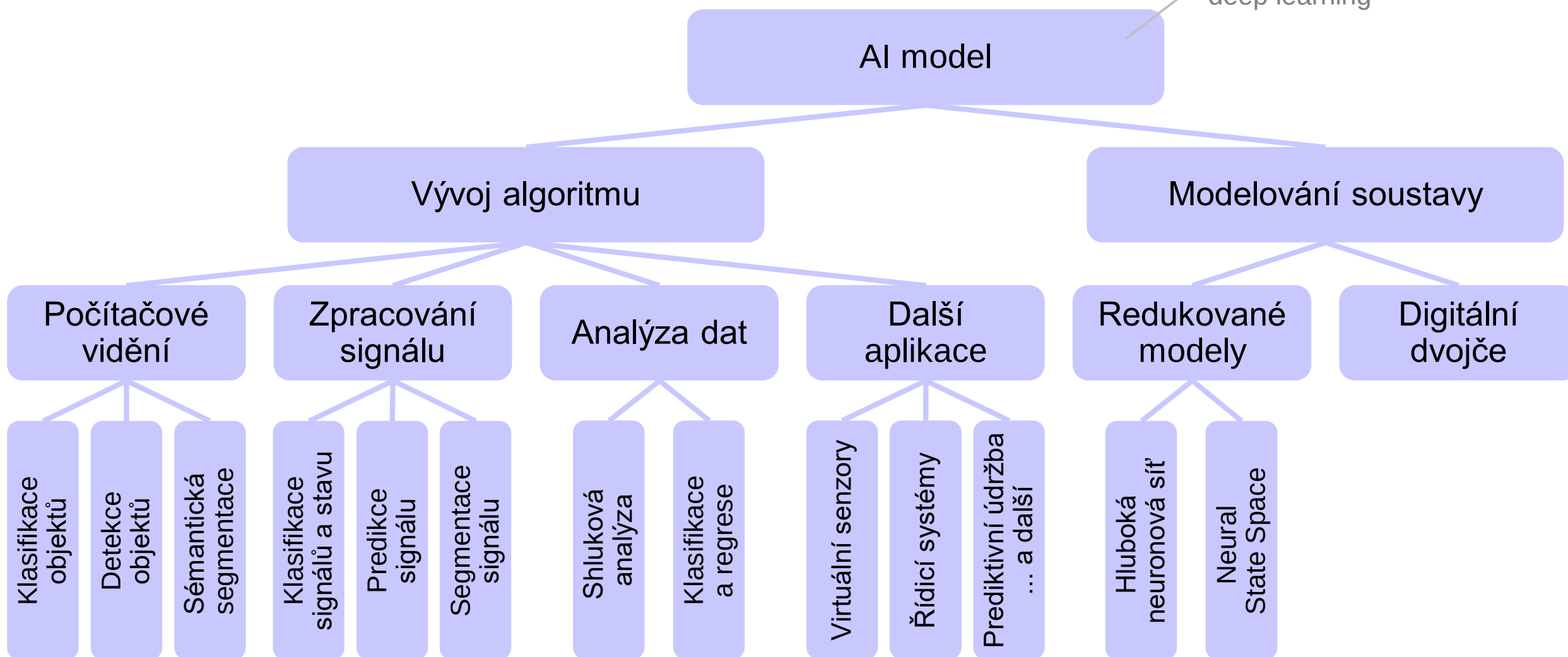


# Vývojový cyklus V pro standardní (ne-AI) embedded systémy



# AI modely a jejich využití

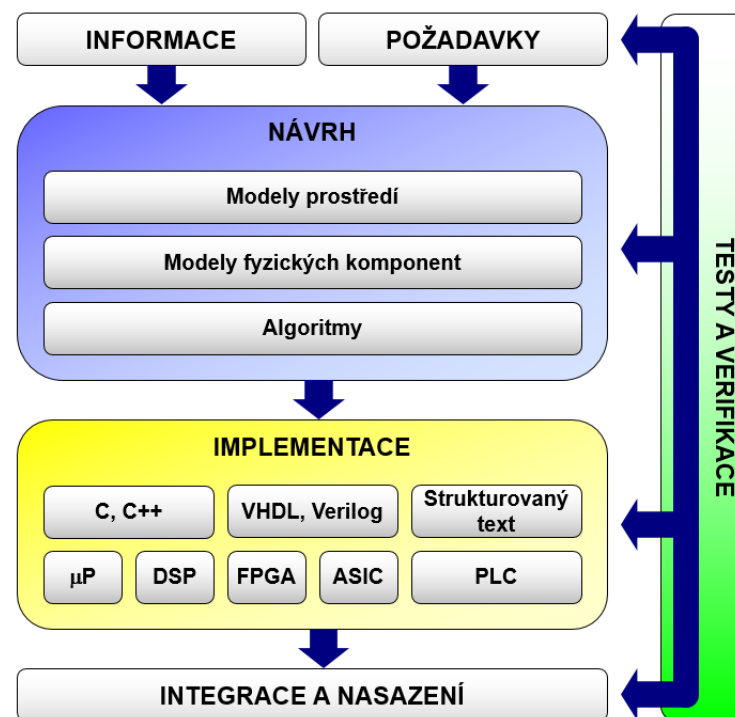
machine learning  
deep learning



# AI a Model-Based Design



+



# Physical AI: posun AI do reálných systémů

- Aplikace AI v systémech, které fungují a interagují ve fyzickém světě
  - posun umělé inteligence za hranice digitálního světa
  - systémy, které dokáží vnímat, rozhodovat se a jednat v reálném čase
- Na rozdíl od tradičních aplikací AI zahrnují systém s uzavřenou smyčkou
  - systém přímo ovlivňuje své okolí prostřednictvím aktuátorů (motory, mechanické součásti)
  - tyto akce následně mění sadu vstupů, které systém přijímá

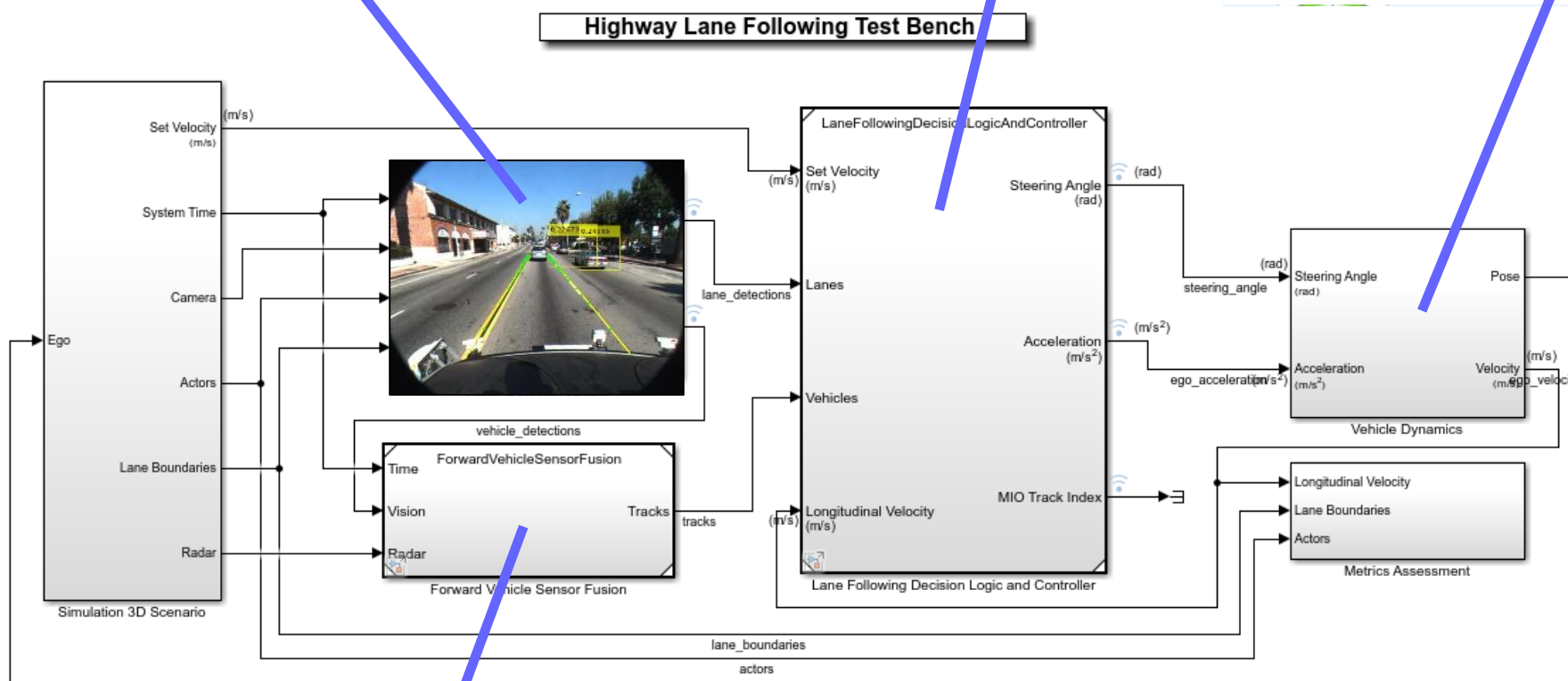


# Typický příklad: Asistenční systém automobilu

AI algoritmus  
detekce jízdních pruhů a vozidel

Řídicí systém

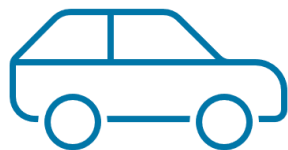
Model dynamiky vozidla  
pro testování chování systému



Senzorická fúze



# Předpisy a certifikace



## Automobilový průmysl

← ICS ← 43 ← 43.040 ← 43.040.10

### ISO/CD PAS 8800

Road Vehicles — Safety and artificial intelligence

#### Abstract

This document defines safety-related properties and risk factors impacting the insufficient performance and malfunctioning behaviour of Artificial Intelligence (AI) within a road vehicle context. It describes a framework that addresses all phases of the development and deployment lifecycle. This includes the derivation of suitable safety requirements on the function, considerations related to data quality and completeness, architectural measures for the control and mitigation of failures, tools used to support AI, verification and validation techniques as well as the evidence required to support an assurance argument for the overall safety of the system.

#### General information

Status : Under development

Edition : 1

Technical Committee : ISO/TC 22/SC 32 Electrical and electronic components and general system aspects

ICS : 43.040.10 Electrical and electronic equipment | 43.040.15 Car informatics. On board computer systems



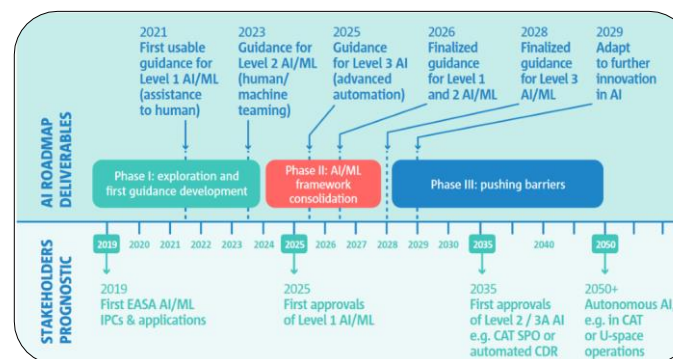
## Letectví

WIP 2023-06-26



### Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI ARP6983

This document discusses guidelines for the development of Aircraft Systems leveraging AI capabilities, taking into account the overall aircraft operating environment and functions. This includes validation of requirements and verification of the design implementation for certification and product assurance and guidelines with the assessment of safety. It provides practices for showing compliance with the regulations and serves to assist a company in developing and meeting its own internal standards by considering the guidelines herein.



