CodeScene

Ergo, SMIRK is Safe: A Safety Case for a Machine Learning Component in a Pedestrian Emergency Brake System

Markus Borg

markus.borg@codescene.com

31st IEEE Int'l. Requirements Engineering Conf.

Sep 6, 2023

Open ML safety case

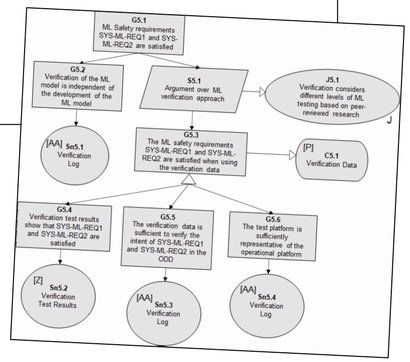
Software Quality Journal (2023) 31:335–403
https://doi.org/10.1007/s11219-022-09613-1

Check for updates

Ergo, SMIRK is safe: a safety case for a machine learning component in a pedestrian automatic emergency brake system

Markus Borg[1,2] · Jens Henriksson[3] · Kasper Socha[1,2] · Olof Lennartsson[4] · Elias Sonnsjö Lönegren[4] · Thanh Bui[1] · Piotr Tomaszewski[1] · Sankar Raman Sathyamoorthy[5] · Sebastian Brink[6] · Mahshid Helali Moghadam[1]

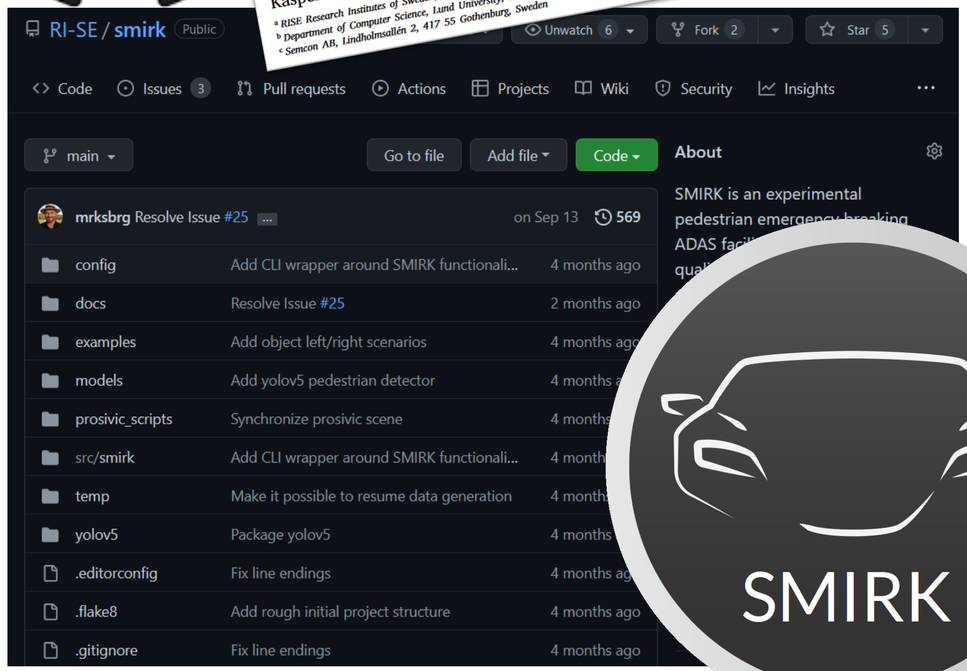
cember 2022 / Published online: 1 March 2023
2023




Contents lists available at ScienceDirect
Software Impacts
journal homepage: www.journals.elsevier.com/software-impacts
ELSEVIER
Original software publication

SMIRK: A machine learning-based pedestrian automatic emergency braking system with a complete safety case

Kasper Socha [a], Markus Borg [a,b,*], Jens Henriksson [c]
[a] RISE Research Institutes of Sweden, Scheelevägen 17, 223 63 Lund, Sweden
[b] Department of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden
[c] Semcon AB, Lindholmsallén 2, 417 55 Gothenburg, Sweden



Open ML-based demonstrator


2023
R

Introduction

Who is Markus?

Development engineer, ABB

- Process automation, safety

PhD student, Lund University

- Traceability, change impact analysis

Senior researcher, RISE

- AI engineering and safety

Principal researcher, CodeScene

- AI for software engineering

LUND UNIVERSITY

RI. SE Research Institutes of Sweden

CodeScene

Markus Borg

Kasper Socha

semcon

Jens Henriksson

Olof Lennartsson

INFOTIV

Elias Sonnsjö Lönegren

Mashid Helali

Thanh Bui

RI.SE

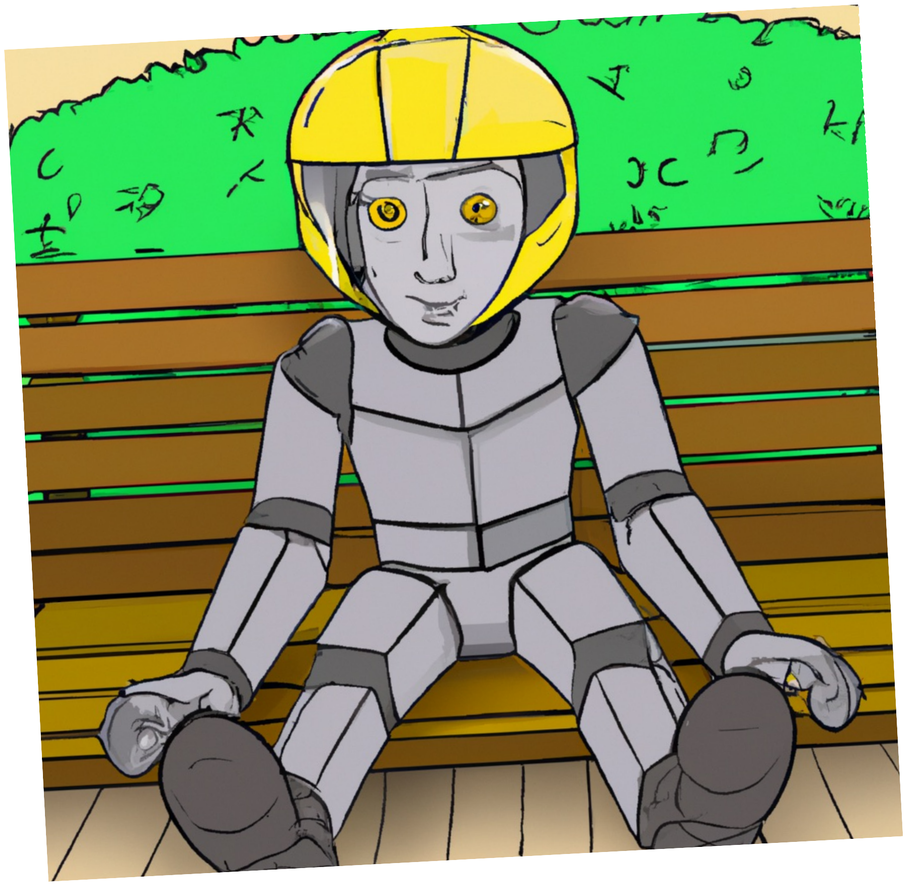
Piotr Tomaszewski

QRTECH
an EMBRON Company

Sankar Sathyamoorthy

COMBITECH

Sebastian Brink

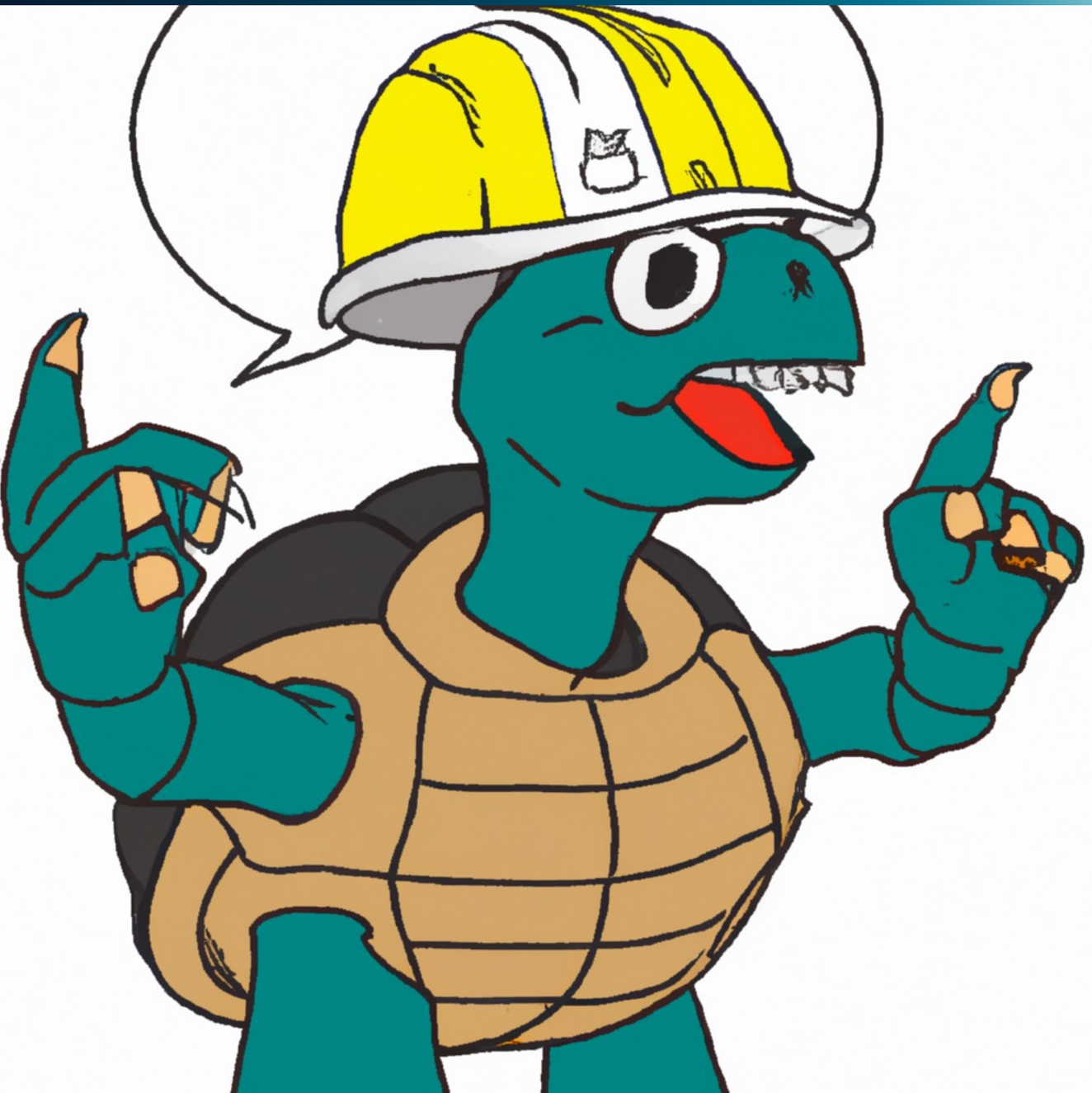


Standards and guidelines are high-level...

... must get our hands dirty with ML details

Lack of:
- experience reports
- open demonstrator systems

“How to demonstrate and share a complete ML safety case for an open ADAS?”



