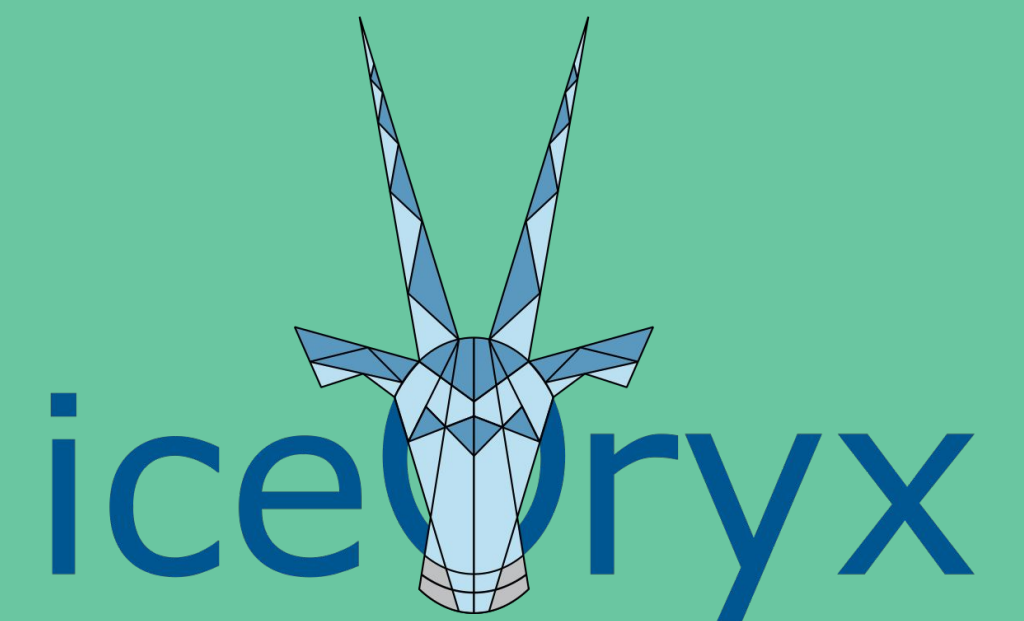


# Safety and Open Source, Oh My?

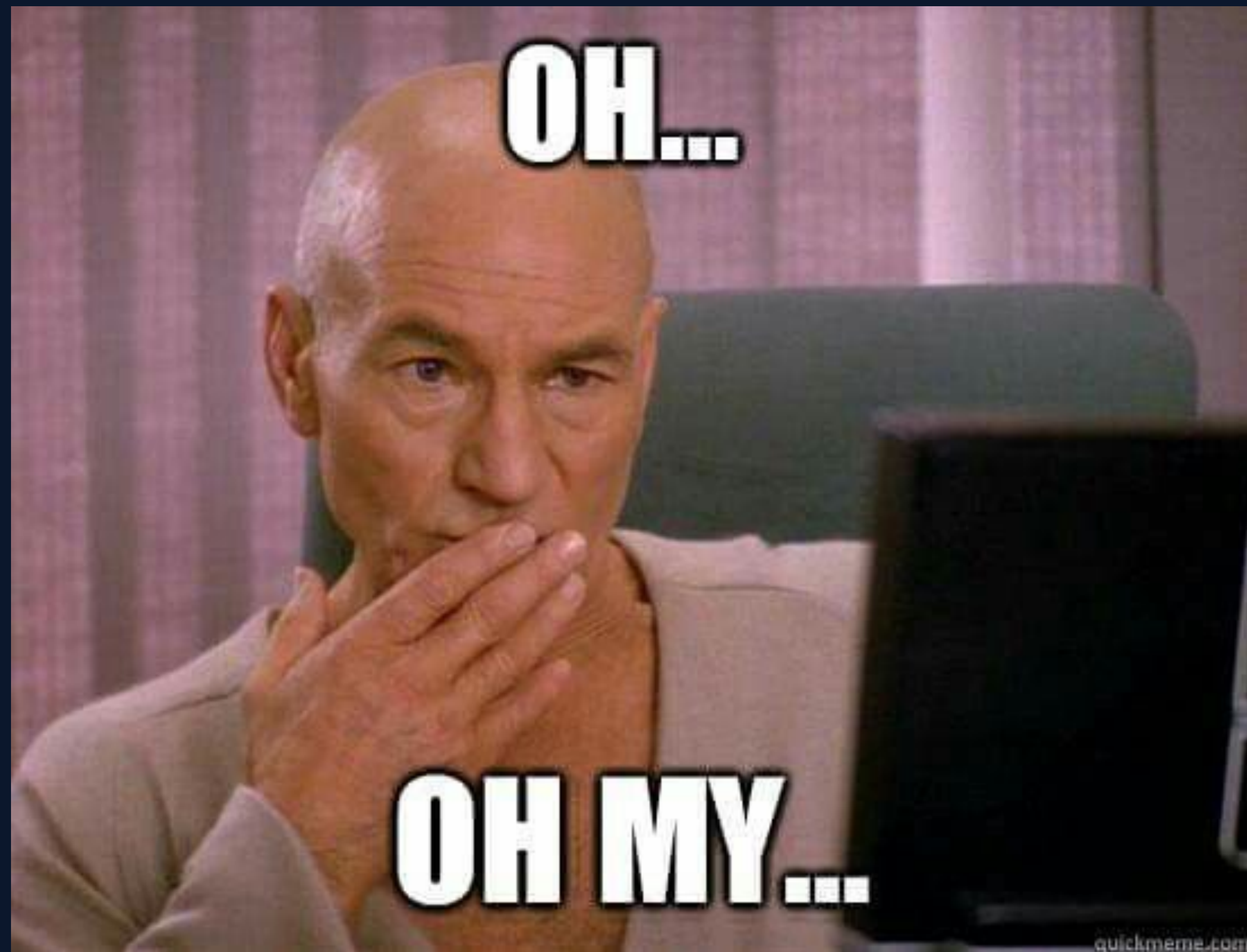
FOSDEM 2021

Simon Hoinkis  
Christian Eltzschig

**Apex.AI**



# Safety and Open Source, Oh My?





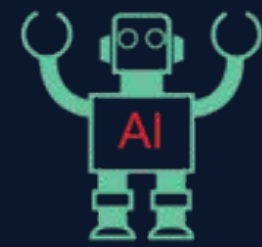
# Safety and Open Source, Oh My?

## Agenda

1. Motivation
2. Introduction
3. Why Processes Are Your Friends
4. Typical Automotive Software Development Process
  - a. V-Model and ASPICE®
  - b. ISO 26262
5. Goals
6. Many Tools, So Safety
7. Review of Tools & Processes
  - a. Community Contributions
  - b. Testing Software
8. Lessons Learned
9. Outlook

# Safety and Open Source, Oh My?

## Motivation



“Transparency builds trust”



# Safety and Open Source, Oh My?

## Motivation

Board member of Bosch, Harald Kröger [said about automotive software development](#):

*“I don’t think it’s sensible, that everyone works alone on this challenge”*

⇒ FOSS is the right choice :-)