European Aviation Safety Agency AI Roadmap



## Medicínská zařízení



Search

Menu

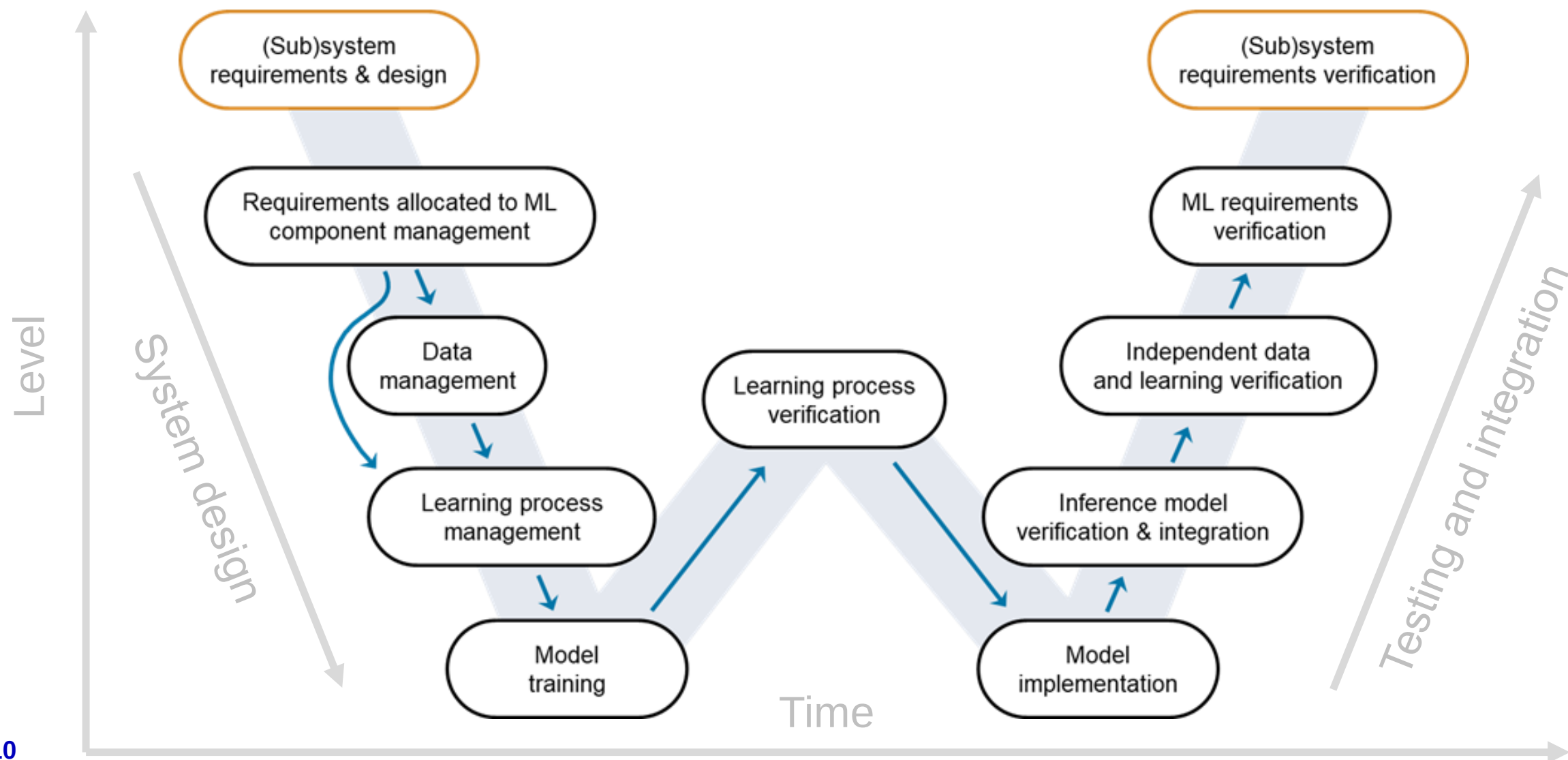
← [Software as a Medical Device \(SaMD\)](#)

## Artificial Intelligence and Machine Learning (AI/ML)-Enabled Medical Devices

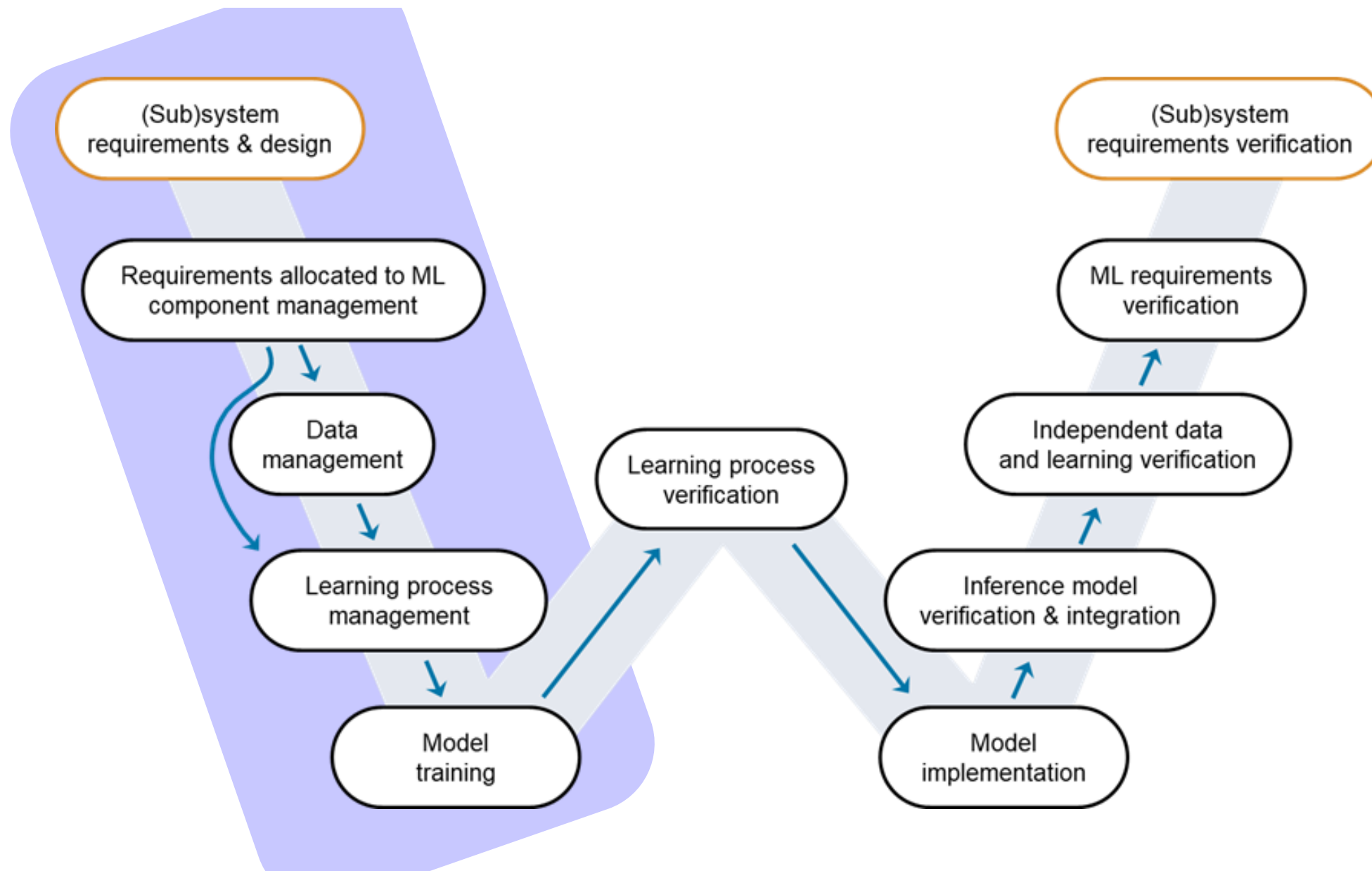
**October 5, 2022 update:** 178 Artificial Intelligence and Machine Learning (AI/ML)-Enabled Medical Devices were added to the list below. With this update, the FDA has also added the ability to download the list as an Excel file.

# Vývojový cyklus W: Adaptace V-cyklu pro AI

Credit: EASA, Daedalean.



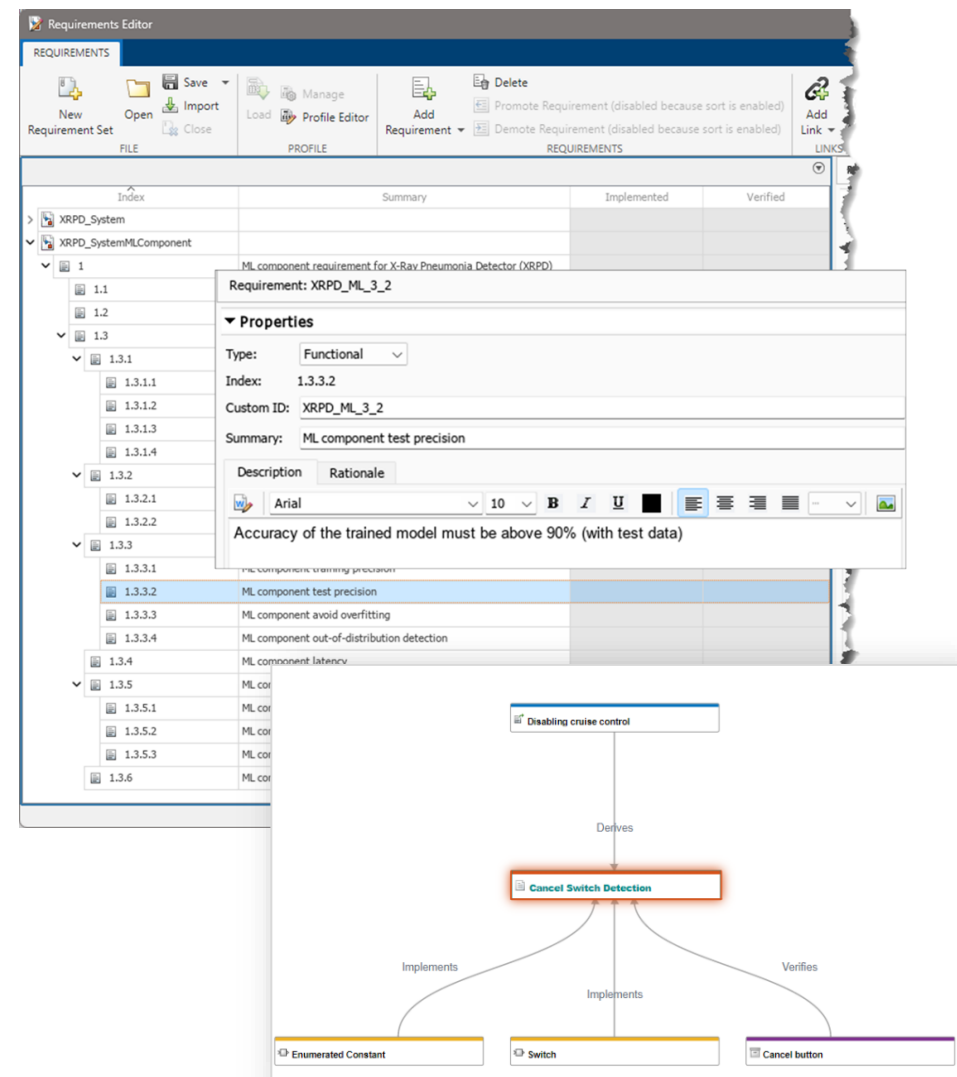
# Od požadavků k robustnímu modelu



# Správa požadavků



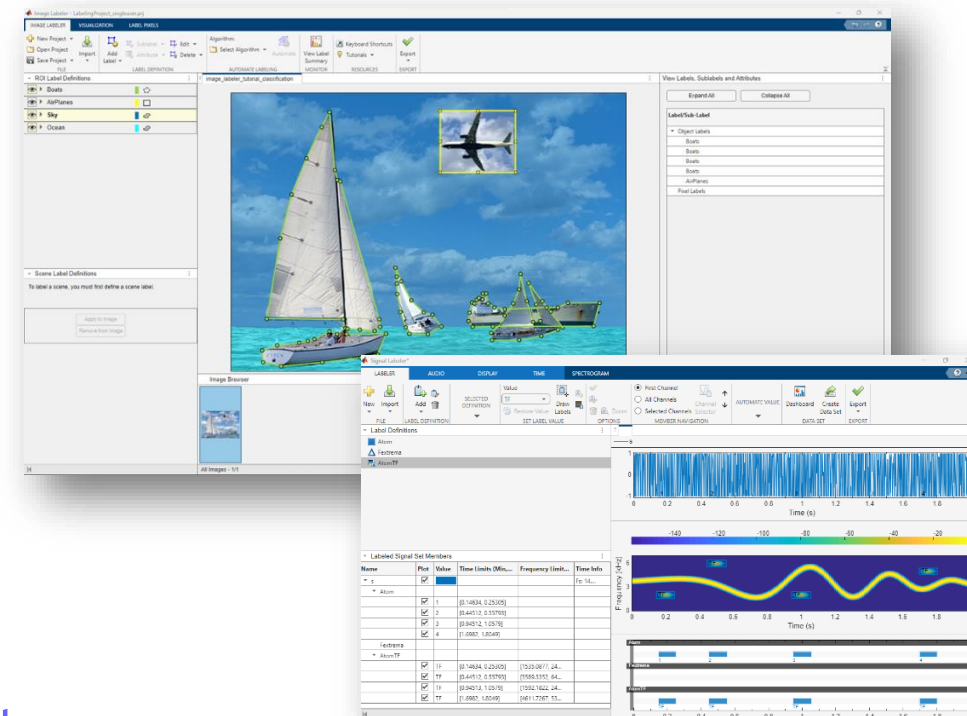
- Požadavky na celkový návrh systému
- Požadavky specifické pro AI komponenty
- Requirements Toolbox
  - vytváření, propojení a trasování požadavků
  - Requirements Editor
  - Traceability Matrix
  - Traceability Diagrams
- Klíčové otázky
  - jsou všechny požadavky implementovány?
  - jak budou požadavky testovány?
  - lze vysvětlit chování modelu?