Safety case

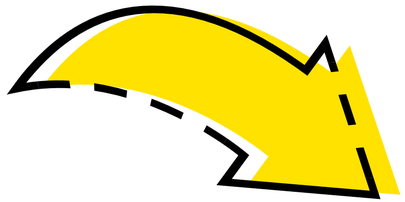
a structured argumentation

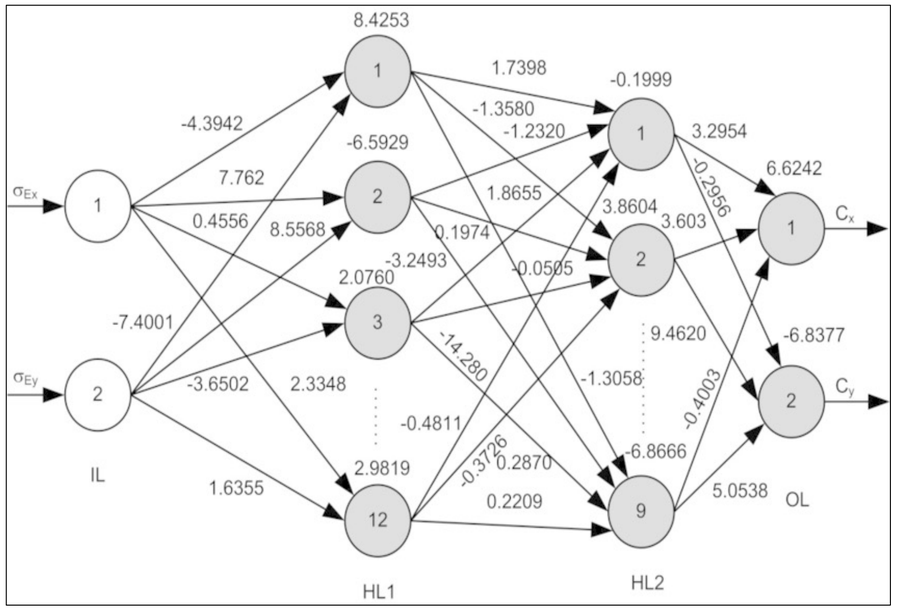
backed up by evidence

Motivation and Context

```
unction fetch(collection, url){
 console.log('fetching a list of docs in collection ' + collection + '...');
 request('GET', url, {
     headers: {
       'Gdata-Version': '3.0',
       'Authorization': 'GoogleLogin auth=' + getAuthToken()
     }
   },
   function(chunk){
     var entries = chunk.split('<entry');
     entries.shift();
     entries.forEach(function(entry){
       var title = entry.match(/<title>(.*?)<\/title>/)[1];
       if (title.match(/\.ngdoc$/)) {
         var exportUrl = entry.match(/<content type='text\/html' src='(.*?
         download(collection, title, exportUrl);
       }
     });
   }
 );
}

function download(collection, name, url) {
  console.log('Downloading:', name, '...');
  request('GET', url + '&exportFormat=txt',
    {
      headers: {
        'Gdata-Version': '3.0',
        'Authorization': 'GoogleLogin auth=' + getAuthToken()
      }
    },
    function(data){
      data = data.replace('\ufeff', '');
      data = data.replace(/\r\n/mg, '\n');
```

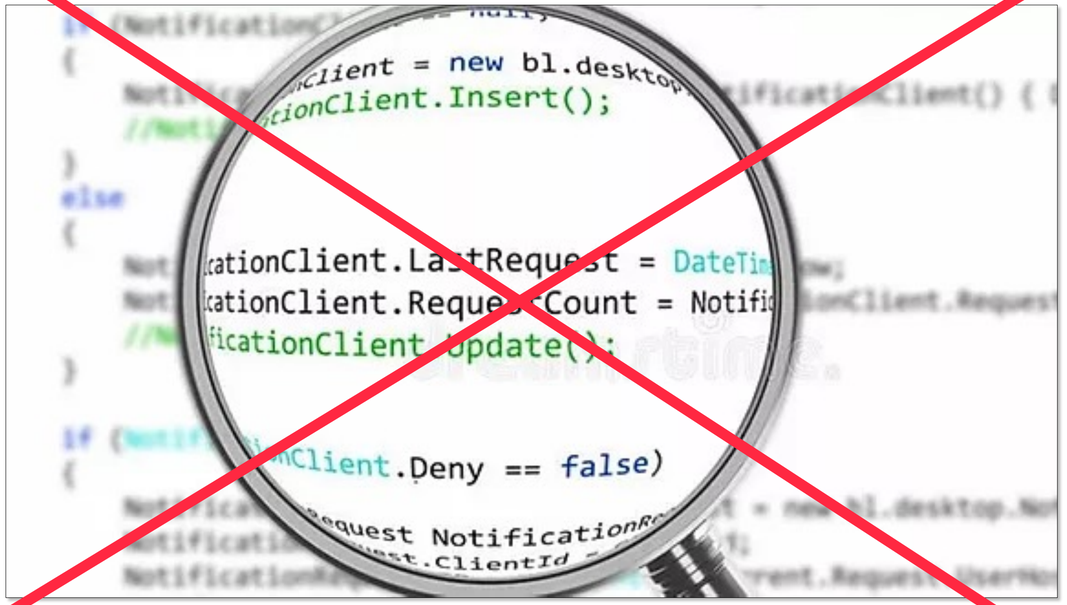




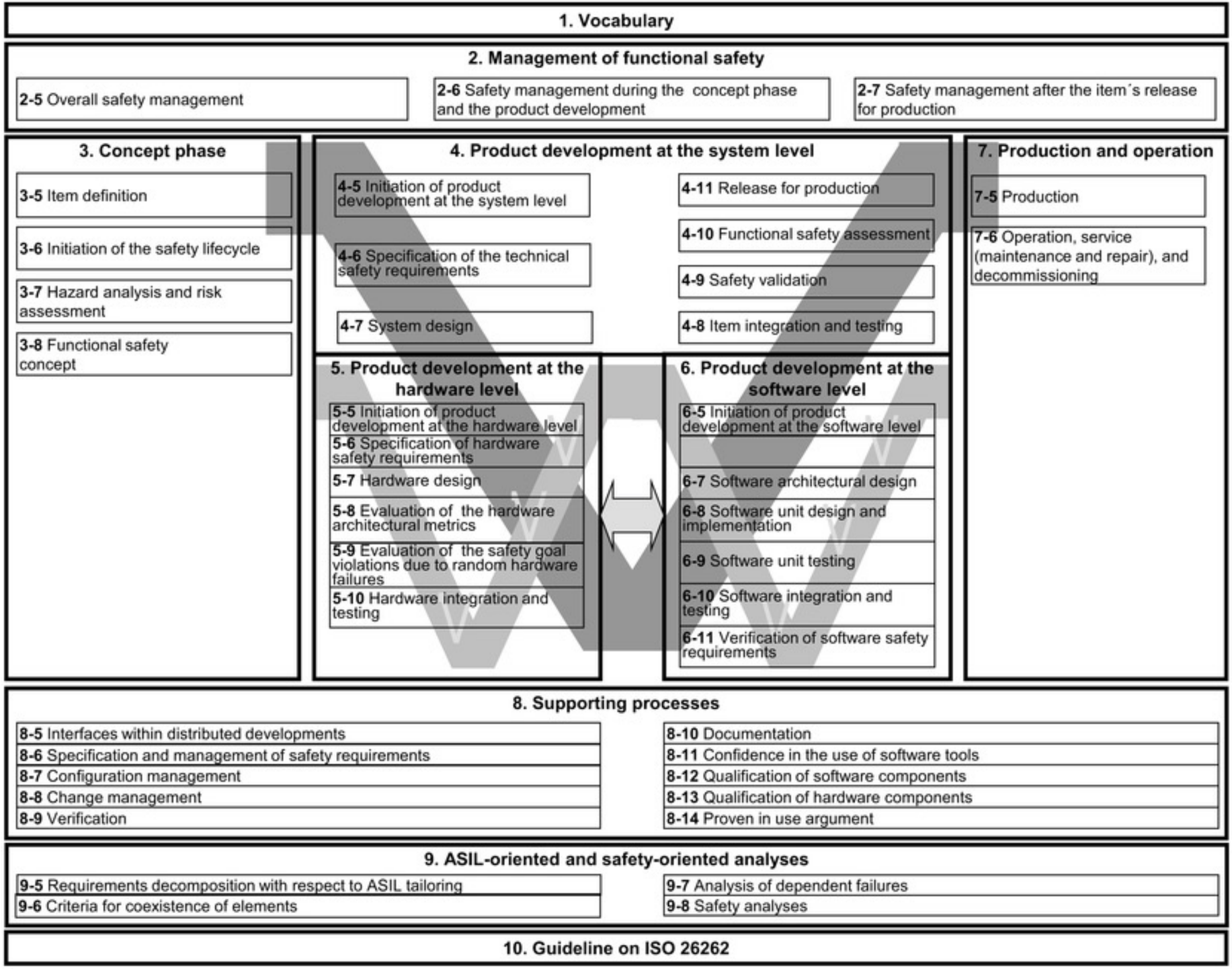
```
    public boolean addAll(int index, Collection c) {
        if(c.isEmpty()) {
            return false;
        } else if(_size == index || _size == 0) {
            return addAll(c);
        } else {
            Listable succ = getListableAt(index);
            Listable pred = (null == succ) ? null : succ.prev();
            Iterator it = c.iterator();
            while(it.hasNext()) {
                pred = insertListable(pred,succ,it.next());
            }
            return true;
        }
    }
```

1 of 2 branches missed.

Press 'F2' for focus

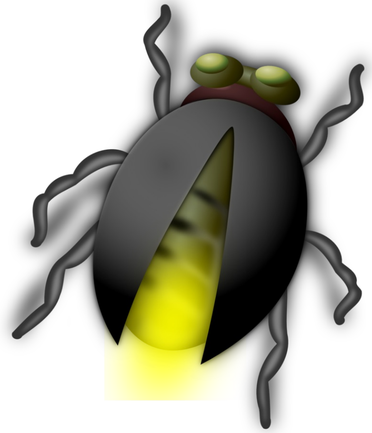


Well Hello There

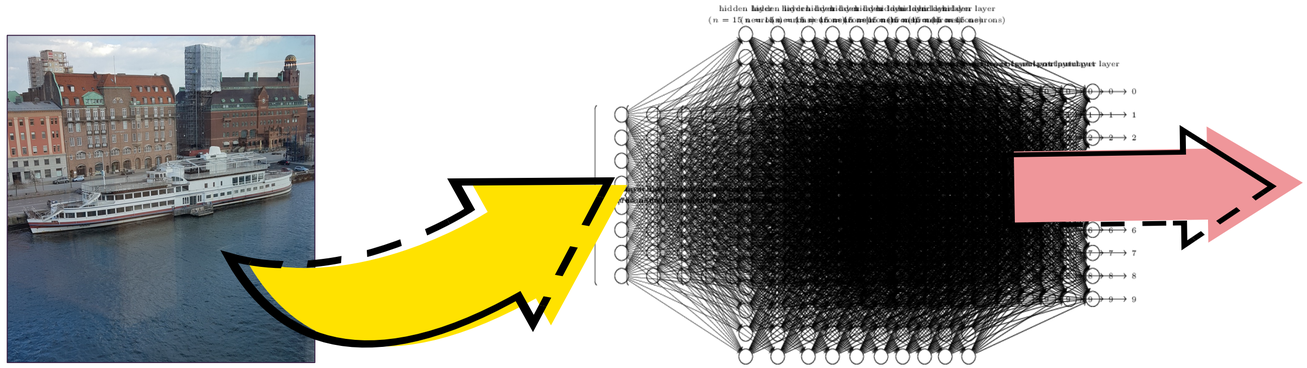


Functional Safety and Machine Learning

"absence of unreasonable risk due to hazards resulting from malfunctions of the electrical/electronic system"



What if...



No object detected

Not a bug – functionality delivered according to training!

CodeScene

Machine Learning Safety

ISO 21448:2022

Safety of the intended functionality (SOTIF)

Automotive Software Safety

Functional Safety

Absence of unreasonable risk due to ...