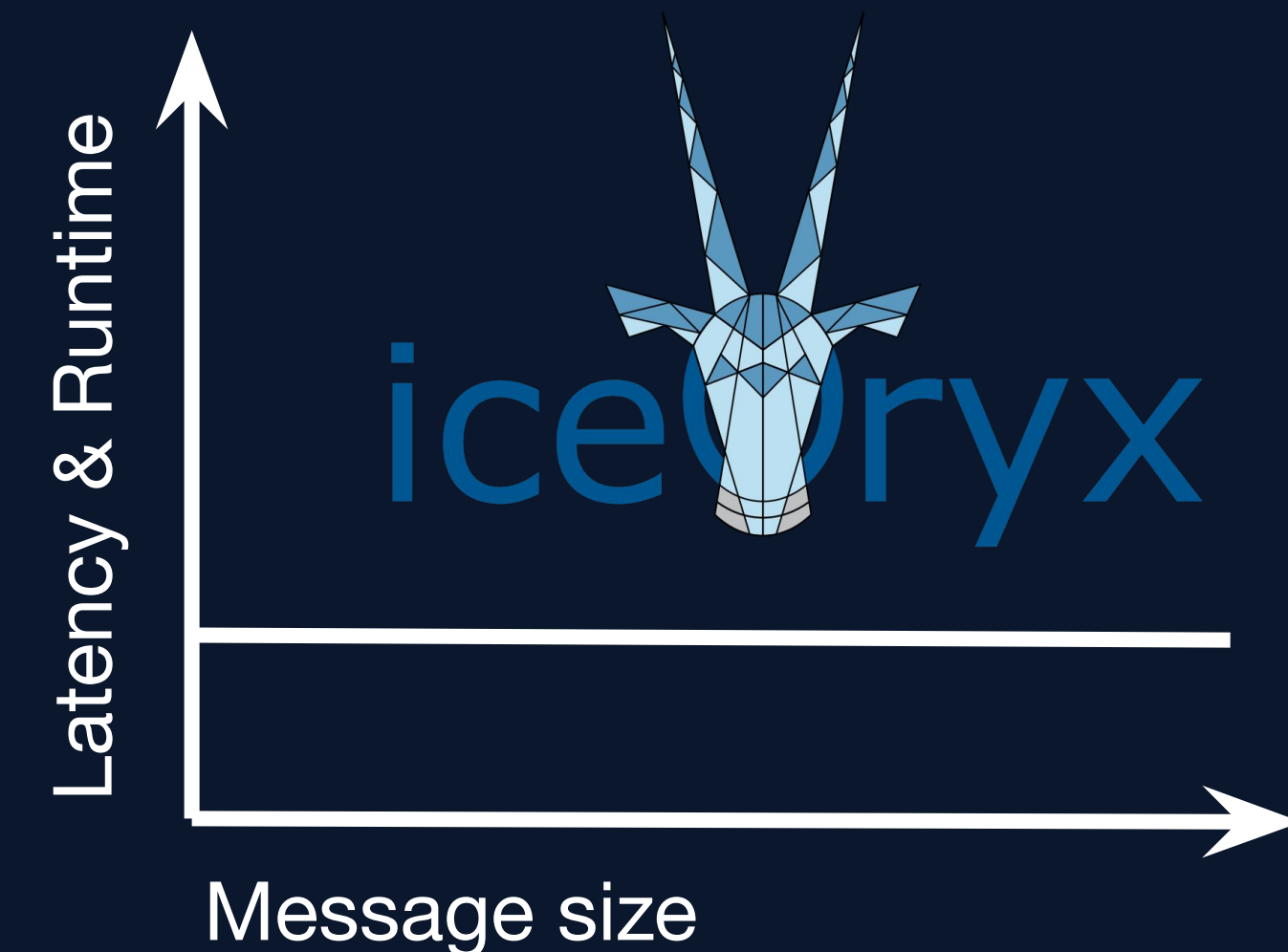
# Safety and Open Source, Oh My?

## Introduction

- Apex.OS Cert
  - Fork of Robotic Operating System (ROS2)
  - Certified according to ISO 26262
  - The “Red Hat Enterprise Linux” (RHEL) for ROS2
- Eclipse iceoryx
  - Zero-copy inter-process communication middleware for safety-critical applications
  - It is being integrated into Apex.OS or can be used standalone
  - For more info, watch last year’s [talk](#)



<https://www.apex.ai/apex-os>



<https://github.com/eclipse-iceoryx/iceoryx>

# Safety and Open Source, Oh My?

## Why Processes Are Your Friends

```
void UndefinedBehavior() {  
    bool myBool;  
  
    if ( myBool )    std::cout << "true" << std::endl;  
    if ( !myBool )  std::cout << "false" << std::endl;  
}
```

Four possible outputs:

1)

2)

3)

4)

# Safety and Open Source, Oh My?

## Why Processes Are Your Friends

```
void UndefinedBehavior() {  
    bool myBool;  
  
    if ( myBool )    std::cout << "true" << std::endl;  
    if ( !myBool )  std::cout << "false" << std::endl;  
}
```

Four possible outputs:

1) true

2) false

3) true  
false

4)



# Safety and Open Source, Oh My?

## Why Processes Are Your Friends

```
void UndefinedBehavior() {  
    bool myBool;  
  
    if ( myBool )    std::cout << "true" << std::endl;  
    if ( !myBool )  std::cout << "false" << std::endl;  
}
```