# Správa dat



- Správa dat určených pro učení AI modelu
- Označování dat pro účely učení
  - Image Labeler, Signal Labeler
- Manipulace s daty
  - přímé načítání dat – pro malé datové sady
  - objekty datastore: imageDatastore, signalDatastore, ...
  - úprava dat: TransformedDatastore, augmentedImageDatastore, ...
- Rozsáhlá data
  - tall arrays



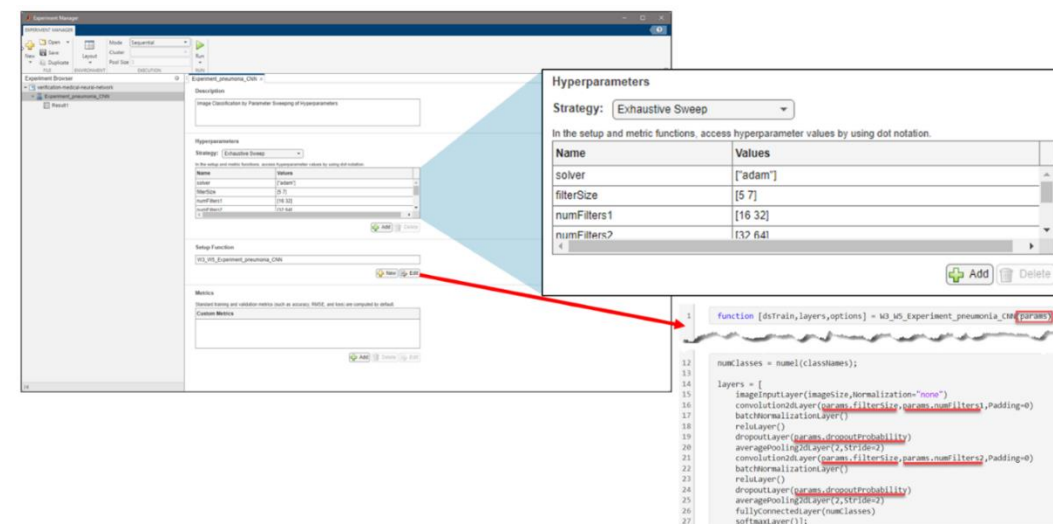
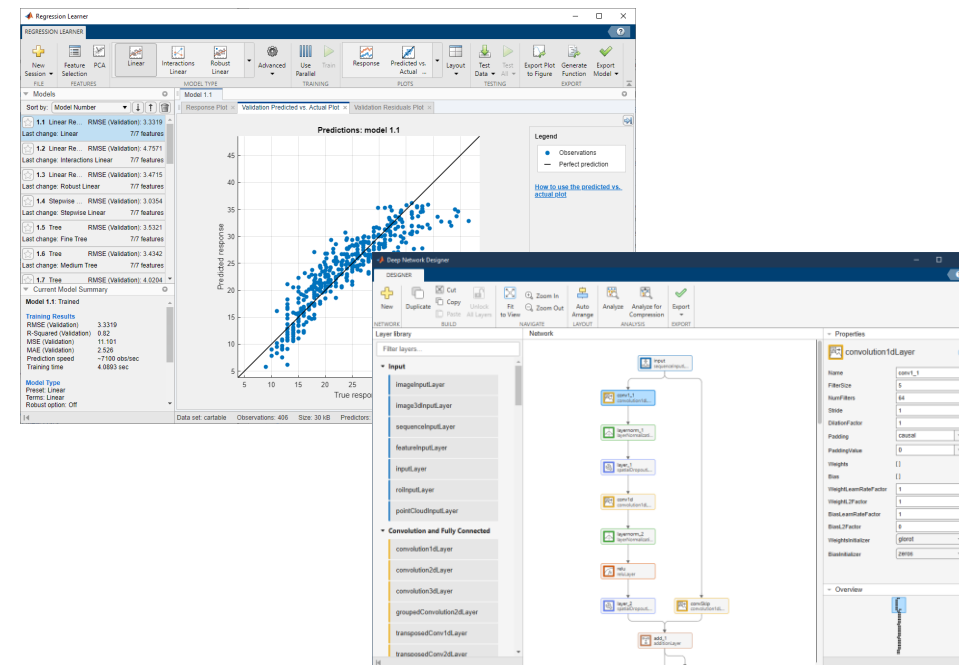
```

imds = ImageDatastore with properties:
    Files: {
        ...\matlab\toolbox\matlab\demos\example.tif';
        ...\matlab\toolbox\matlab\matlab_images\tif\corn.tif'
    }
    Folders: {
        ...\matlab\toolbox\matlab'
    }
    Labels: [demos; imagesci]
    AlternateFileSystemRoots: {}
    ReadSize: 1
    SupportedOutputFormats: ["png" "jpg" "jpeg" "tif" "tiff"]
    DefaultOutputFormat: "png"
    ReadFcn: @readDatastoreImage
  
```

# Správa procesu učení



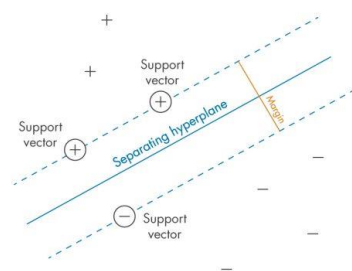
- Přípravná fáze před vlastním učením
- Výběr modelu / architektury sítě
- Nastavení možností pro učení
  - učící algoritmus, ztrátová funkce, hyperparametry
- Interaktivní nástroje nebo funkce
  - Deep Network Designer
  - Classification Learner
  - Regression Learner
- Hledání optimálních hyperparametrů
  - nastavení experimentů
  - Experiment Manager



# AI modely v prostředí MATLAB

Machine learning

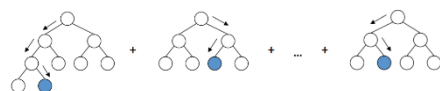
Deep learning



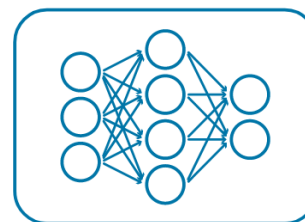
SVM



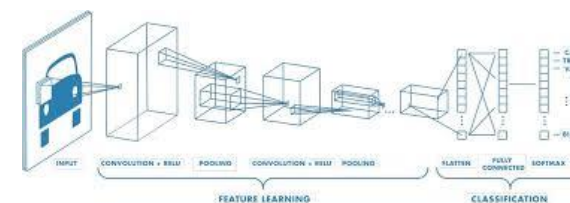
Clustering



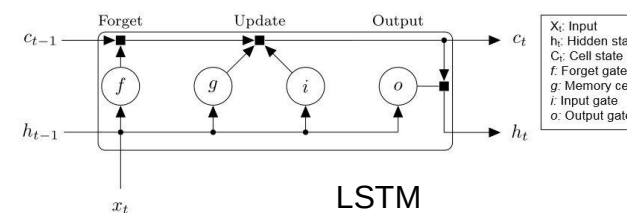
Decision trees



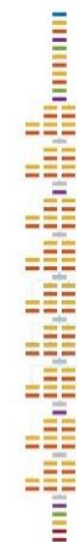
FC



CNN



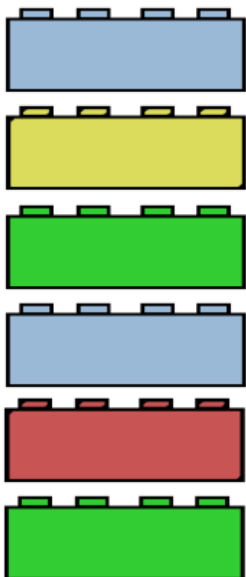
LSTM



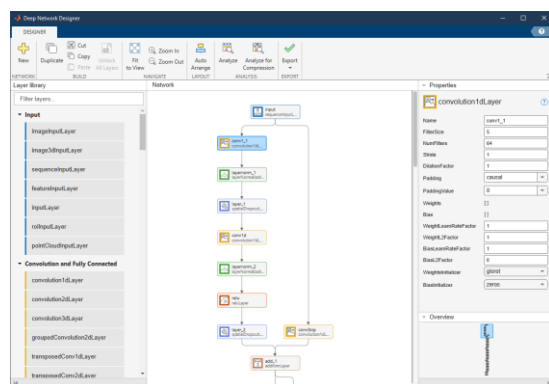


# Vytvoření hluboké neuronové sítě v prostředí MATLAB

~100 typů vrstev



Deep Network Designer



Připravené funkce  
(low-code přístup)

```
layers = [imageInputLayer(inputSize)
          convolution2dLayer(filterSize,numFilters)
          reluLayer()
          maxPooling2dLayer(poolSize)
          fullyConnectedLayer(outputSize)
          softmaxLayer()];
```

```
opts = trainingOptions('solver');
net = trainnet(data, layers, lossfcn, opts);
```

```
scores = minibatchpredict(net, newData);
label = scores2label(scores, classNames);
```

Přizpůsobení na míru

uživatelské smyčky pro učení sítí  
automatická diferenciace  
více vstupů a výstupů  
uživatelské ztrátové funkce  
...

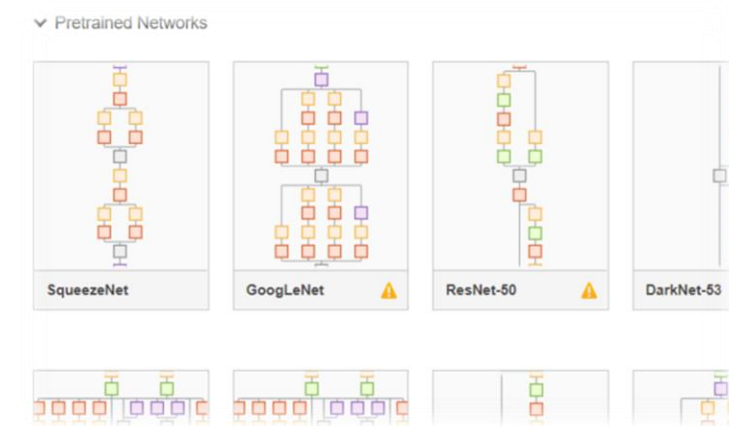
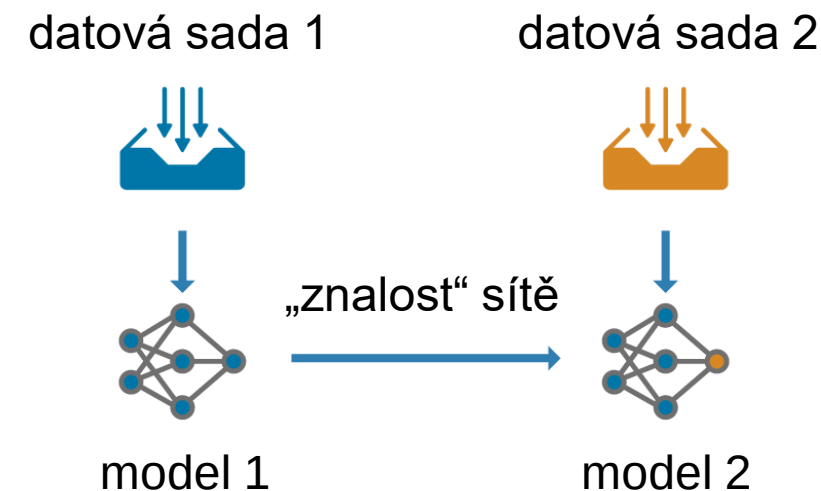
GAN, PINN, ...