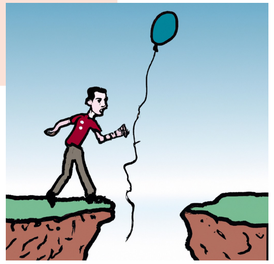
...malfunctions

ISO 26262

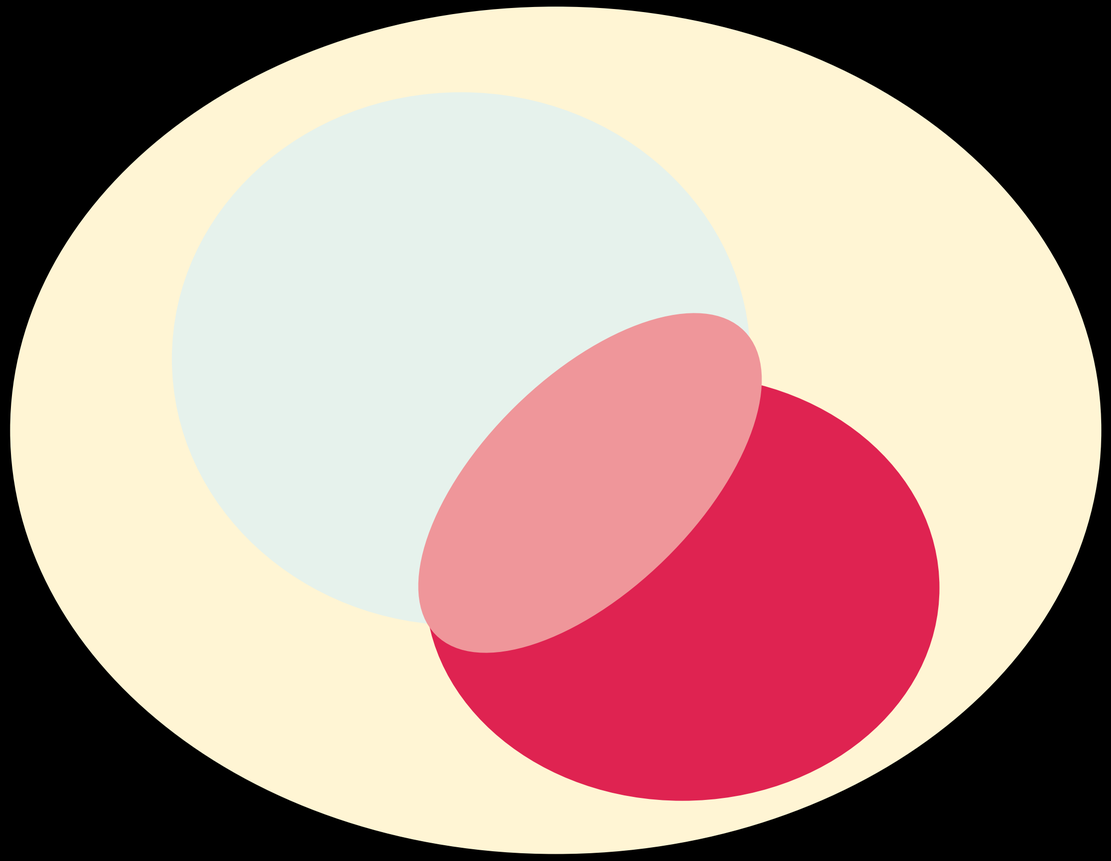
...functional insufficiencies

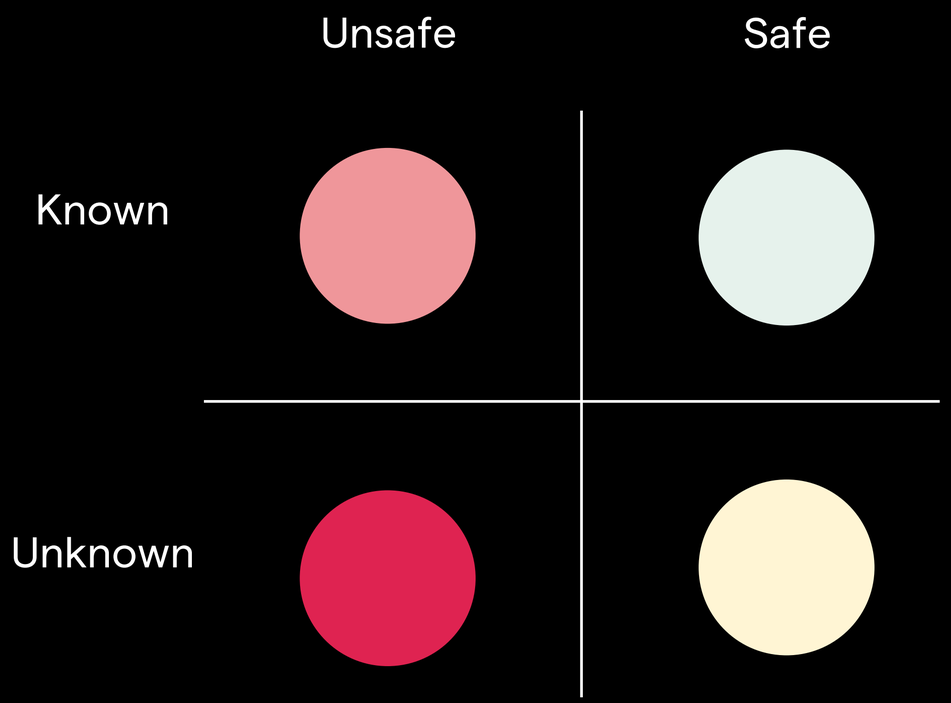
ISO 21448

SOTIF

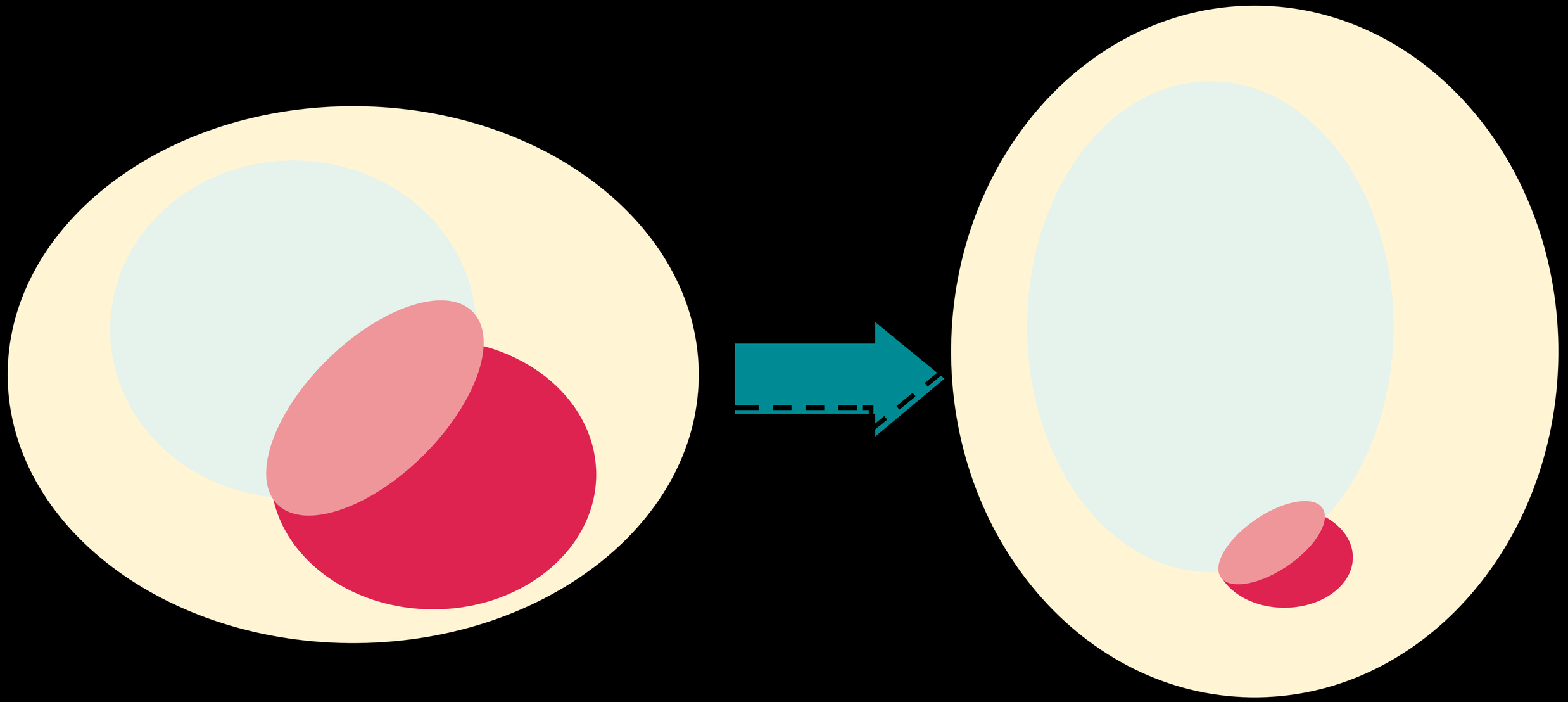


Structure of SOTIF

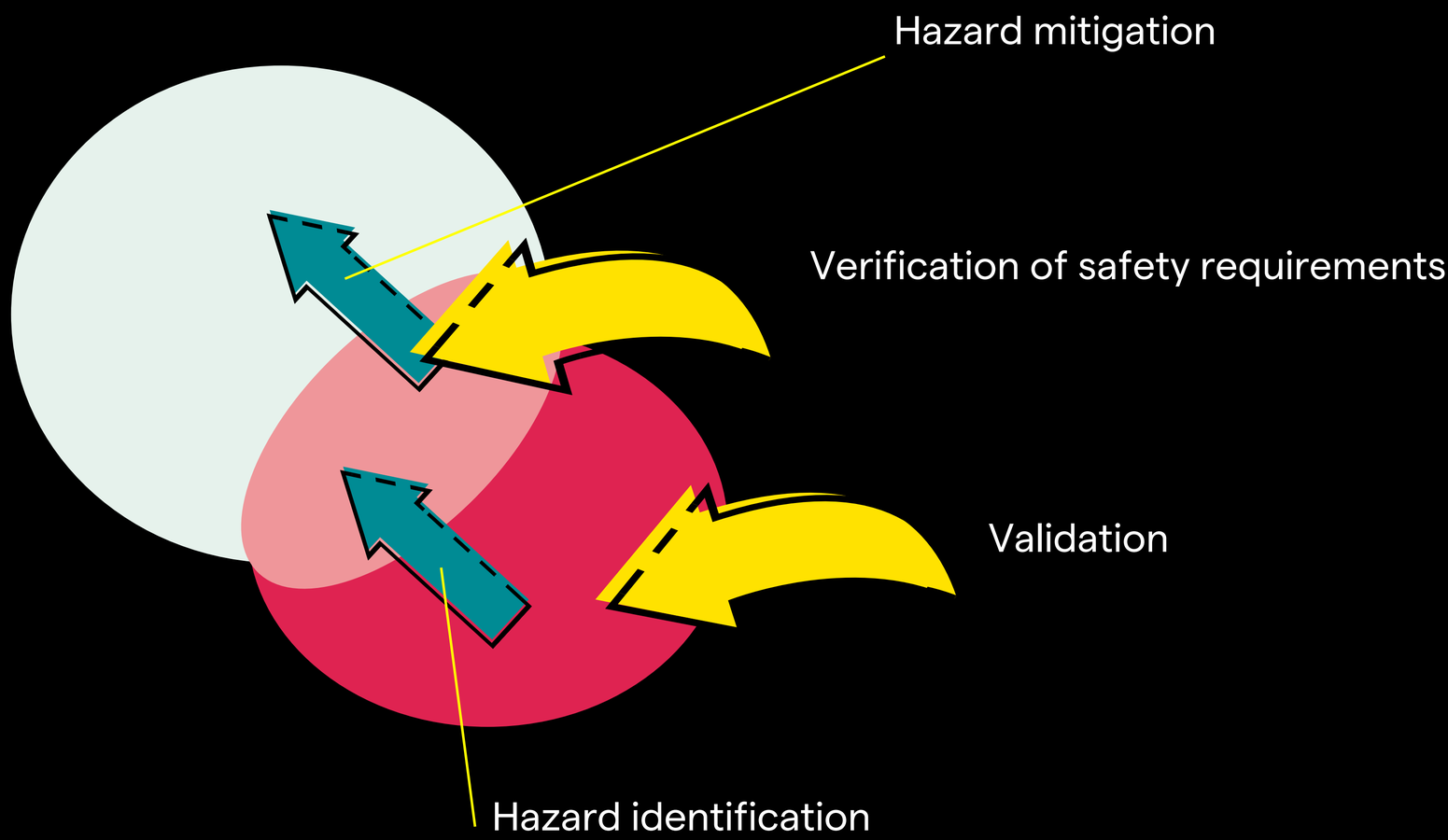


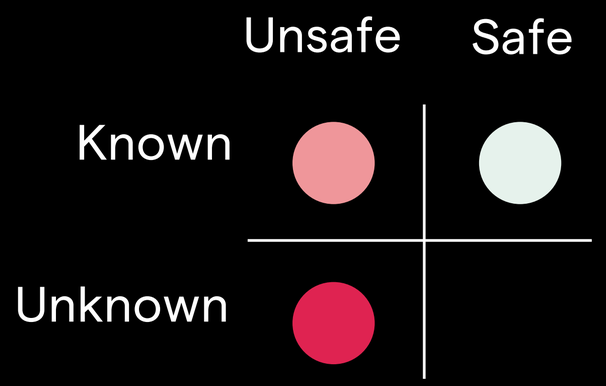


Goal of the SOTIF process



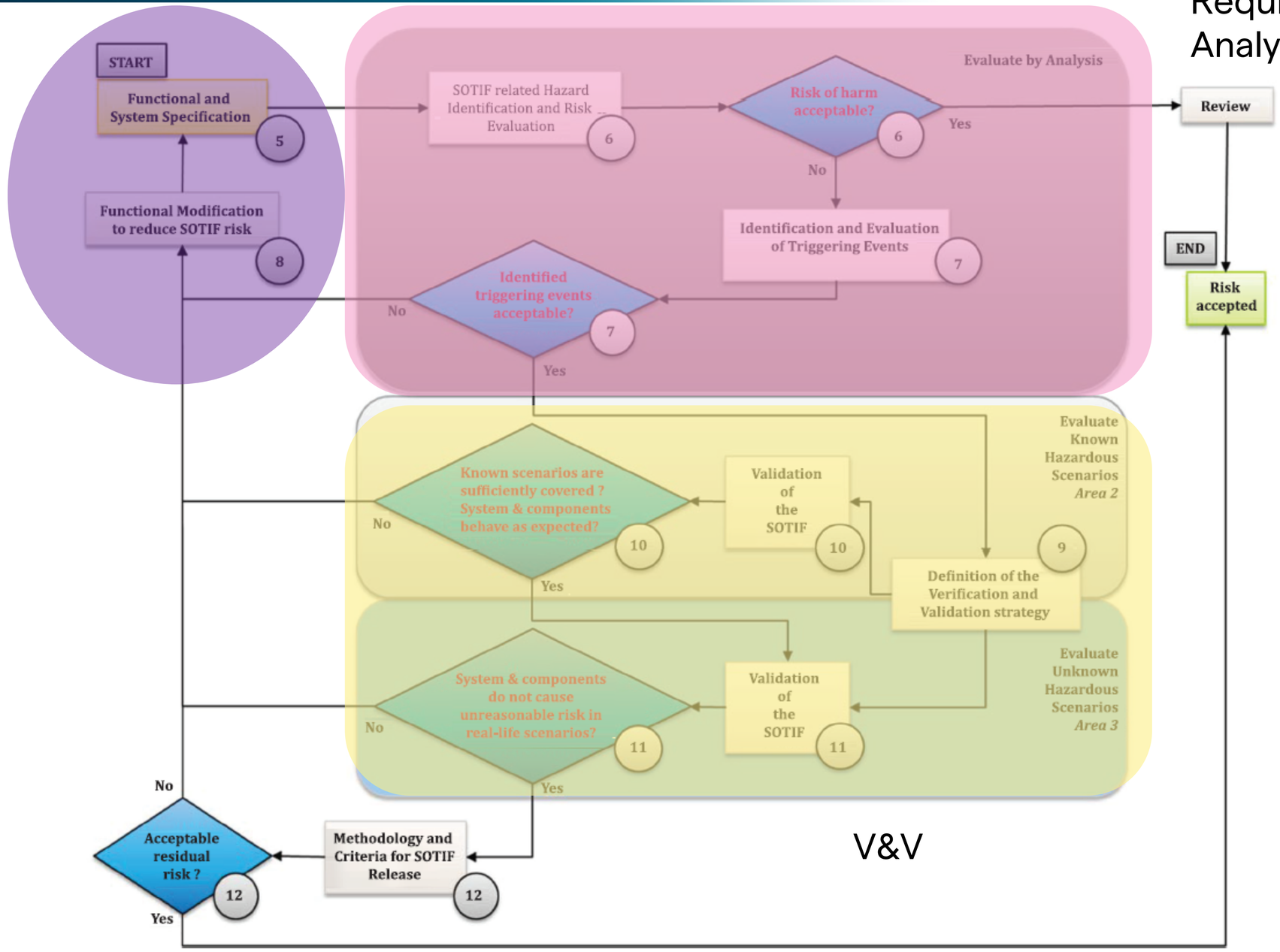
How to Minimize the Unsafe Areas?



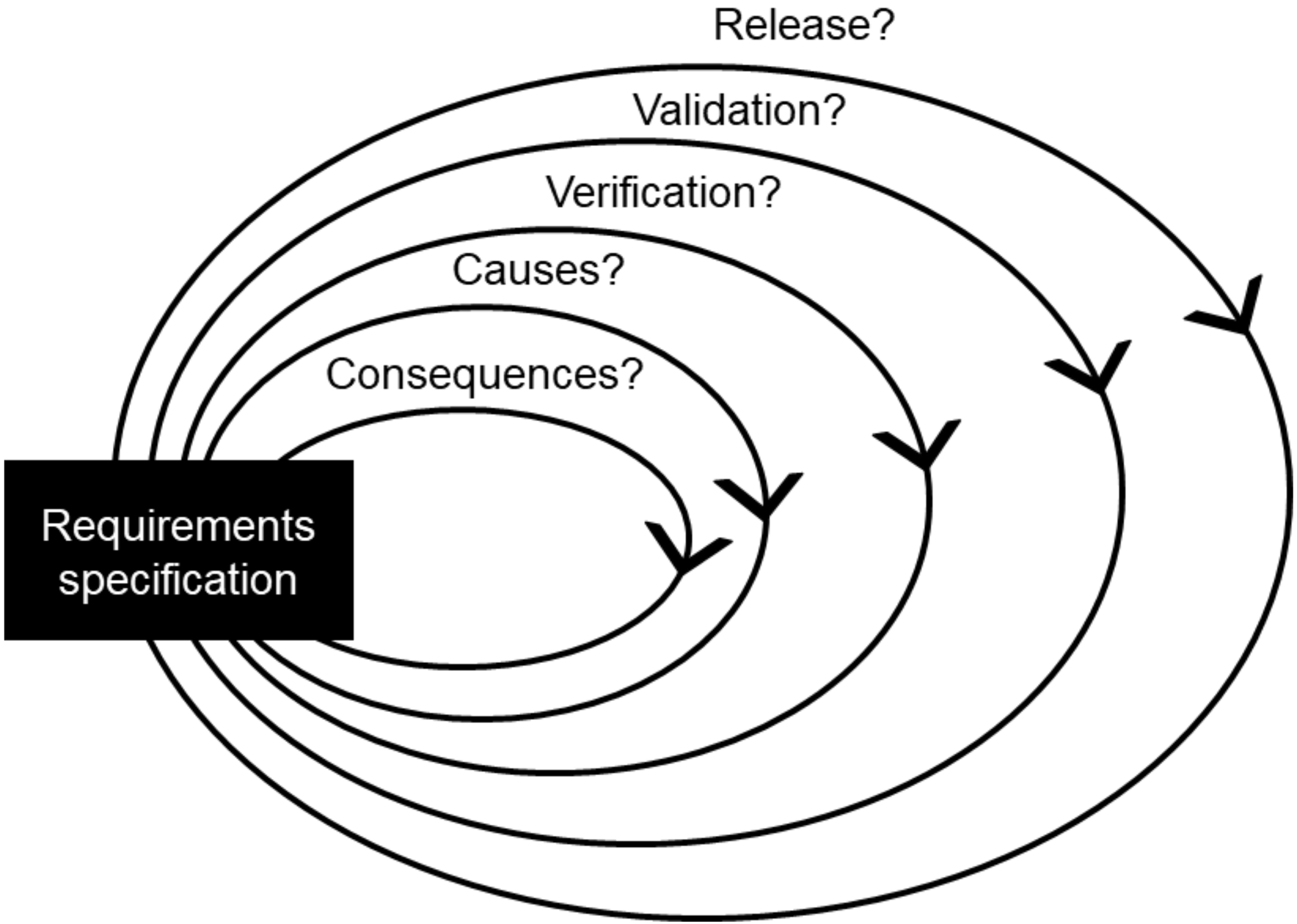


Requirements Specification

Requirements Analysis



V&V


Release?
Validation?
Verification?
Causes?
Consequences?
Requirements specification