*vhodné pro většinu úloh*



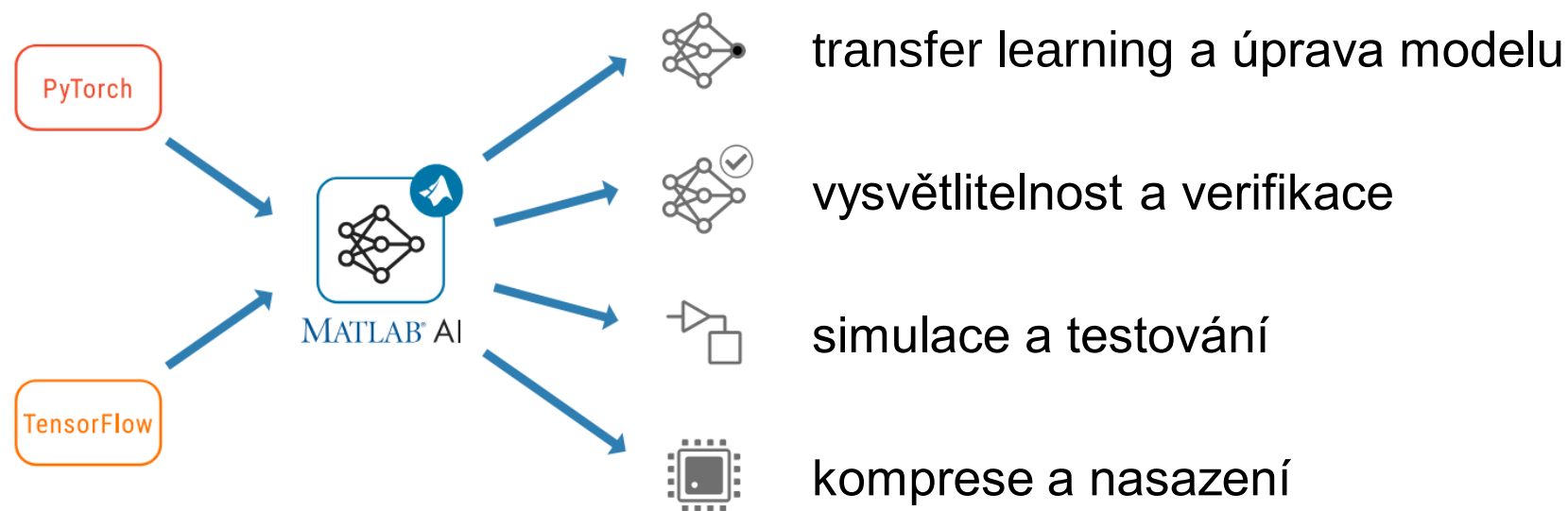
# Transfer learning

- Využití předdefinovaných a před-učených sítí
- Funkce pro import s možností automatické úpravy
  - `[net, classNames] = imagePretrainedNetwork(name)`
- Manuální úprava pomocí funkcí
  - `replaceLayer`, `addLayers`, `connectLayers`, `removeLayers`
- Úprava v aplikaci Deep Network Designer
  - označení a smazání vrstev, přidání nových vrstev
  - odemčení vrstvy pro úpravu parametrů 



# MATLAB a spolupráce s dalšími nástroji

- Import modelů z prostředí TensorFlow, PyTorch a v otevřeném formátu ONNX



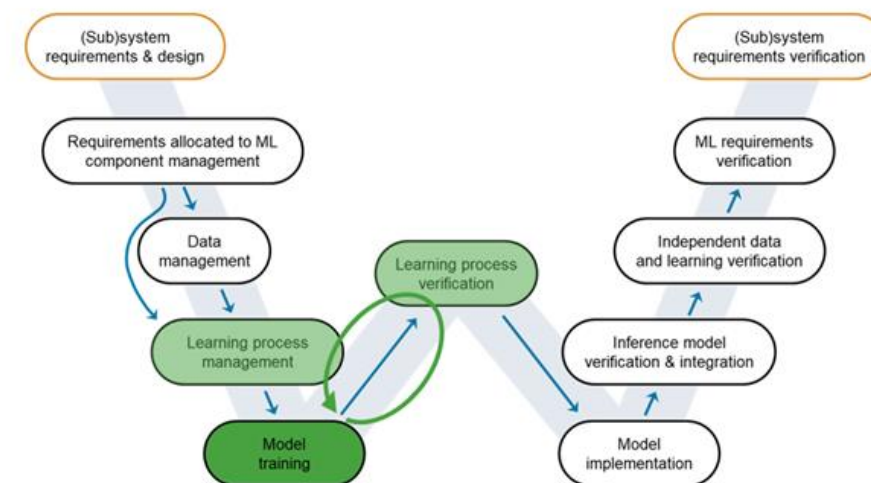
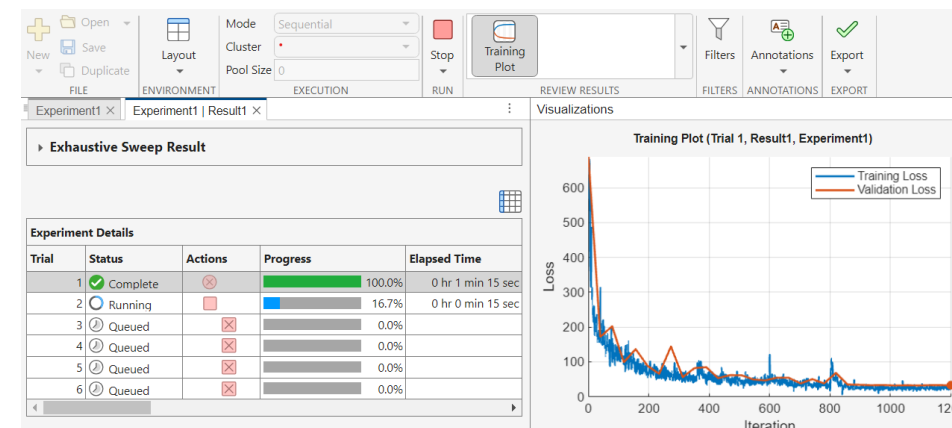
- Export modelů



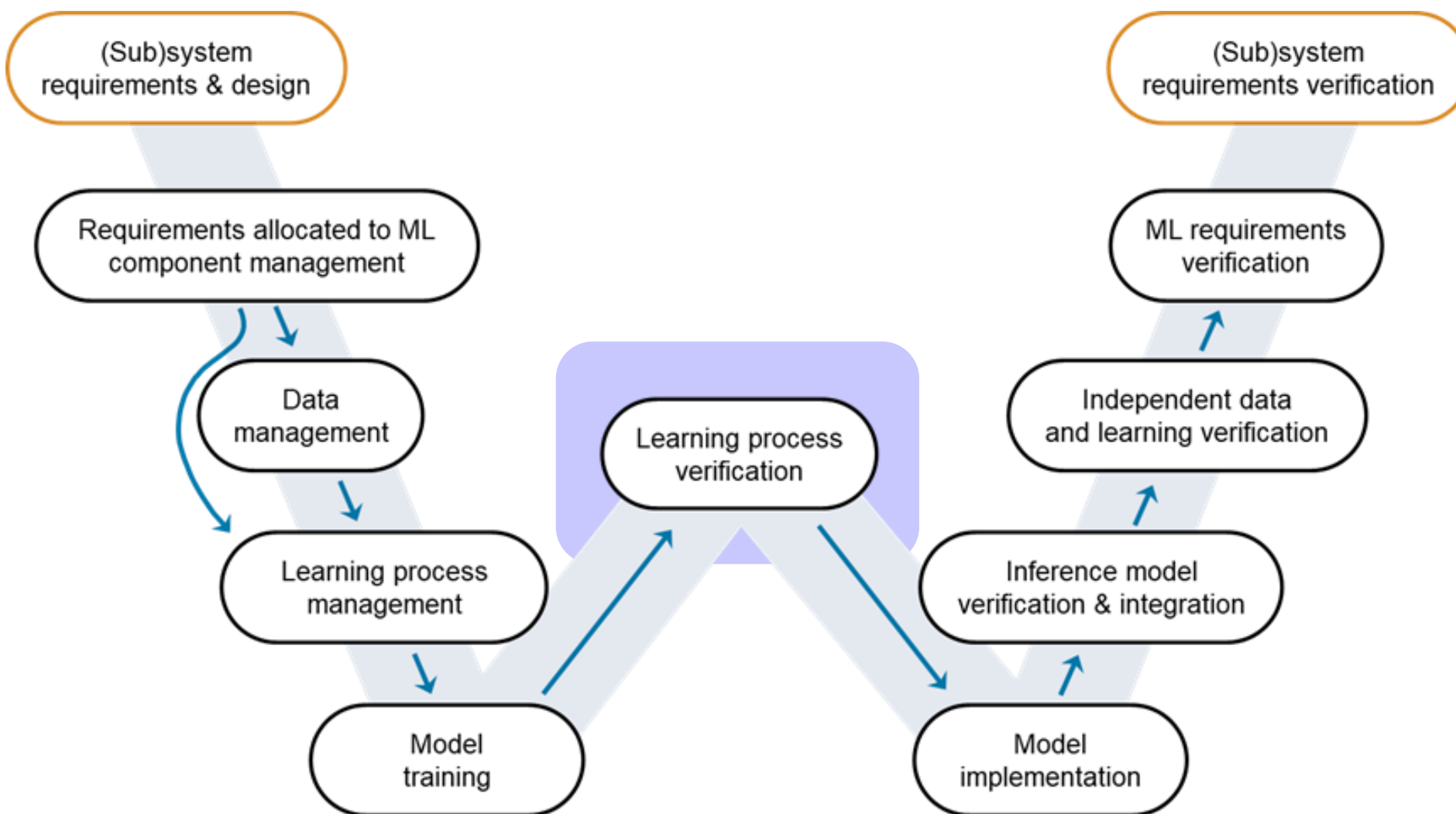
# Učení modelu



- Učení zvoleného modelu / skupiny modelů
- Posouzení dosažené přesnosti modelu
- Hledání optimálních hyperparametrů
  - spuštění experimentů
  - výběr nejlepšího modelu
- Iterativní proces v rámci W-cyklu
  - změny nastavení v případě nedostatečných výsledků
  - výběr odlišných metod pro učení, ...