Development of SMIRK

Open Source ADAS MVP

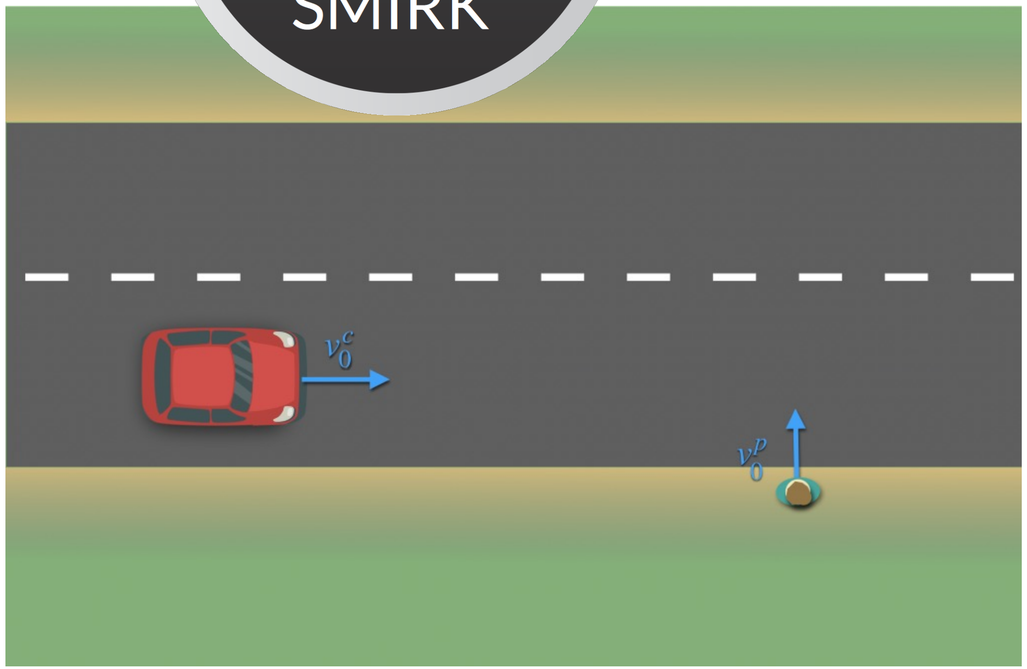
- In ESI Pro-SiVIC

- Pedestrian emergency braking

- Mono-camera and radar

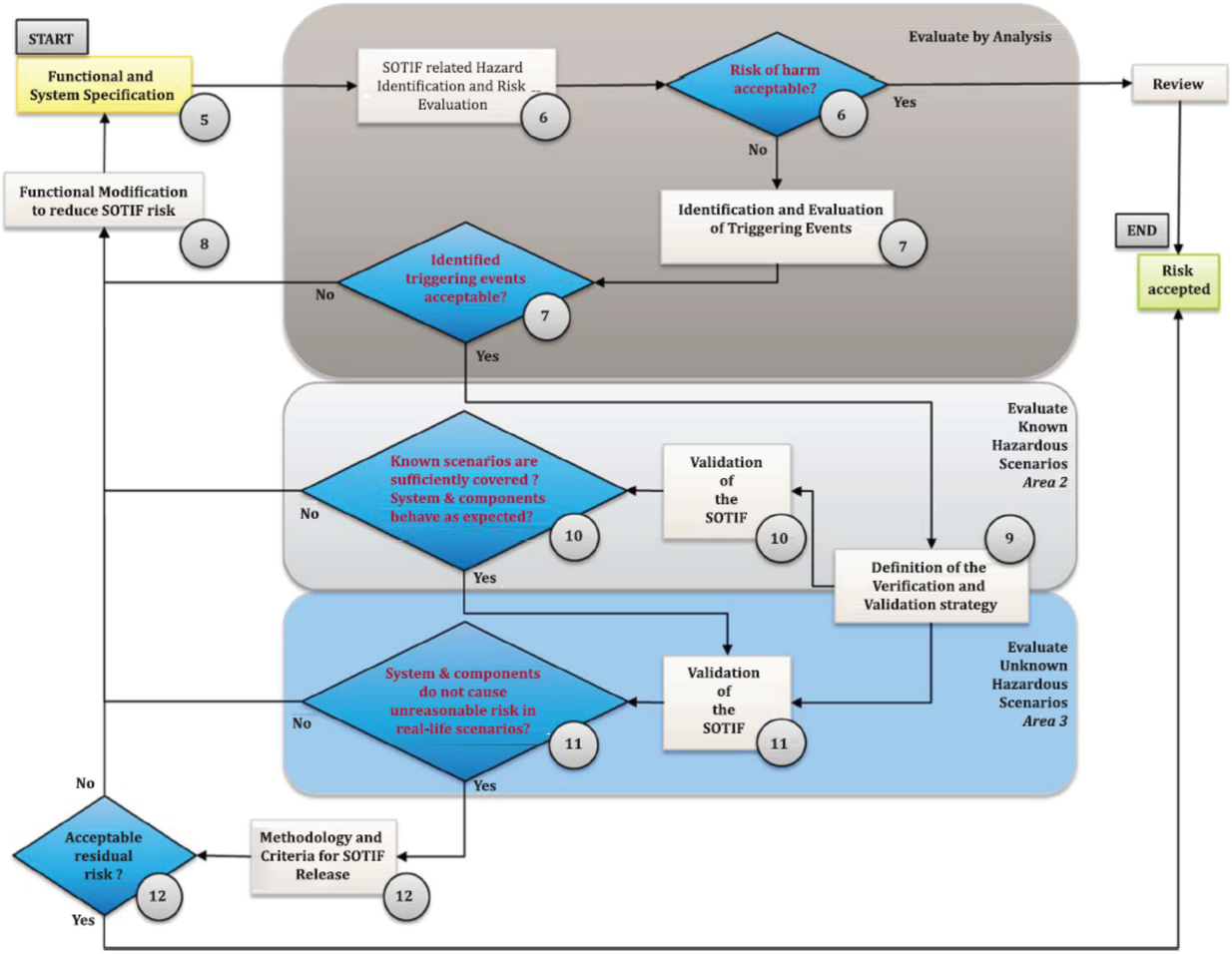
- ML-based pedestrian recognition

SMIRK

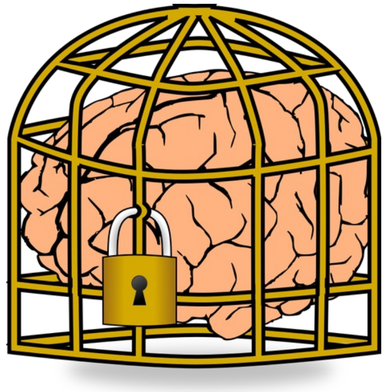


github.com/RI-SE/smirk

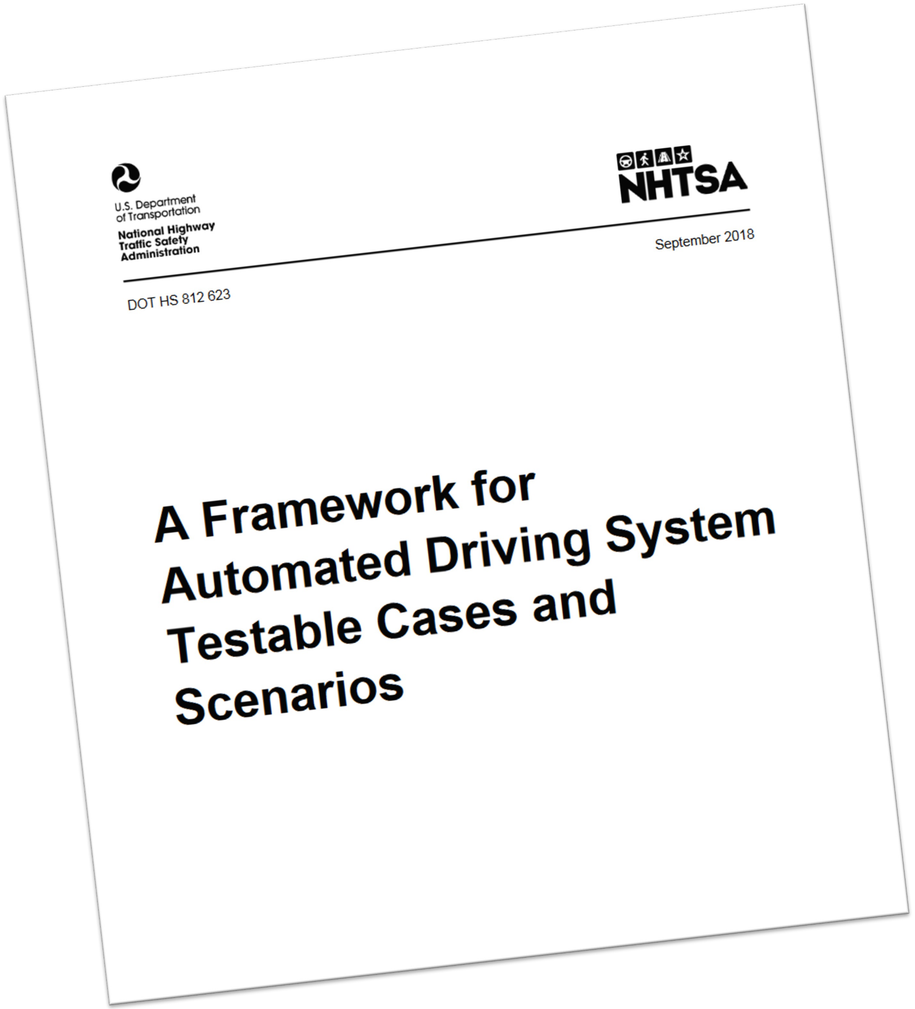
Followed the SOTIF process



Primary hazard to tackle: False postives



MVP Operational Design Domain



U.S. Department of Transportation
National Highway Traffic Safety Administration

NHTSA

September 2018

DOT HS 812 623

A Framework for Automated Driving System Testable Cases and Scenarios


ODD Elements
Physical Infrastructure
Roadway Types
highway, urban, rural, parking, multi-lane, single lane, on/off ramps, intersections, roundabouts, ...
Roadway Surfaces
asphalt, concrete, mixed, grating, brick, dirt, gravel, ...
Roadway Edges
line markers, temporary line markers, shoulder, concrete barriers, rails, cones, ...
Operational Constraints
Objects
Straight rural road, good conditions, single pedestrian
Environmental Conditions
Weather
wind, rain, snow, sleet, temperature, ...
Weather-Induced Roadway Conditions
standing water, icy roads, snow on road, ...
Particulate Matter
fog, smoke, smog, dust, ...
Illumination
day, sun front-lighting, dawn, dusk, night, street lights, headlights, oncoming vehicle lights, ...

Requirements engineering...

System requirements

3.3 Machine Learning Safety Requirements [H]

This section refines SYS-SAF-REQ into two separate requirements corresponding to false positives and false negatives, respectively.

- **SYS-ML-REQ1: The pedestrian recognition component shall detect pedestrians if the radar tracking component returns TTC < 4s for the corresponding object.**
- **SYS-ML-REQ2: The pedestrian recognition component shall reject input that does not resemble the training data.**

3.3.1 Performance Requirements

This section specifies performance requirements corresponding to the ML safety requirements with a focus on quantitative targets for the pedestrian recognition component. All requirements below are restricted to pedestrians on or close to the road.

- **SYS-PER-REQ1: The pedestrian recognition component shall identify pedestrians with an accuracy of 0.93 when they are within 50 meters.**
- **SYS-PER-REQ2: The false negative rate of the pedestrian recognition component shall not exceed 7% for pedestrians when they are detected by the radar tracking component within 50 meters.**
- **SYS-PER-REQ3: The false positive rate of the pedestrian recognition component shall not exceed 0.01% for objects detected by the radar tracking component with a TTC < 4s**
- **SYS-PER-REQ4: In a sequence of images from a video feed any pedestrian to be detected shall not be missed in more than 1 out of 5 frames.**
- **SYS-PER-REQ5: The pedestrian recognition component shall determine the position of pedestrians within 50 cm of their actual position.**
- **SYS-PER-REQ6: The pedestrian recognition component shall allow an inference speed of at least 10 FPS on the target platform.**

Data requirements

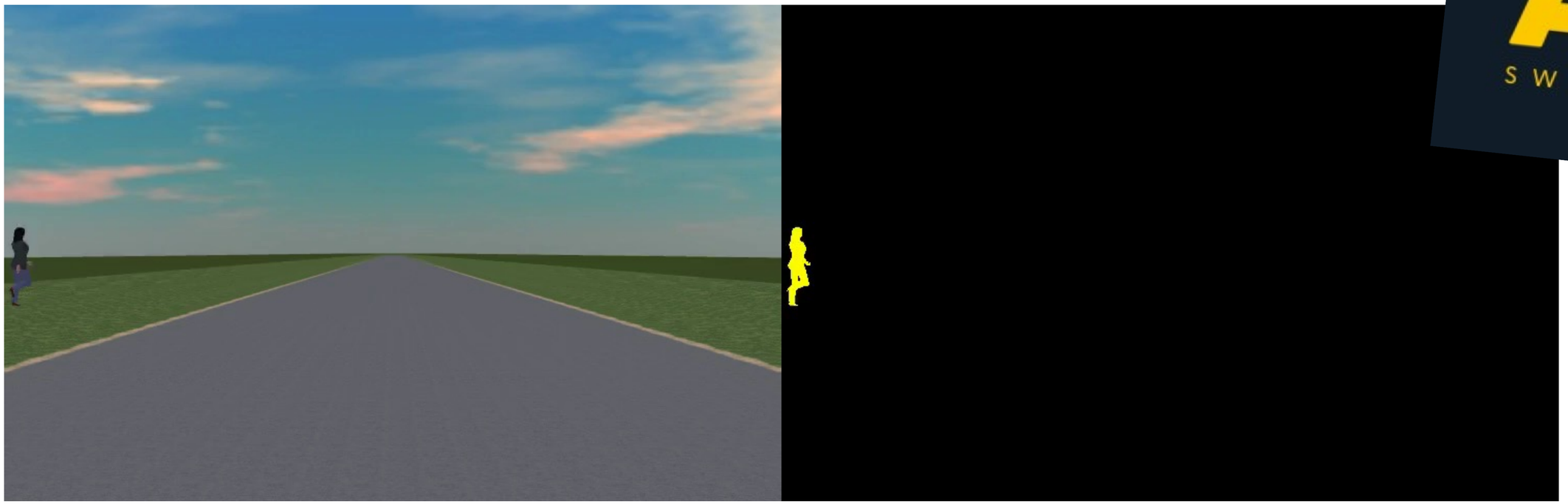
2.1 Relevant

This desideratum considers the intersection between the dataset and the supported dynamic driving task in the ODD. The SMIRK training data will not cover operational environments that are outside of the ODD, e.g., images collected in heavy snowfall.

- DAT-REL-REQ1: All data samples shall represent images of a road from the perspective of a vehicle.
- DAT-REL-REQ2: The format of each data sample shall be representative of that which is captured using sensors deployed on the ego vehicle.
- DAT-REL-REQ3: Each data sample shall assume sensor positioning representative of the positioning used on the ego vehicle.
- DAT-REL-REQ4: All data samples shall represent images of a road that corresponds to the ODD.
- DAT-REL-REQ5: All data samples containing pedestrians shall include one single pedestrian.
- DAT-REL-REQ6: Pedestrians included in data samples shall be of a type that may appear in the ODD.
- DAT-REL-REQ7: All data samples representing non-pedestrian OOD objects shall be of a type that may appear in the ODD.

Generate Training Data in ESI Pro-SiVIC

Synthetic data that cover the Operational Design Domain



https://github.com/RI-SE/smirk/tree/main/pedestrian-generator

CodeScene


```
C:\Users\eitn35\Desktop\smirk>python src/test.py
Loading simple_aeb.script
pedestrian_observer ID : 23579
DDS configuration: .json file not found, using .in
i file
(13960|15324) NOTICE: using DCPSRTISerialization v
alue from command option (overrides value if it's
in config file).
Creating DDS subscriber for pedestrian_observer (i
d=23579) (topic Name: manObserver)
main_camera/cam ID : 59726
DDS configuration: .json file not found, using .in
i file
Creating DDS subscriber for main_camera/cam (id=59
726) (topic Name: camera)
radar/radar ID : 101602
DDS configuration: .json file not found, using .in
i file
Creating DDS subscriber for radar/radar (id=101602
) (topic Name: radar)
collision_observer ID : 100492
DDS configuration: .json file not found, using .in
i file
Creating DDS subscriber for collision_observer (id
=100492) (topic Name: distanceObserver)
Subscription Matched Status changed
Subscription Matched Status changed
ego_car/car ID : 21661
Subscription Matched Status changed
DDS configuration: .json file not found, using .in
i file
Subscription Matched Status changed
Creating DDS subscriber for ego_car/car (id=21661)
 (topic Name: carOrder)
ego_car/car ID : 21661
DDS configuration: .json file not found, using .in
i file
Subscription Matched Status changed
Creating DDS publisher for ego_car/car (id=21661)
(topic Name: carOrder)

Loading SSD from tensorflow hub.
Model loaded.
Model warmup.
Model ready.
```

The SMIRK MVP

CodeScene

Safety Case Development

UNIVERSITY of York

Assuring Autonomy International Programme

About AAIP | Work with us | Research | Projects | Body of Knowledge | Training and ed

> Assuring Autonomy International Programme

ASSURING AUTONOMY
INTERNATIONAL PROGRAMME

Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS)

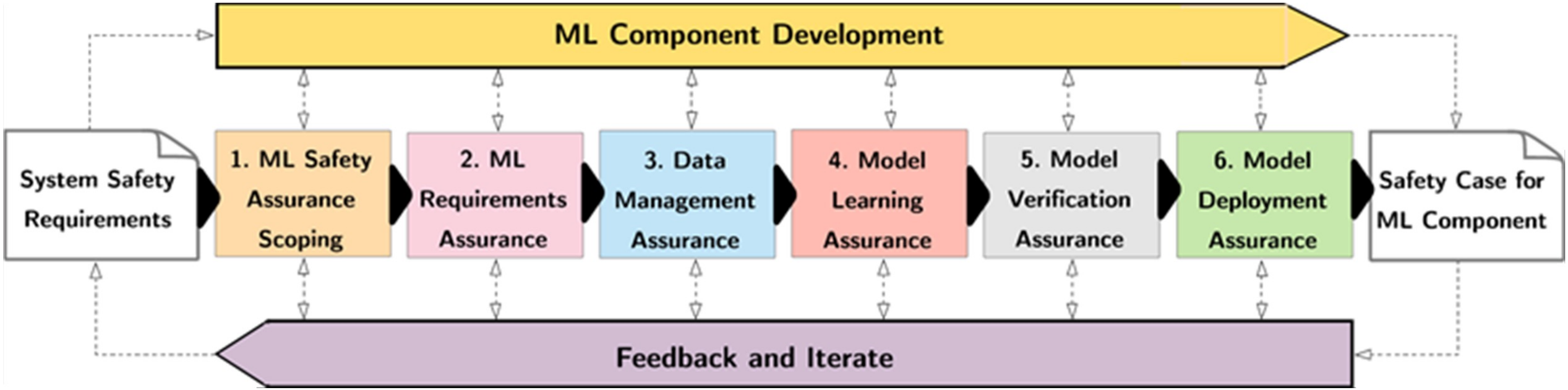
Richard Hawkins, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu and Ibrahim Habli.

Assuring Autonomy International Programme (AAIP)
University of York

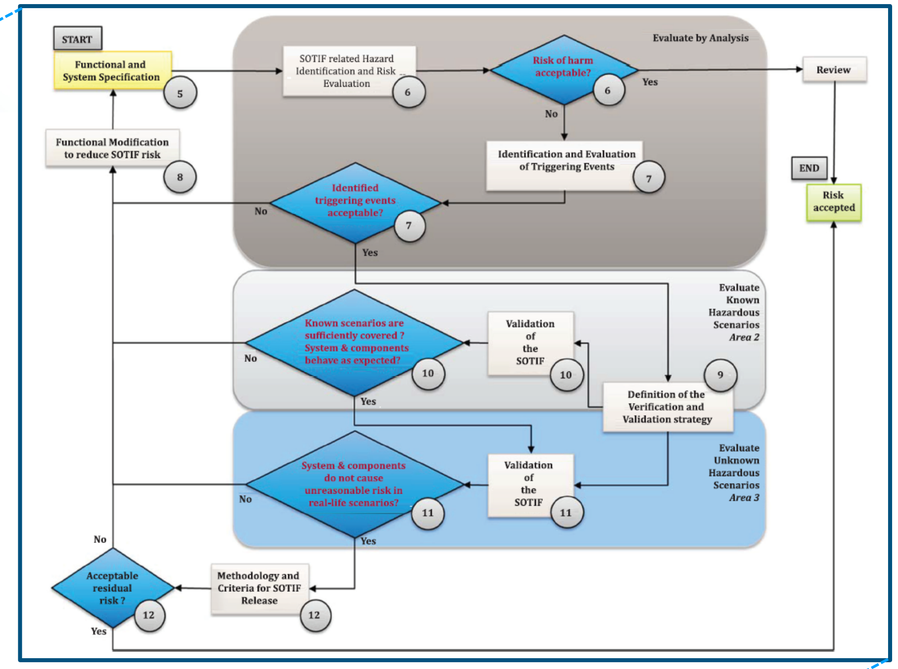
Version 1.1, March 2021

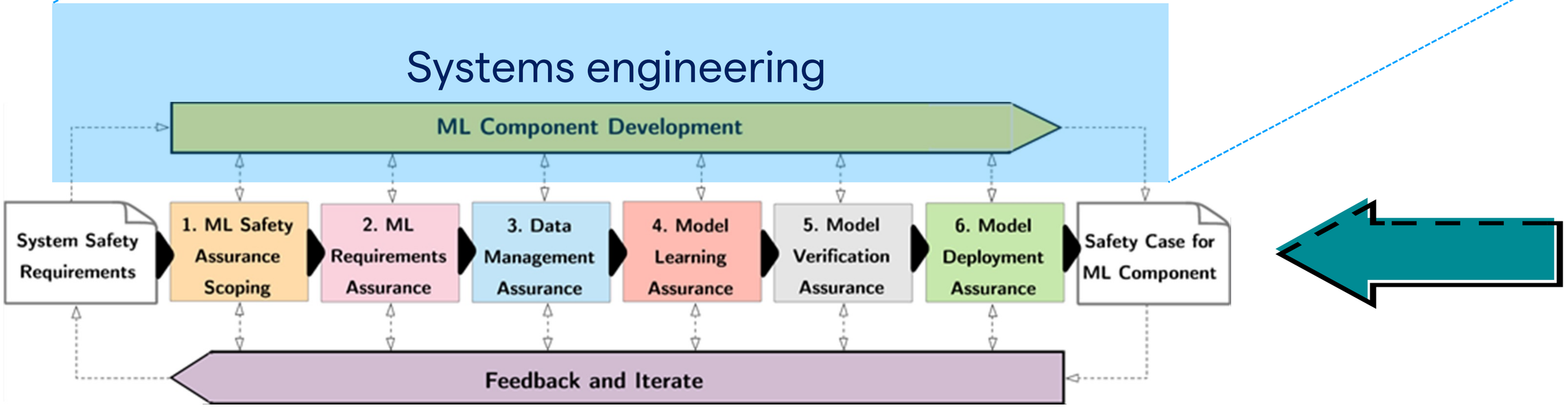
Contact : firstname.lastname@york.ac.uk.

Follow the AMLAS process

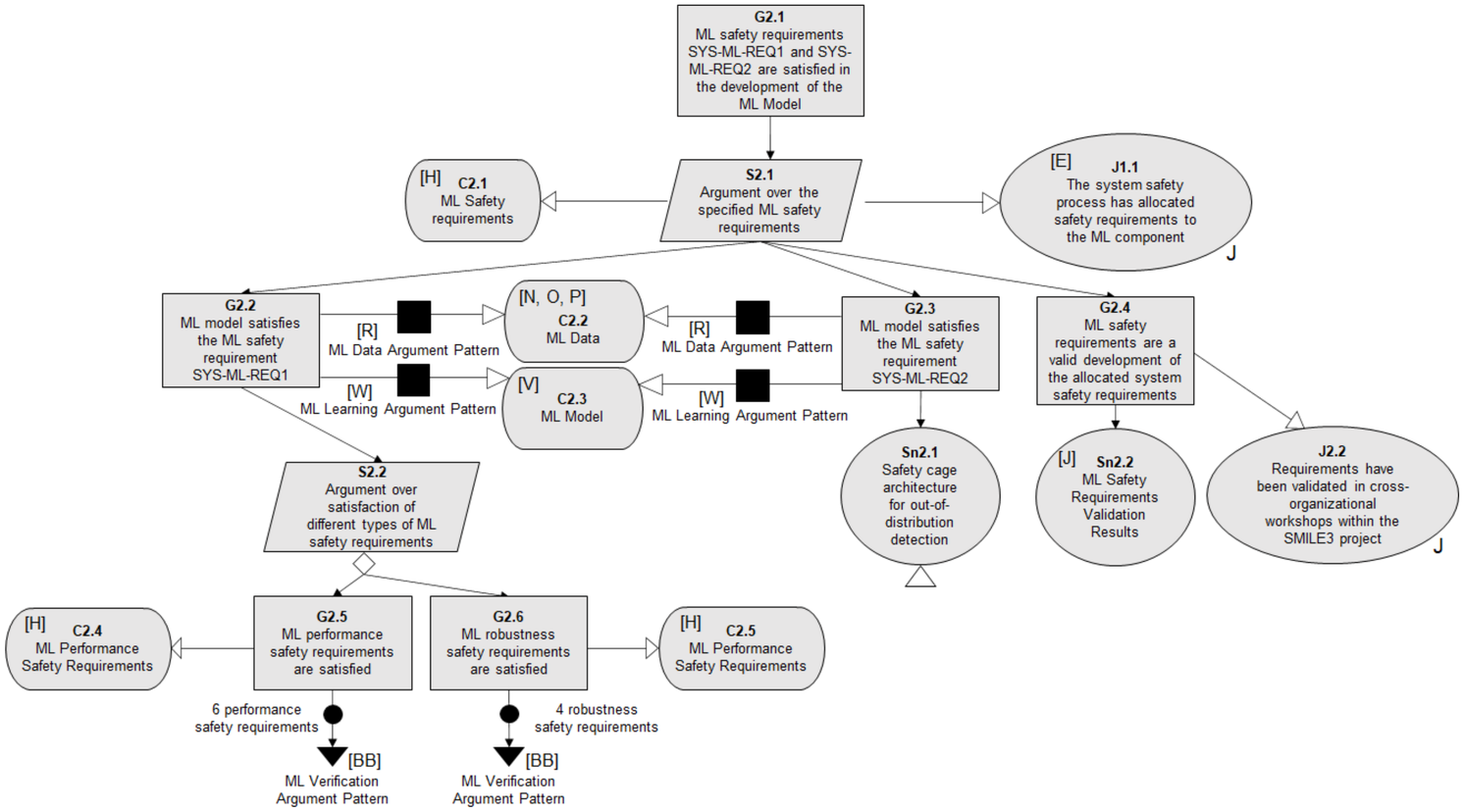


SOTIF + AMLAS


START
Functional and System Specification
5
SOTIF related Hazard Identification and Risk Evaluation
6
Risk of harm acceptable?
6
Yes
No
Evaluate by Analysis
Review
Functional Modification to reduce SOTIF risk
8
Identification and Evaluation of Triggering Events
7
END
Risk accepted
Identified triggering events acceptable?
7
No
Yes
Evaluate Known Hazardous Scenarios Area 2
Known scenarios are sufficiently covered ? System & components behave as expected?
10
No
Validation of the SOTIF
10
9
Definition of the Verification and Validation strategy
Evaluate Unknown Hazardous Scenarios Area 3
System & components do not cause unreasonable risk in real-life scenarios?
11
No
Validation of the SOTIF
11
Yes
No
Acceptable residual risk ?
12
Yes
Methodology and Criteria for SOTIF Release
12