# Verifikace procesu učení



# Verifikace procesu učení

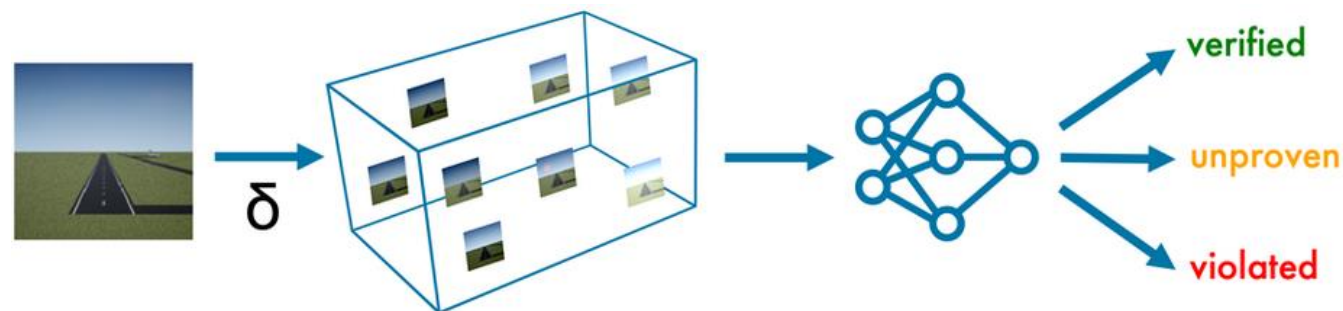
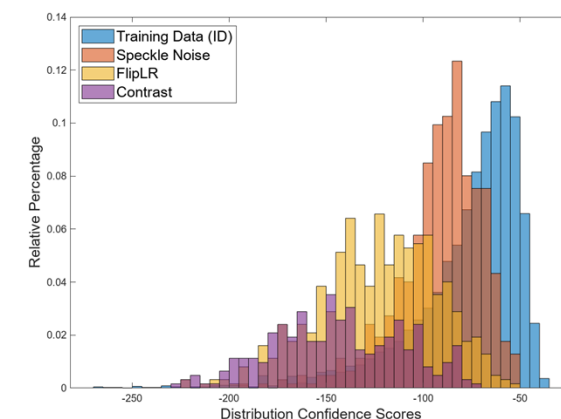
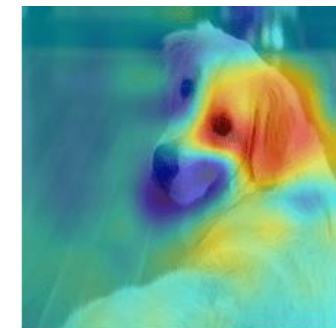


- Testování modelu
  - důležité je testovat s nezávislou sadou dat
  - metriky pro vyhodnocení úspěšnosti sítě
- Pochopení chování modelu (XAI)
  - interpretovatelnost modelu, vysvětlitelnost modelu
  - vizualizační techniky pro pochopení chování modelu
- Ověření robustnosti modelu
  - adversarial examples, formální verifikace
  - detekce „out-of-distribution“ dat

Accuracy: 90.71%

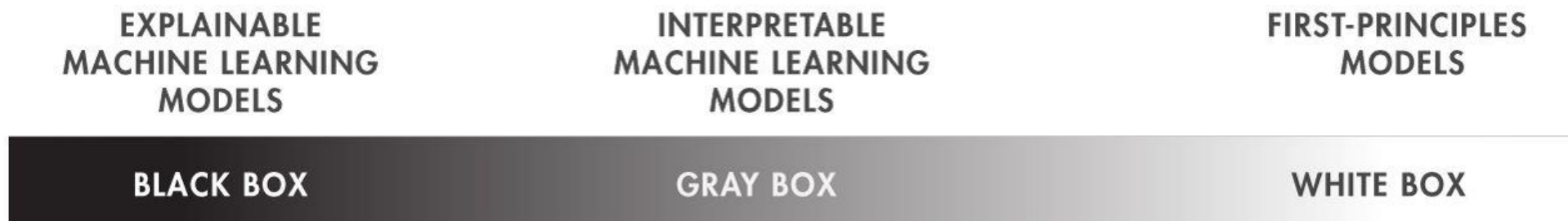
confusionchart(T,Y)

| True Class | Predicted Class |           |
|------------|-----------------|-----------|
|            | normal          | pneumonia |
| normal     | 193             | 41        |
| pneumonia  | 17              | 373       |



# Interpretovatelnost vs. vysvětlitelnost

- Interpretovatelnost (interpretability)
  - odhalení, jak AI model dospěl k danému výsledků
  - založeno na pochopení vnitřního mechanismu modelu
- Vysvětlitelnost (explainability)
  - vysvětlení chování modelu „lidskými termíny“
  - bez nutnosti porozumění vnitřnímu mechanismu (typické pro deep learning)

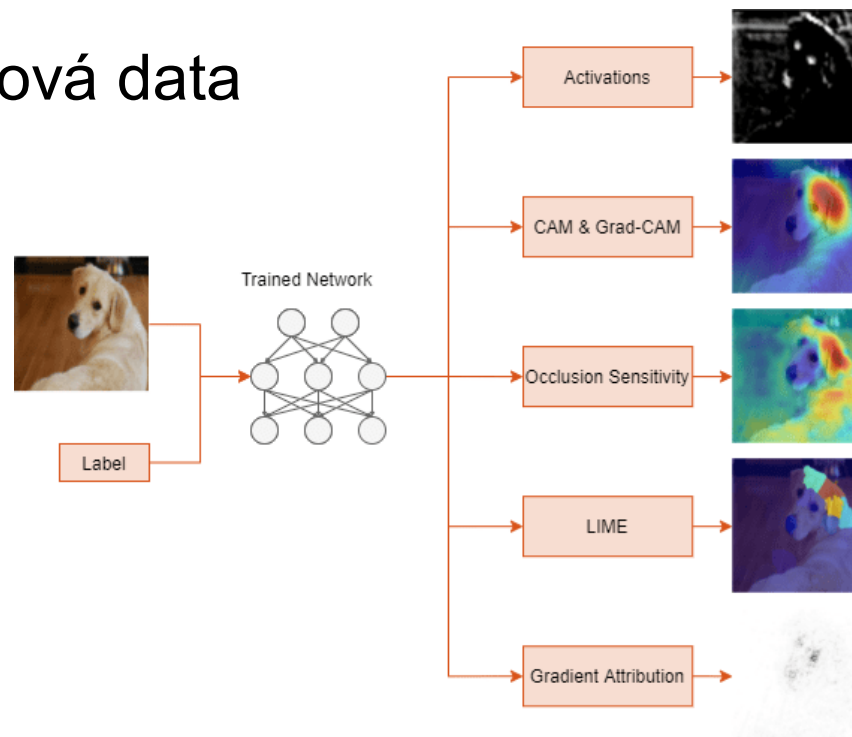


# Vizualizační metody pro deep learning

- Metody pro obrazová data



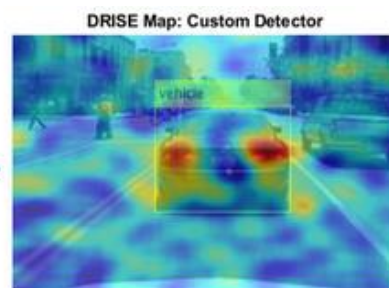
deep dream



- Detekce objektů

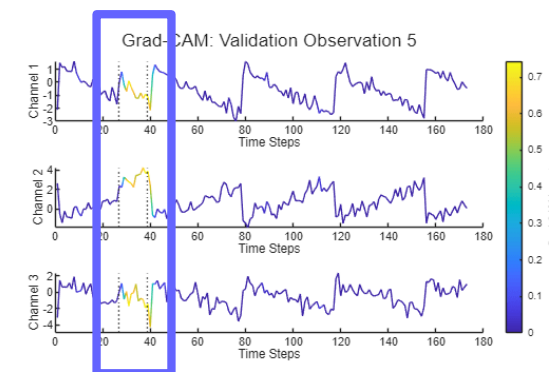


detekce objektů

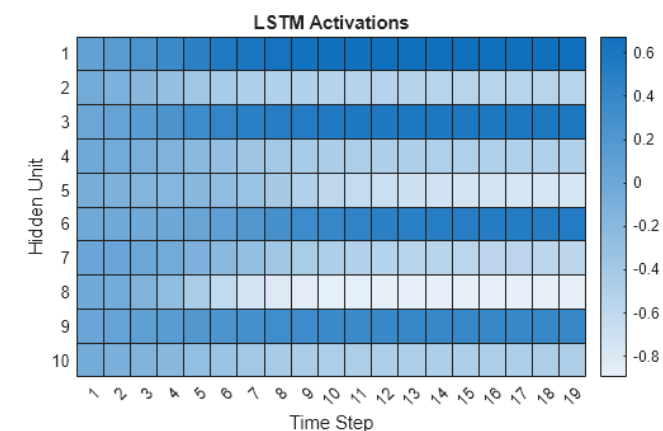


vysvětlení detekce (D-RISE)

## Metody pro ne-obrazová data



Grad-CAM



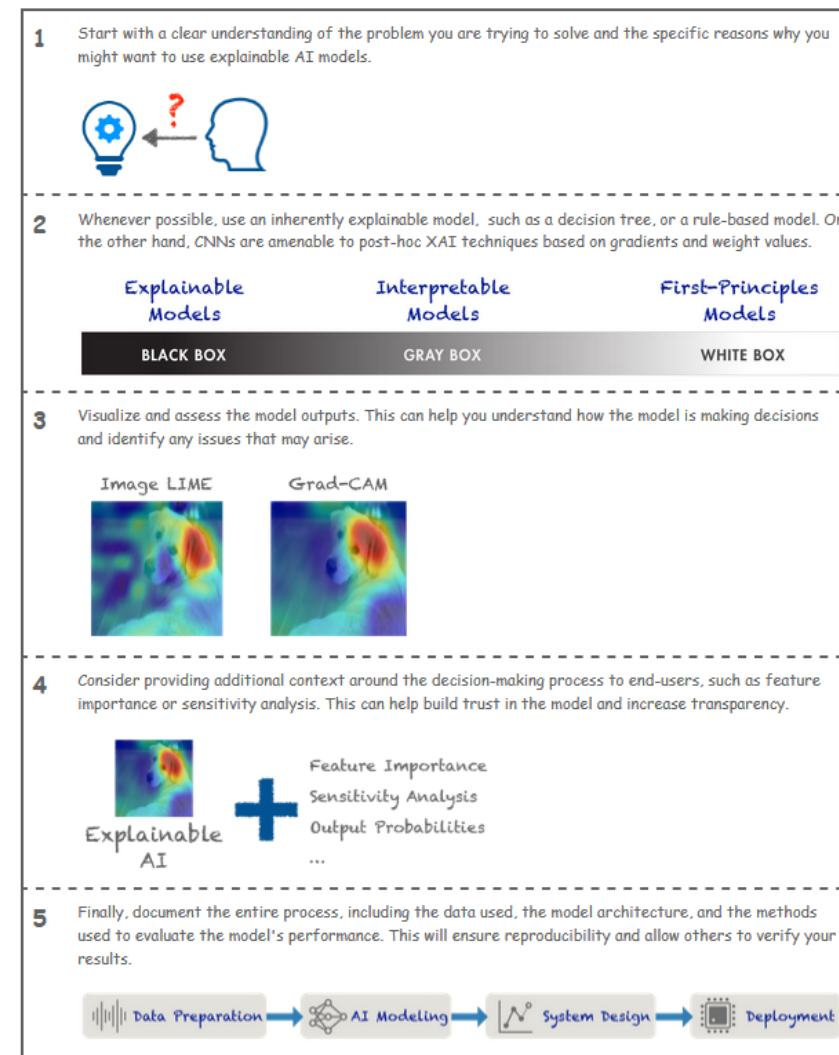
Activations



# Ověření správného fungování sítě

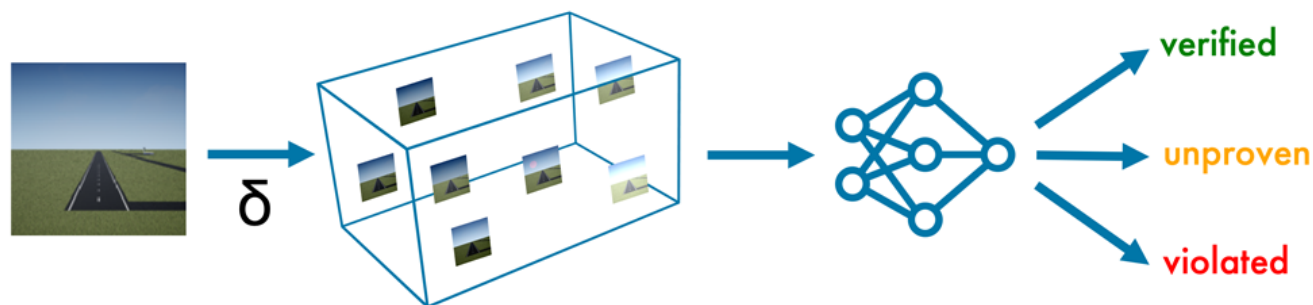
- „Cheat sheet“ s praktickými radami

1. pochopení problému ~ proč potřebujete XAI
2. inherentně vysvětlitelné modely vs. post-hoc XAI
3. vizualizace a posouzení výsledků
4. dodatečný kontext
5. dokumentace celého procesu