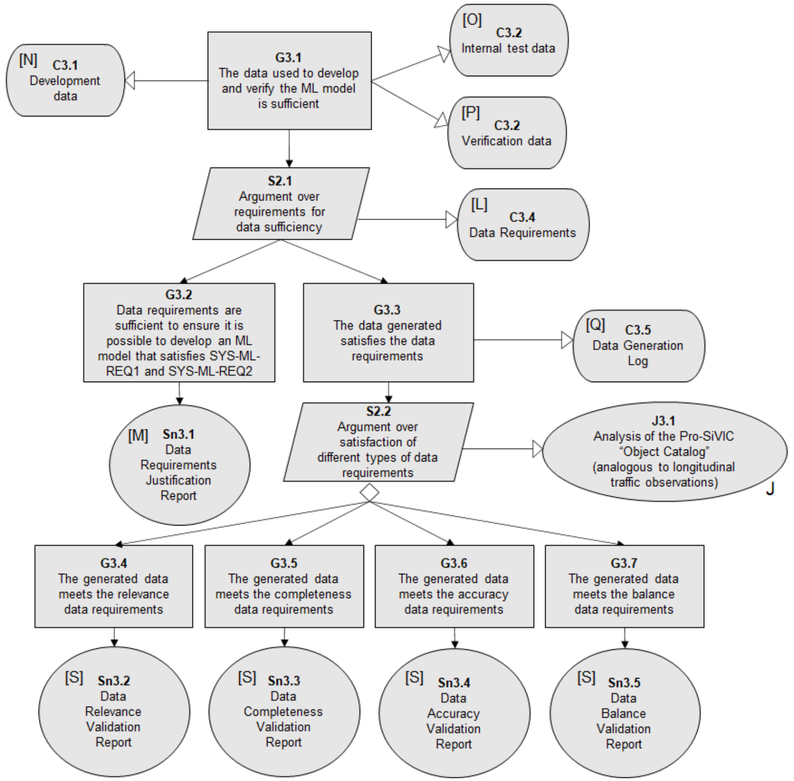

Systems engineering
ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate

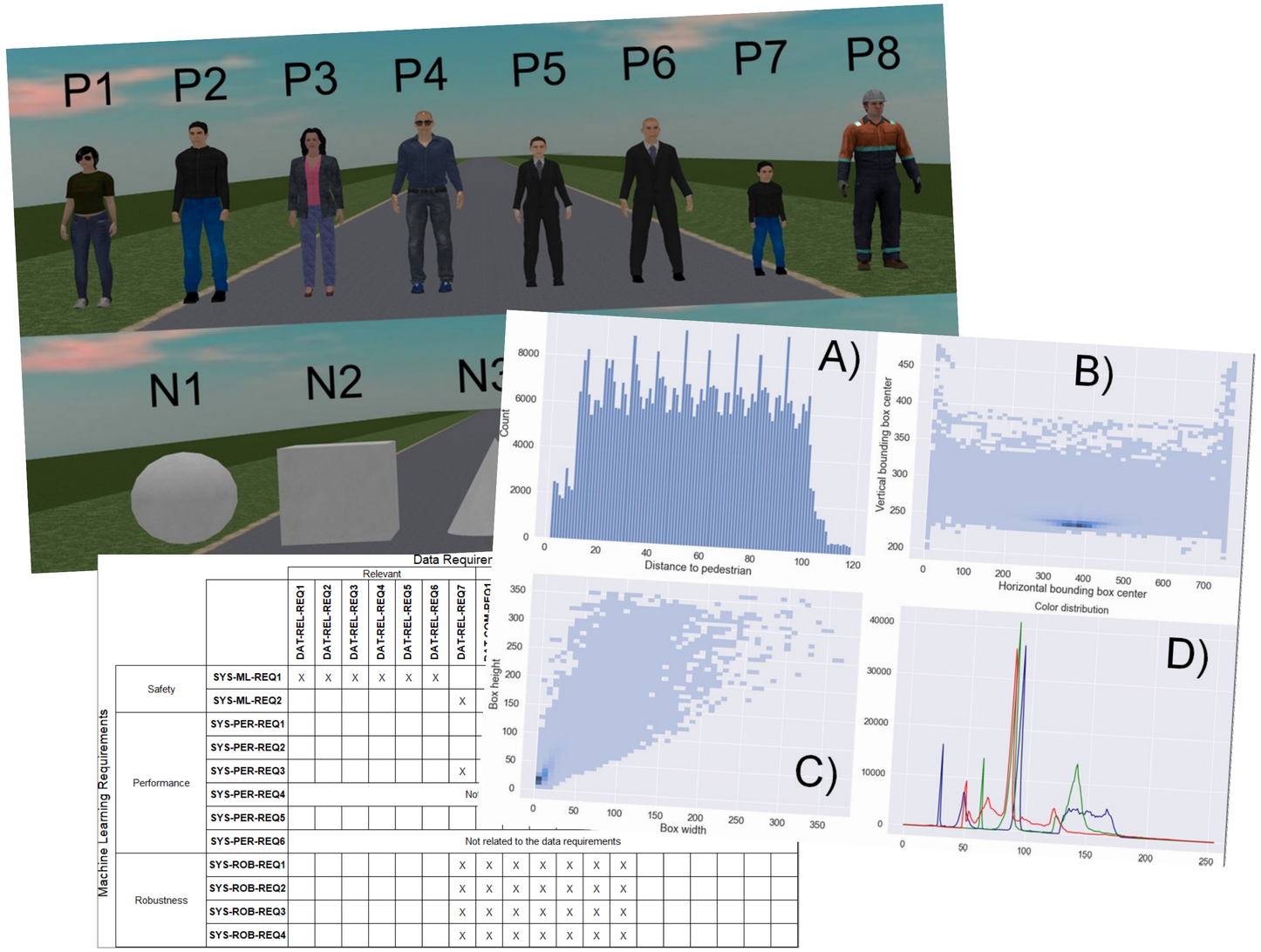
2. Requirements Assurance



Formal inspections

3. Data Management Assurance

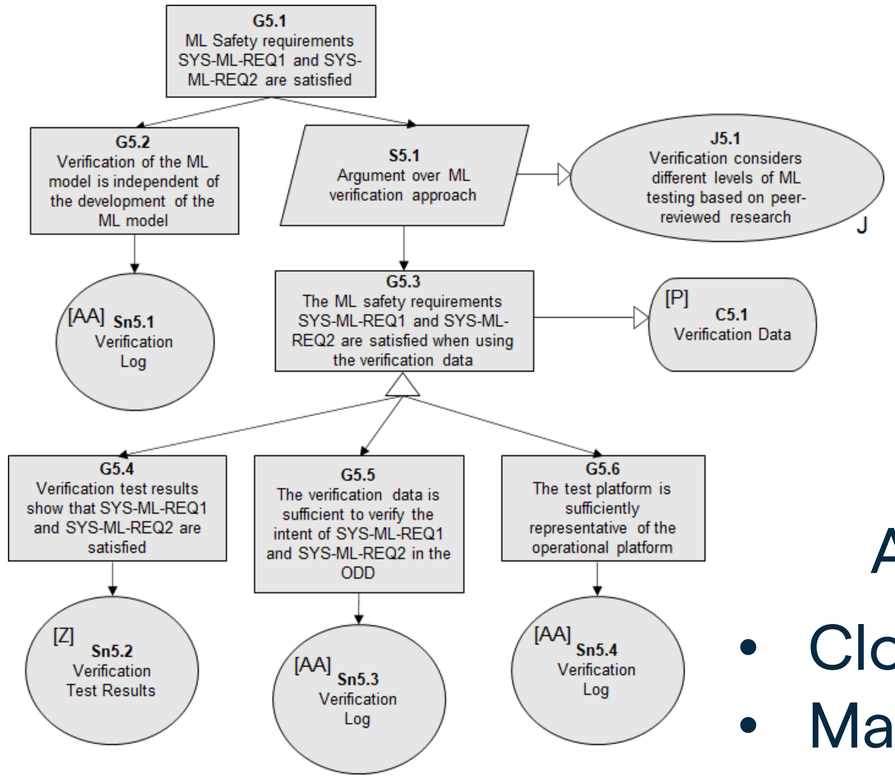


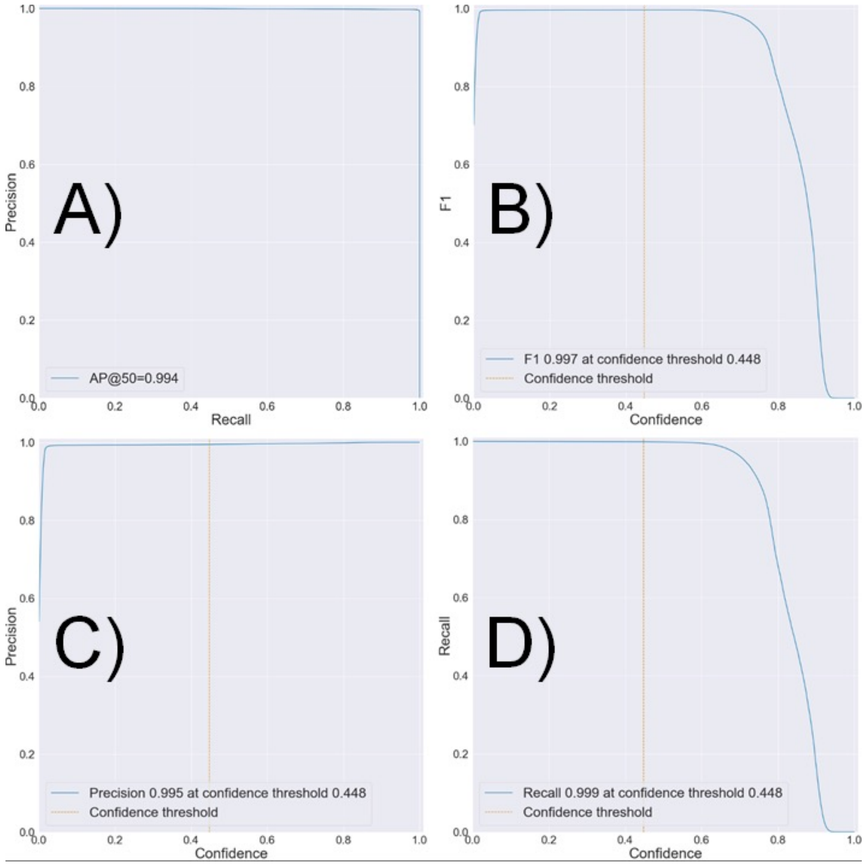


1) Relevance 2) Completeness

3) Accuracy 4) Balance

5. Model Verification Assurance



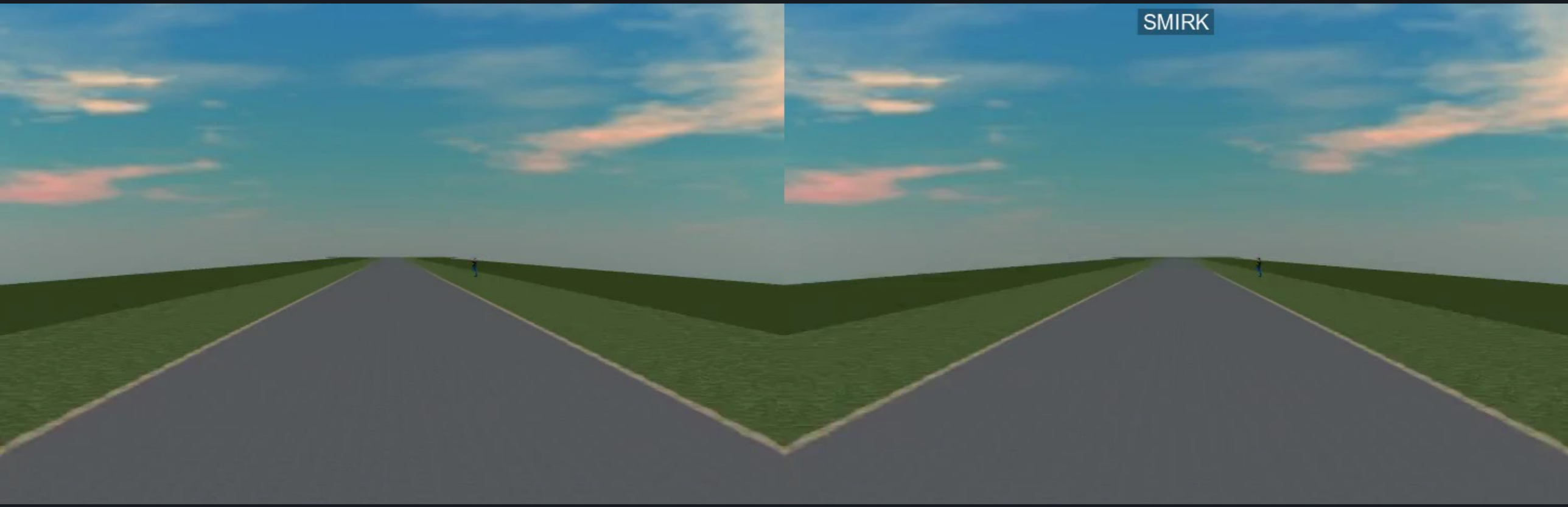


Analysis of subsets

- Close/Far away
- Male/Female/Children
- Standing/Walking/Running
- ...