# Verifikace AI modelů

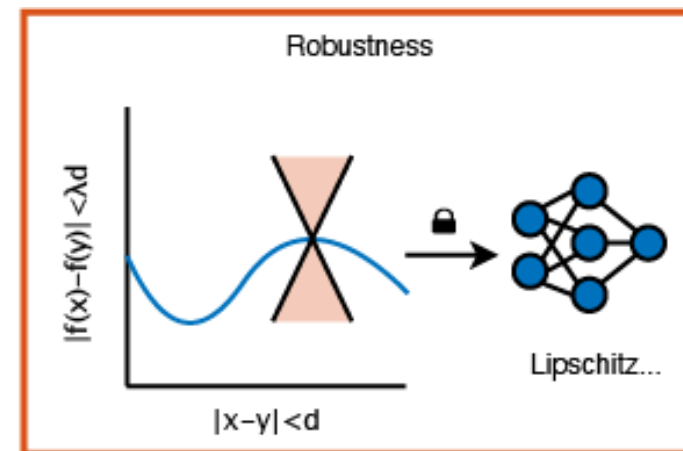
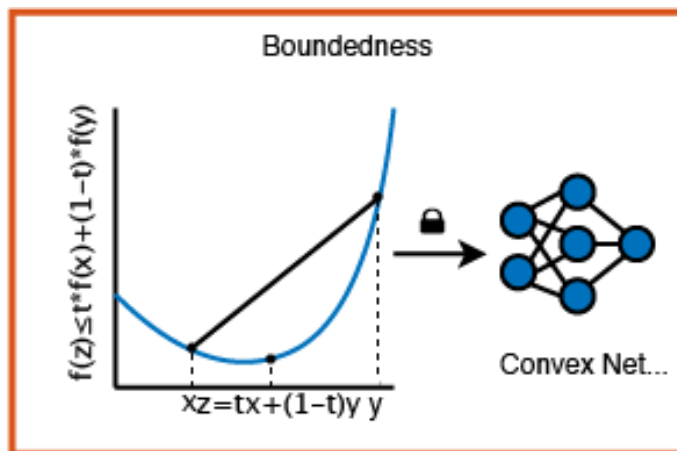
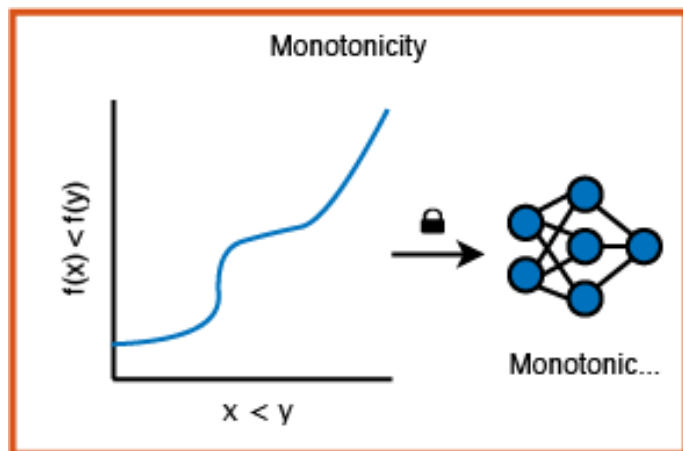
- Deep Learning Toolbox Verification Library
  - cíl: zajistit robustnost a spolehlivost hlubokých neuronových sítí
- Formální metody pro ověření robustnosti
  - proti tzv. adversarial examples



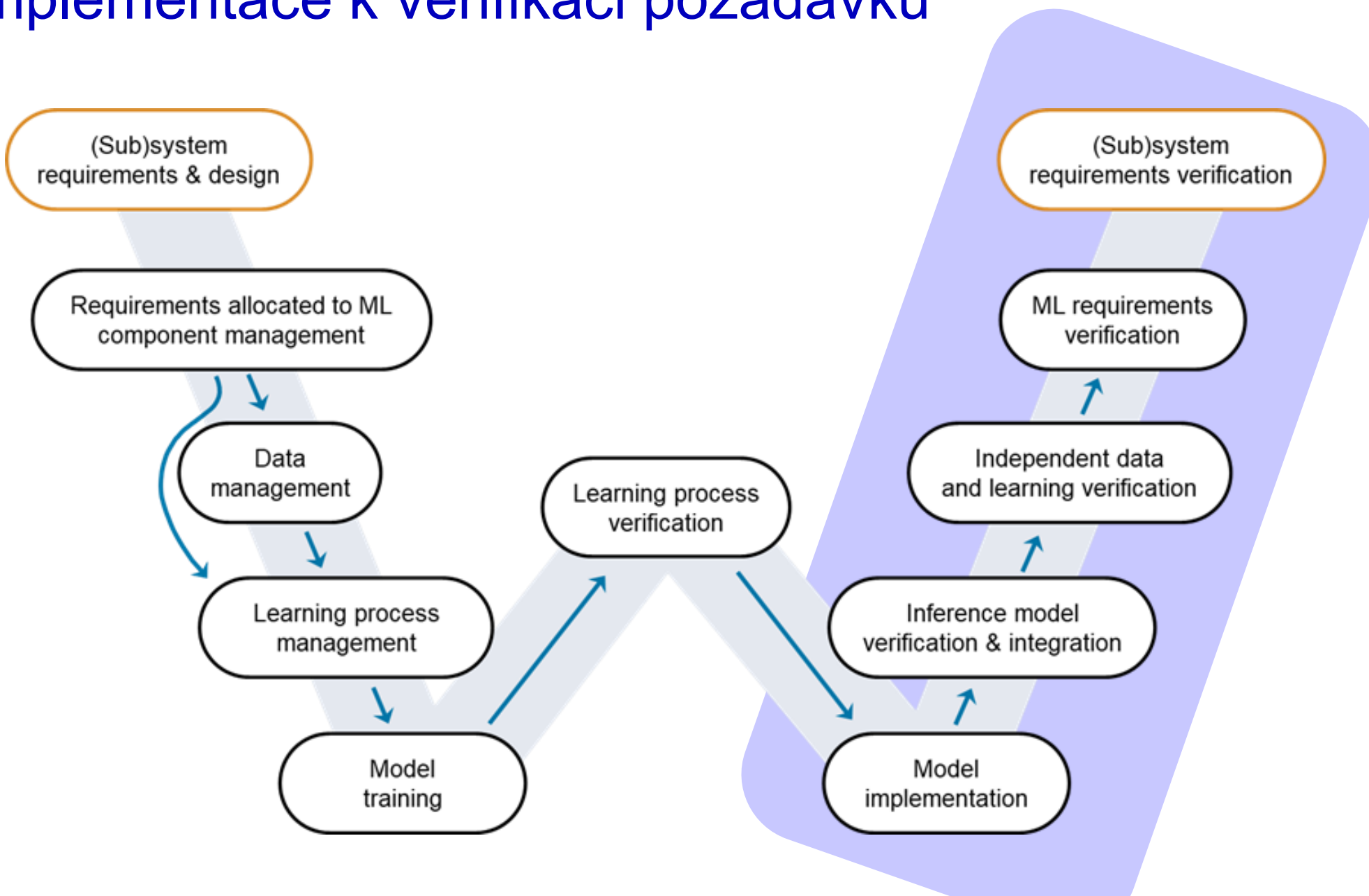
- Odhad citlivosti sítě na drobné změny vstupu
- Rozdělení dat na in-distribution a out-of-distribution
  - lze sledovat za běhu nasazené sítě

# Constrained Deep Learning

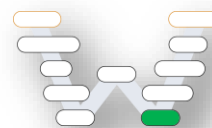
- Neuronové sítě, které integrují do procesu učení zadaná omezení
  - například monotónnost, konvexnost, apod.



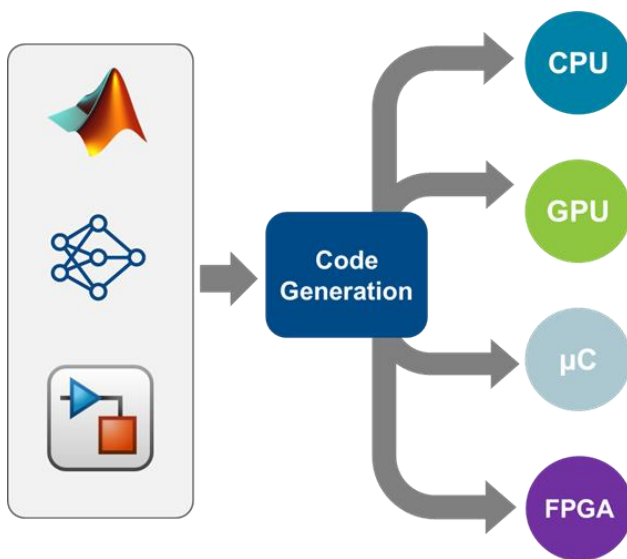
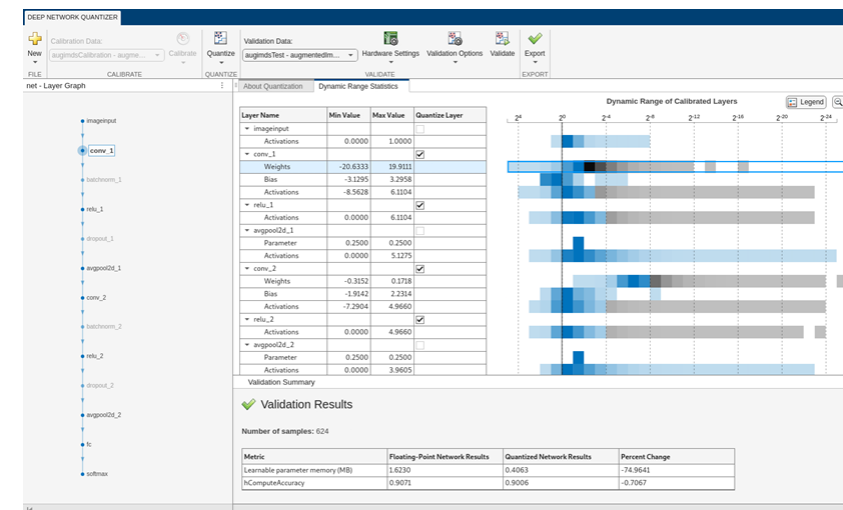
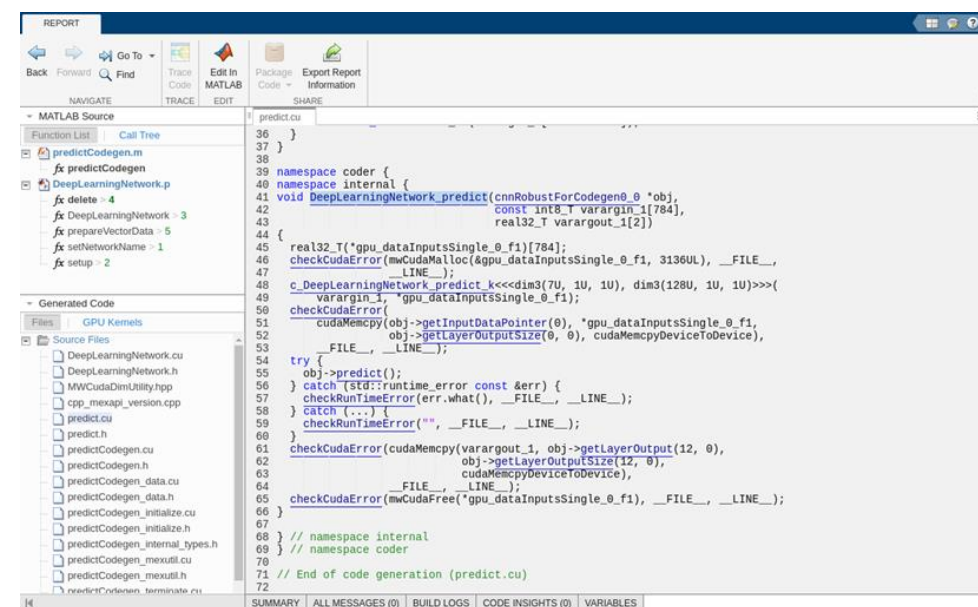
# Od implementace k verifikaci požadavků



# Implementace modelu



- Příprava modelu pro implementaci
  - optimalizace modelu pro cílovou platformu
  - kvantizace, pruning, projekce
- Automatické generování kódu

The screenshot shows a MATLAB report titled 'REPORT' with a 'MATLAB Source' tab. The 'Generated Code' section displays C++ code for a deep learning network. The code includes namespace definitions, data pointers, and CUDA calls for memory management and computation. The code is structured as follows:

```

36 }
37 }
38 namespace code {
39 namespace internal {
40 void DeepLearningNetwork_predict(cnnRobustForCodegen_0 *obj,
41 const int8_T varargin_1[784],
42 real32_T varargin_1[2])
43 {
44     real32_T(*gpu_dataInputsSingle_0_f1)[784];
45     checkCudaError(cudaMalloc(&gpu_dataInputsSingle_0_f1, 3136UL), __FILE__,
46                     __LINE__);
47     c_DeepLearningNetwork_predict_k<<<dim3(7U, 1U, 1U), dim3(128U, 1U, 1U)>>>
48     varargin_1, *gpu_dataInputsSingle_0_f1);
49     checkCudaError(cudaMemcpy(obj->getInputDataPointer(0), *gpu_dataInputsSingle_0_f1,
50                             obj->getLayerOutputSize(0, 0), cudaMemcpyDeviceToDevice),
51                     __FILE__, __LINE__);
52     try {
53         obj->predict();
54     } catch (std::runtime_error const &err) {
55         checkRuntimeError(err.what(), __FILE__, __LINE__);
56     } catch (...) {
57         checkRuntimeError("", __FILE__, __LINE__);
58     }
59     checkCudaError(cudaMemcpy(varargout_1, obj->getLayerOutput(12, 0),
60                             obj->getLayerOutputSize(12, 0),
61                             cudaMemcpyDeviceToDevice),
62                     __FILE__, __LINE__);
63     checkCudaError(cudaFree(*gpu_dataInputsSingle_0_f1), __FILE__, __LINE__);
64 }
65 }
66 // namespace internal
67 // namespace code
68 // End of code generation (predict.cu)
69
70
71
72
    
```

- Design

Compression Analysis

### Analysis for Network Compression

Analysis Date: 04-Jul-2024 18:29:44

Maximum Compression

Pruning

96.2%

Projection

97.9%

Combine techniques for further compression

Minimum Network Size

Current Network

40.8 KB

Pruning

22.2 KB

Projection

22.2 KB

74 layers

274.8k total parameters

1 MB memory

Layer Information

Settings

|    | Name        | Type                | Pruning | Projection | Number of Learnables | Learnables Memory (KB) | Number of States |   |
|----|-------------|---------------------|---------|------------|----------------------|------------------------|------------------|---|
| 2  | conv1p      | 2-D Convolution     | ✓       | ✓          | 448                  | 1.7500                 | 0                | + |
| 3  | BN1p        | Batch Normalization | ✓       | ⊗          | 32                   | 0.1250                 | 32               |   |
| 5  | ST1U1_conv1 | 2-D Convolution     | ✓       | ✓          | 2320                 | 9.0625                 | 0                |   |
| 6  | ST1U1_BN1   | Batch Normalization | ✓       | ⊗          | 32                   | 0.1250                 | 32               |   |
| 8  | ST1U1_conv2 | 2-D Convolution     | ✓       | ✓          | 2320                 | 9.0625                 | 0                |   |
| 9  | ST1U1_BN2   | Batch Normalization | ✓       | ⊗          | 32                   | 0.1250                 | 32               |   |
| 12 | ST1U2_conv1 | 2-D Convolution     | ✓       | ✓          | 2320                 | 9.0625                 | 0                |   |
| 13 | ST1U2_BN1   | Batch Normalization | ✓       | ⊗          | 32                   | 0.1250                 | 32               |   |

✓ Fully compressible

✓ Partially compressible

⚠ Blocked by network architecture

⊗ Not supported

Layer Parameter Memory

Orange

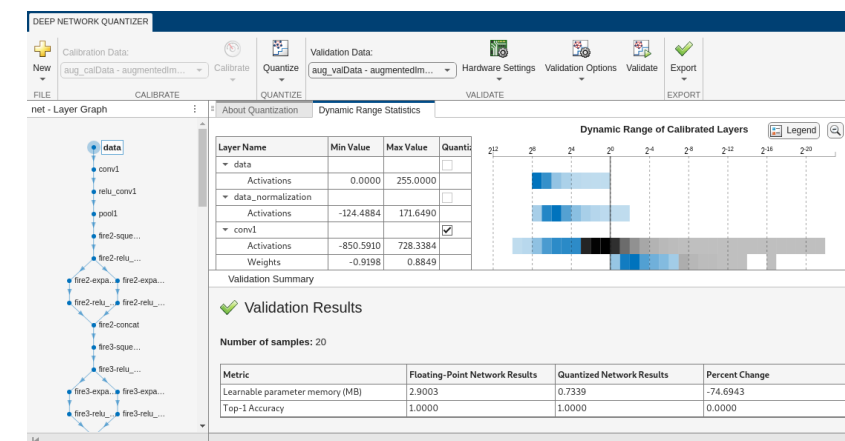
2-D Convolution
 

Yellow

Batch Normalization
 

Purple

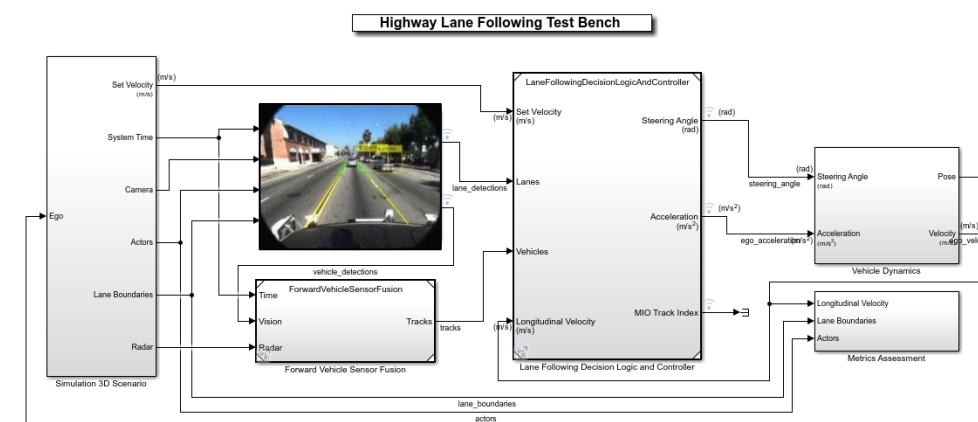
Fully Connected



# Verifikace a integrace modelu pro inferenci



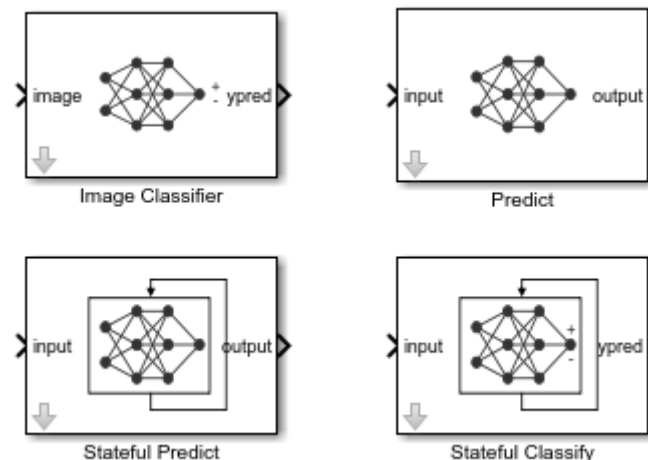
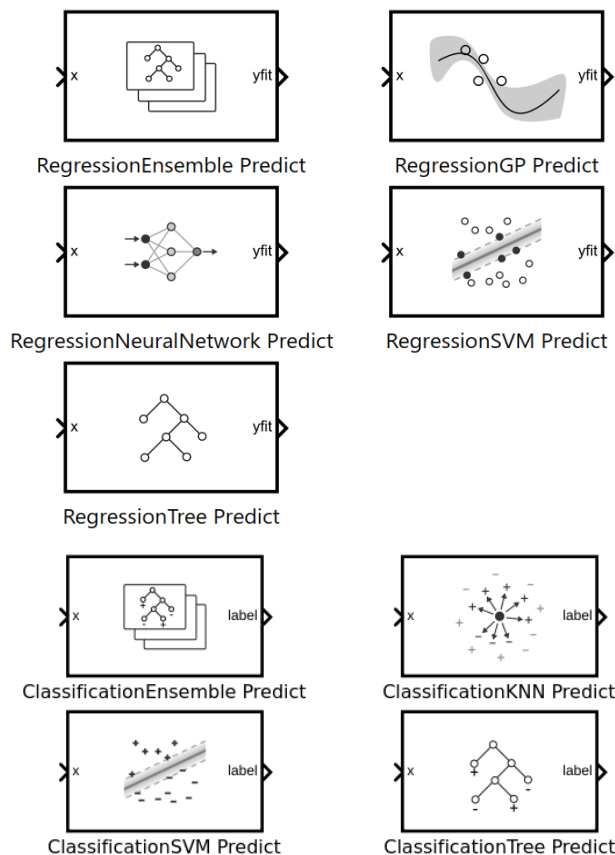
- Dvě důležité provázané fáze nasazení AI modelu
  - přechod modelu z teoretické roviny do prakticky provozovaného nástroje
- Verifikace implementovaného modelu
  - ověření přesnosti modelu v implementované podobě
  - zásadní zejména pro bezpečnostně kritické aplikace
- Integrace modelu
  - zapojení modelu do širšího návrhu
  - propojení s dalšími algoritmy
  - testování v kontextu celkové funkčnosti
- Simulink jako platforma pro integraci
  - integrace může být řešena současně s předchozí fází implementace
  - testování prostřednictvím simulací a nástrojů pro tvorbu a správu testů (Simulink test)