Wrap-up and Lessons Learned

Demo



Without braking

With SMIRK

Lessons Learned

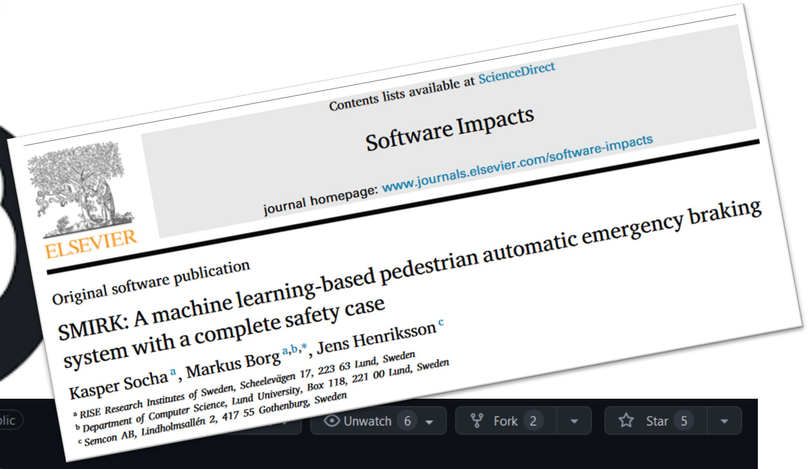
SOTIF and AMLAS compatible

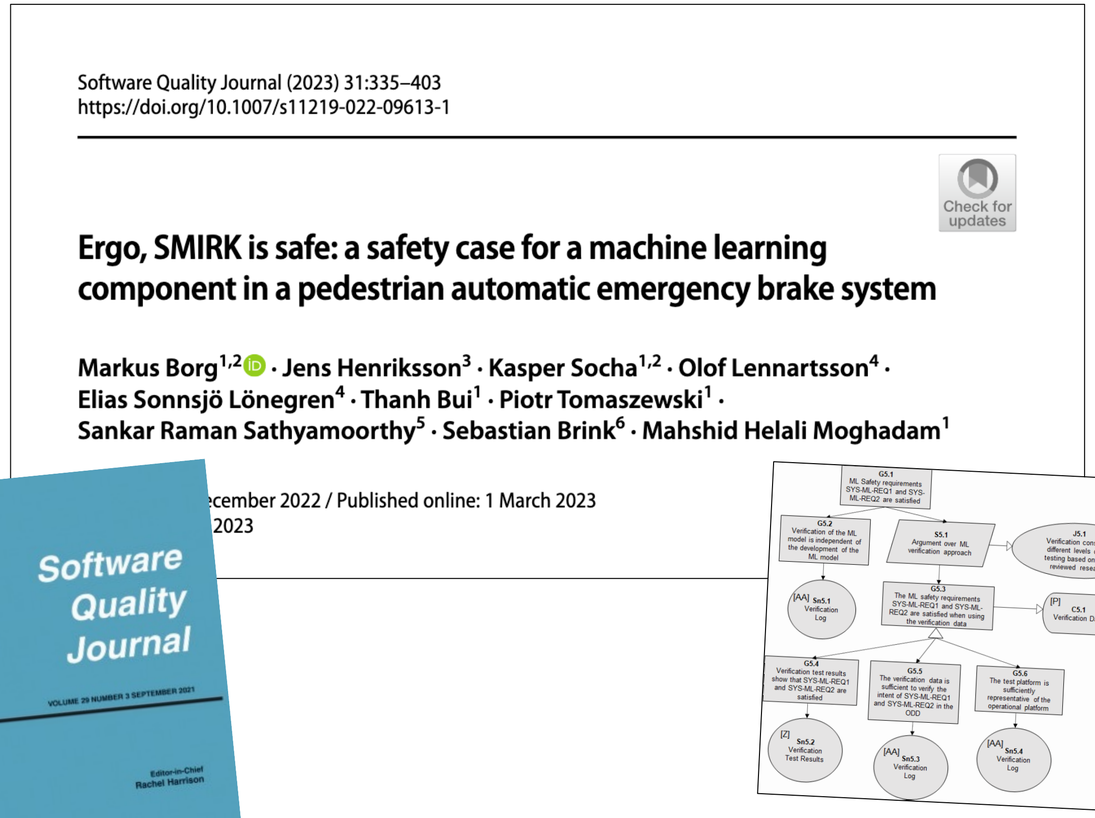
Requirements engineering at the core

Massive safety case for a minimal example

Evaluation of object detection models is hard

Open ML safety case



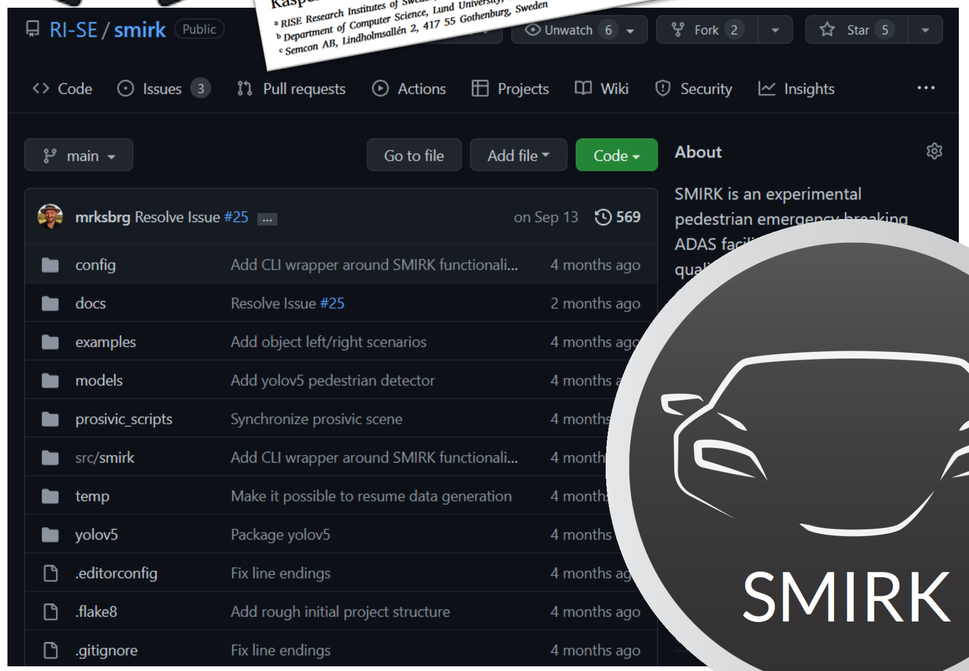

Software Quality Journal (2023) 31:335–403
https://doi.org/10.1007/s11219-022-09613-1

Check for updates

Ergo, SMIRK is safe: a safety case for a machine learning component in a pedestrian automatic emergency brake system

Markus Borg[1,2] · Jens Henriksson[3] · Kasper Socha[1,2] · Olof Lennartsson[4] · Elias Sonnsjö Lönegren[4] · Thanh Bui[1] · Piotr Tomaszewski[1] · Sankar Raman Sathyamoorthy[5] · Sebastian Brink[6] · Mahshid Helali Moghadam[1]

ecember 2022 / Published online: 1 March 2023
2023

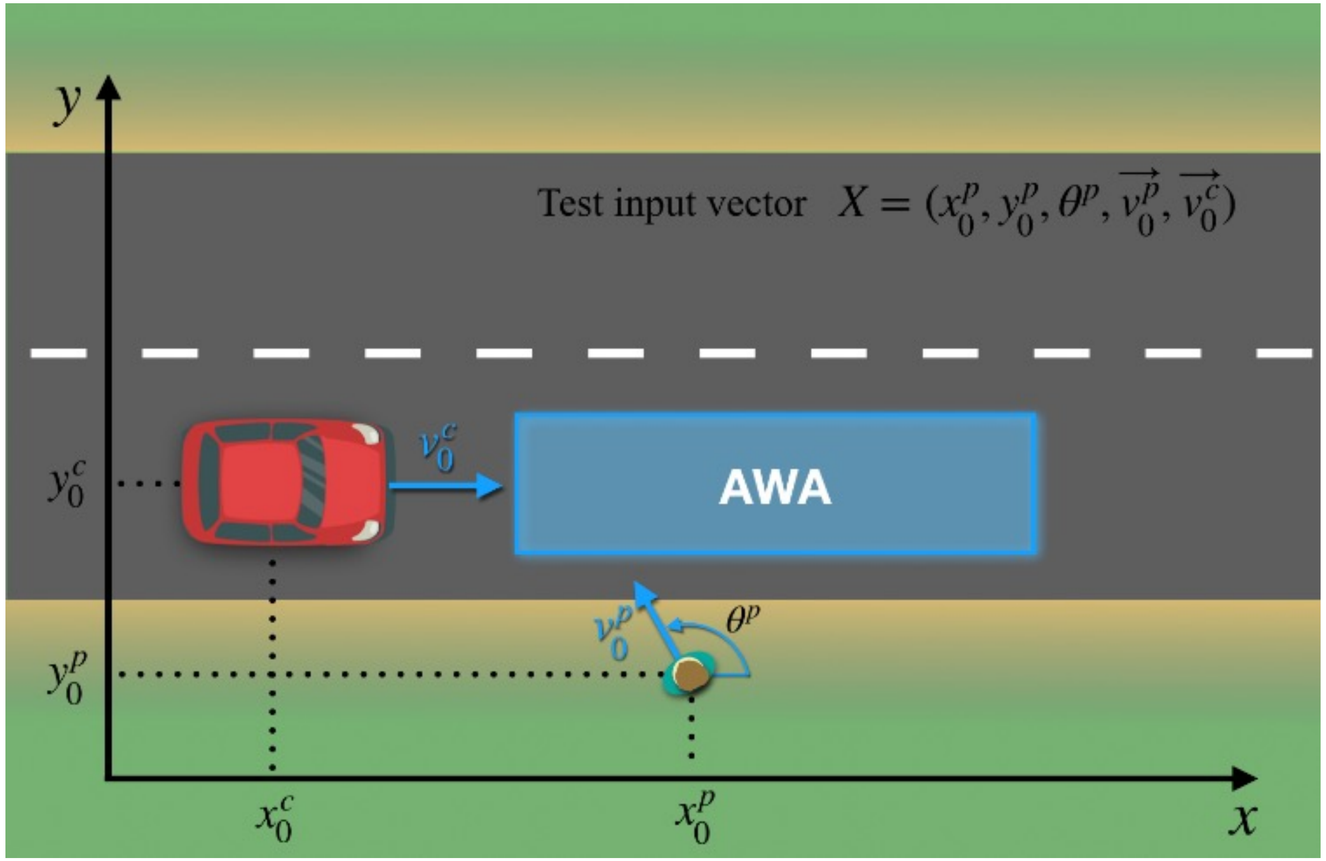


Open ML-based demonstrator

markus.borg@codescene.com

Backups

Reverse engineering from PeVi



RESEARCH-ARTICLE

Testing advanced driver assistance systems using multi-objective search and neural networks

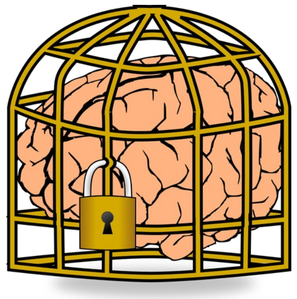
Authors: Raja Ben Abdessalem, Shiva Nejati, Lionel C. Briand, Thomas Stifter Authors Info & Affiliations

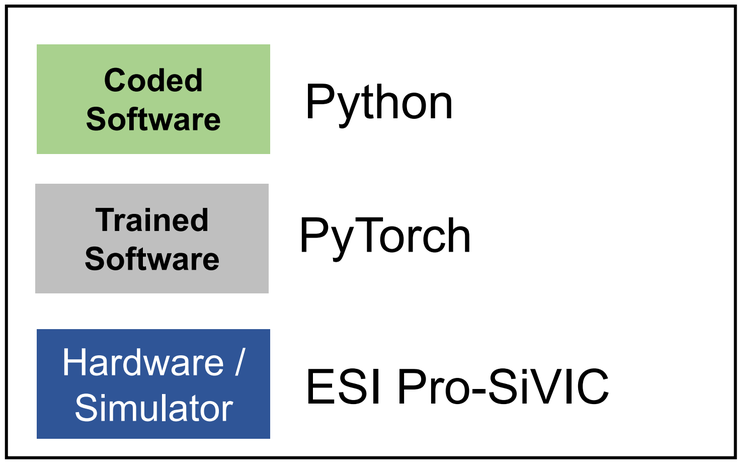
Publication: ASE 2016: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering • August 2016 • Pages 63–74 • https://doi.org/10.1145/2970276.2970311

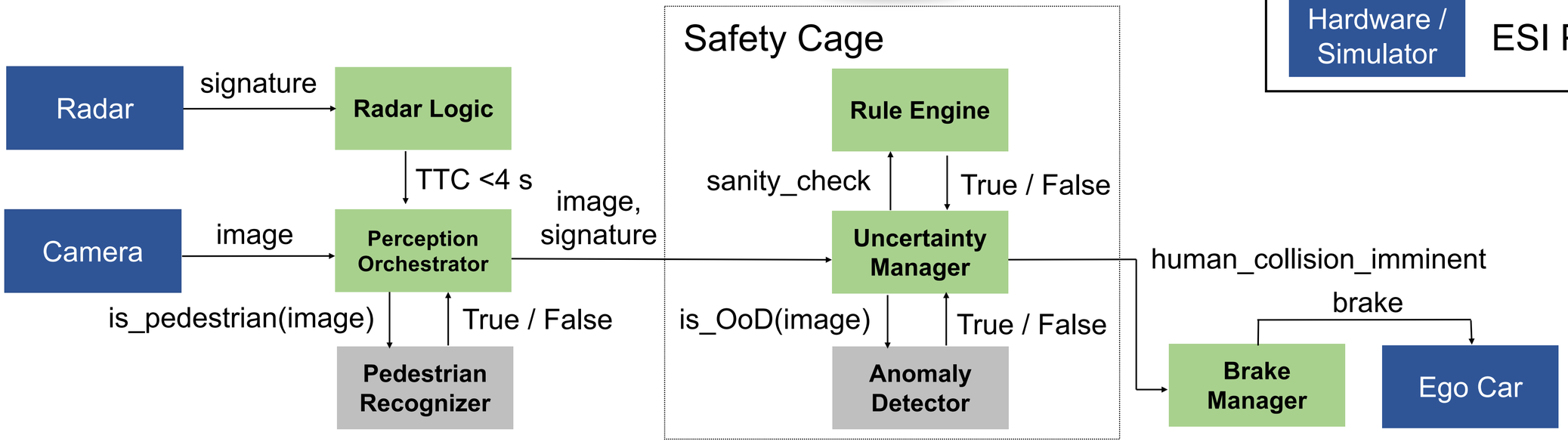
UNIVERSITÉ DU LUXEMBOURG

a sense for innovation IEE

Logical View of the SMIRK Architecture







References

- Code: https://github.com/RI-SE/smirk/
- Data: https://www.ai.se/en/data-factory/datasets/data-factory-datasets/smirk-dataset
- Demonstrator: Socha, Borg, and Henriksson, SMIRK: A Machine Learning-Based Pedestrian Automatic Emergency Braking System with a Complete Safety Case, Software Impacts, Volume 13, 2022.
- Safety Case: Borg, Henriksson, Socha, Lennartsson, Lönegren Sonnsjö, Bui, Tomaszewski, Sathyamoorthy, Brink, and Helali, Ergo, Ergo, SMIRK is Safe: A Safety Case for a Machine Learning Component in a Pedestrian Automatic Emergency Brake System, https://arxiv.org/abs/2204.07874

- Ashmore, Calinescu, and Paterson, Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges, *ACM Computing Surveys*, 54(5), 2021.
- Ben Abdessalem, Nejati, Briand, and Stifter, Testing Advanced Driver Assistance Systems Using Multi-objective Search and Neural Networks, In *Proc. of the 31st Int'l. Conf. on Automated Software Engineering*, 2016.
- Thorn, Kimmel, Chaka *et al.*, A Framework for Automated Driving System Testable Cases and Scenarios, National Highway Traffic Safety Administration US Department of Transportation, 2018.
- Hawkins, Paterson, Picardi *et al.*, Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS), v1.1, Assuring Autonomy Int'l. Programme, University of York, 2021.