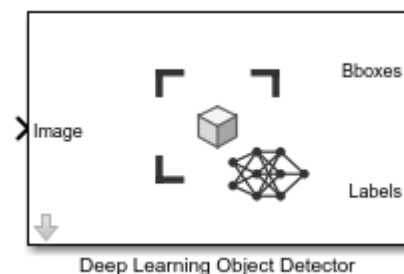


# Jak zahrnout AI model do Simulinku

## Modely pro machine learning

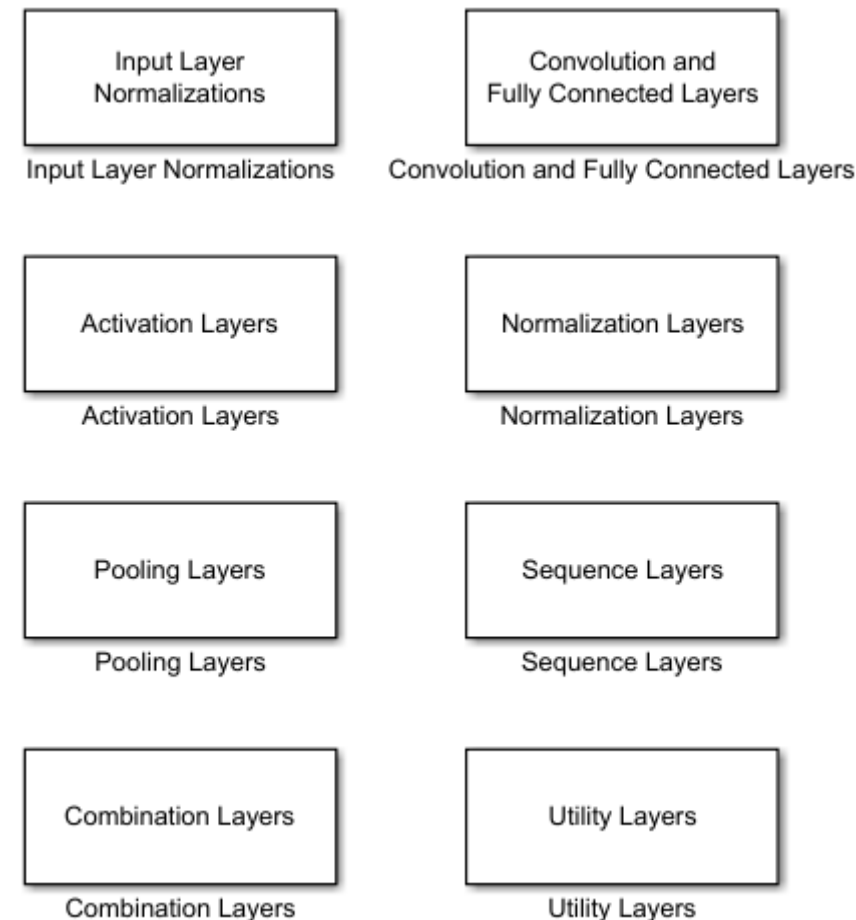


## Kompletní neuronové sítě

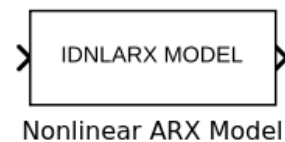
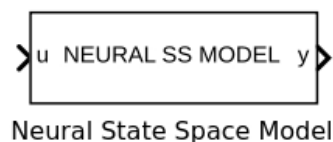


## Detekce objektů

## Bloky pro neuronové vrstvy

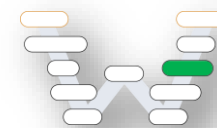


NSS, ARX



... a další připravené bloky v toolboxech.

# Nezávislé ověření dat a procesu učení



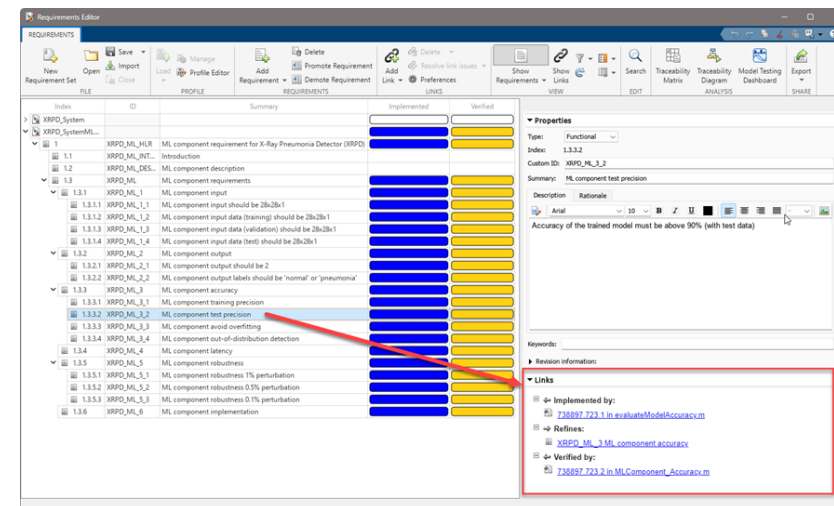
- Následuje po důkladném ověření inferenčního modelu na cílové platformě
- Ověření, zda byly datové sady v průběhu vývoje řádně spravovány
  - nezávislé přezkoumání které potvrdí, že trénovací, validační a testovací datové sady:
    - a) splňují striktní požadavky na správu dat
    - b) jsou úplné a reprezentativní pro vstupní datový prostor dané aplikace
- Ověření procesu učení
  - ověření, zda byl trénovaný model uspokojivě verifikován, včetně nezbytných analýz pokrytí



# Verifikace požadavků

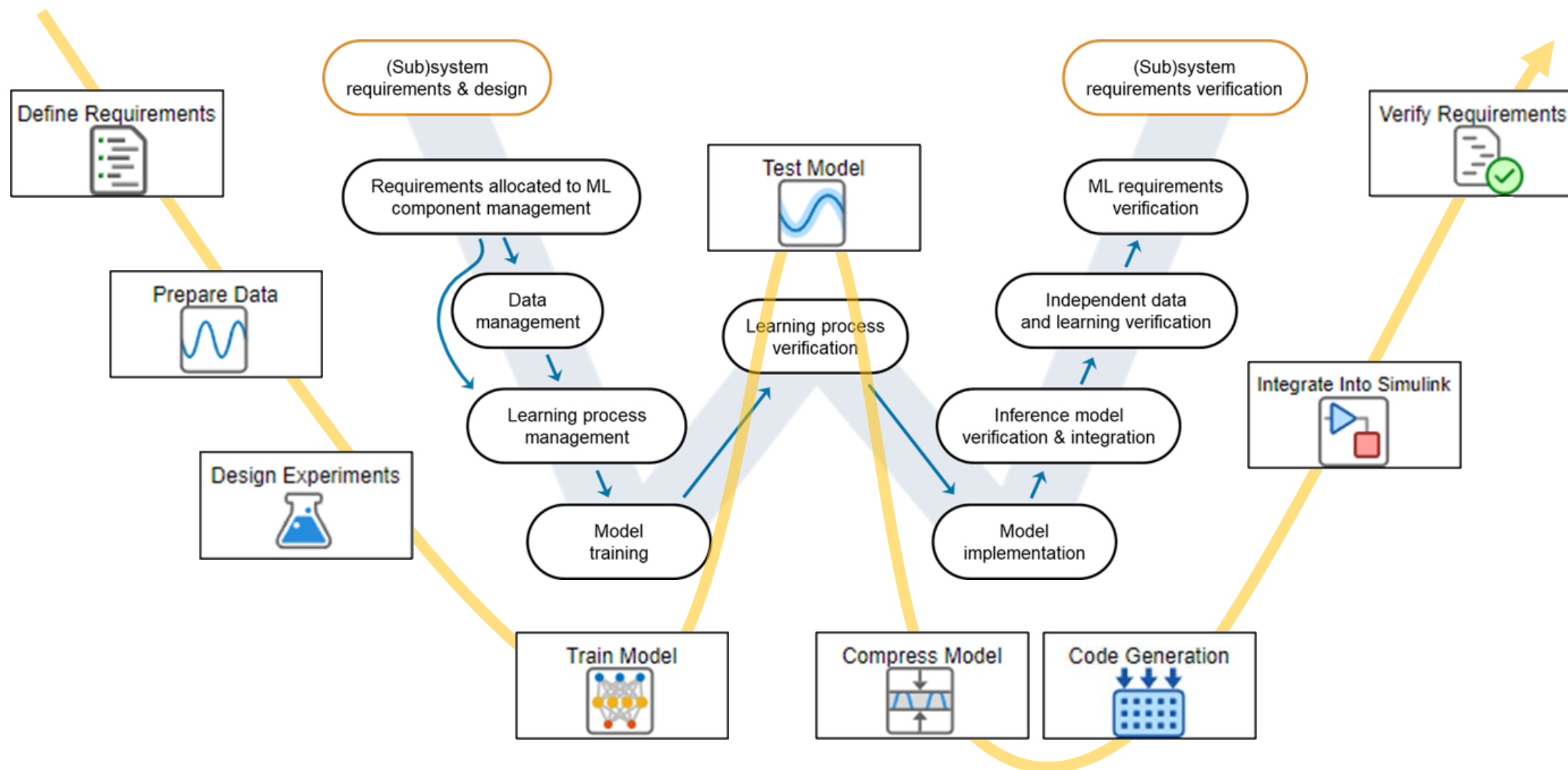


- Závěrečná část W-cyklu
- Ověření, že byly veškeré požadavky
  - a) implementovány v podobě navrhované funkčnosti
  - b) splněny v podobě provedených testů
- Provázání požadavků s implementovanými funkcemi a testy
  - klíčové pro uzavření smyčky vývojového procesu
  - spuštěním všech definovaných testů ověříme, že požadavky byly adekvátně implementovány



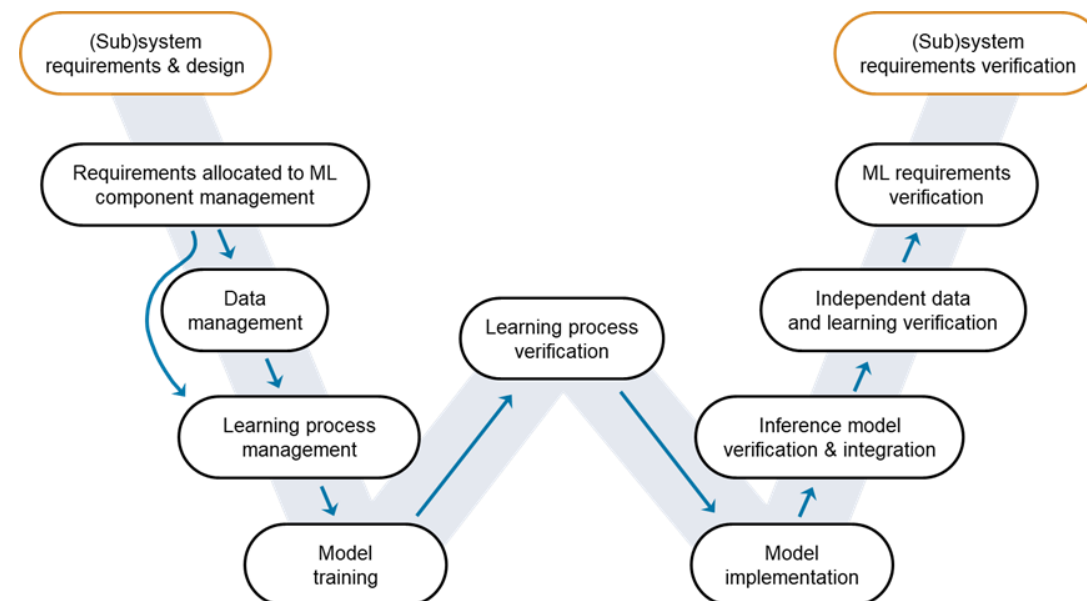
| Index               | ID             | Summary                                                      | Implemented | Verified |
|---------------------|----------------|--------------------------------------------------------------|-------------|----------|
| > XRPD_System       |                |                                                              |             |          |
| > XRPD_SystemMLC... |                |                                                              |             |          |
| 1                   | XRPD_ML_HLR    | ML component requirement for X-Ray Pneumonia Detector (XRPD) |             |          |
| 1.1                 | XRPD_ML_INT... | Introduction                                                 |             |          |
| 1.2                 | XRPD_ML_DES... | ML component description                                     |             |          |
| 1.3                 | XRPD_ML        | ML component requirements                                    |             |          |
| 1.3.1               | XRPD_ML_1      | ML component input                                           |             |          |
| 1.3.1.1             | XRPD_ML_1_1    | ML component input should be 28x28x1                         |             |          |
| 1.3.1.2             | XRPD_ML_1_2    | ML component input data (training) should be 28x28x1         |             |          |

# Shrnutí: Vývojový cyklus W ~ propojení AI a MBD



# Další informace a příklady

- Série článků
  - [část 1. úvod](#)
  - [část 2. požadavky a modelování](#)
  - [část 3. verifikace učení](#)
  - [část 4. implementace a verifikace požadavků](#)
- Ukázky:
  - [End-to-End AI Workflows](#)
  - [Verify an Airborne Deep Learning System](#)



Credit: EASA, Daedalean.

# Vybrané uživatelské reference

- AI-Powered Virtual Sensor Delivers Real-Time Cabin Air Mass Flow Estimation
  - Mercedes-Benz Research & Development
- TinyML to Enhance Performance of Field-Oriented Control
  - STMicroelectronics
- Implementing a Workflow for Deploying and Integrating Deep Learning Networks on PLCs for Industrial Automation
  - Beckhoff Automation GmbH & Co. KG



Děkuji za pozornost