Four possible outputs:

1) true

2) false

3) true  
false

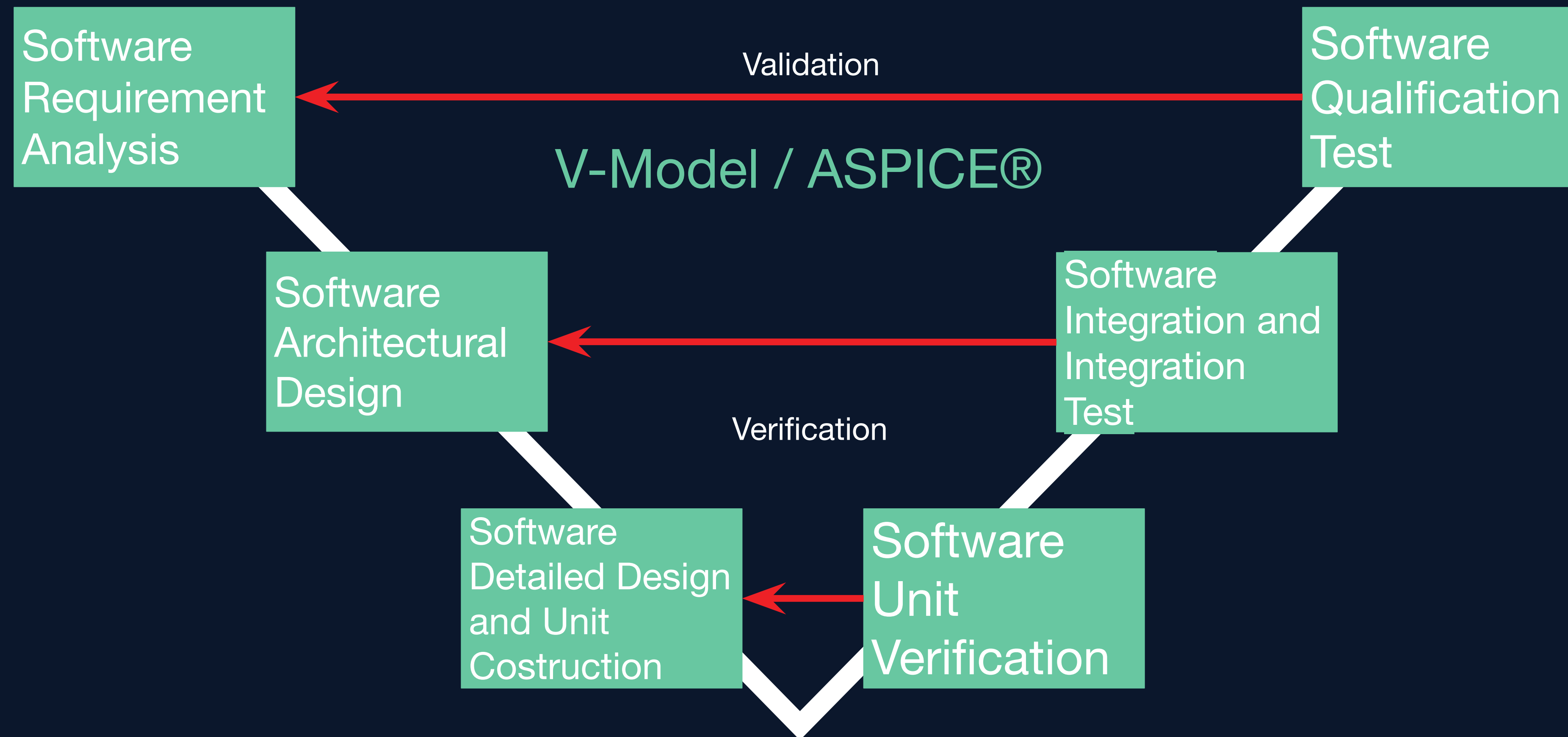
4)

Undefined behavior means: anything can happen!

# Safety and Open Source, Oh My?

## Typical Automotive Software Development Process

**Validation:** Am I'm doing the right thing?  
**Verification:** Am I'm doing the thing right?



Automotive Spice = Automotive Software Process Improvement and Capability dEtermination

# Safety and Open Source, Oh My?

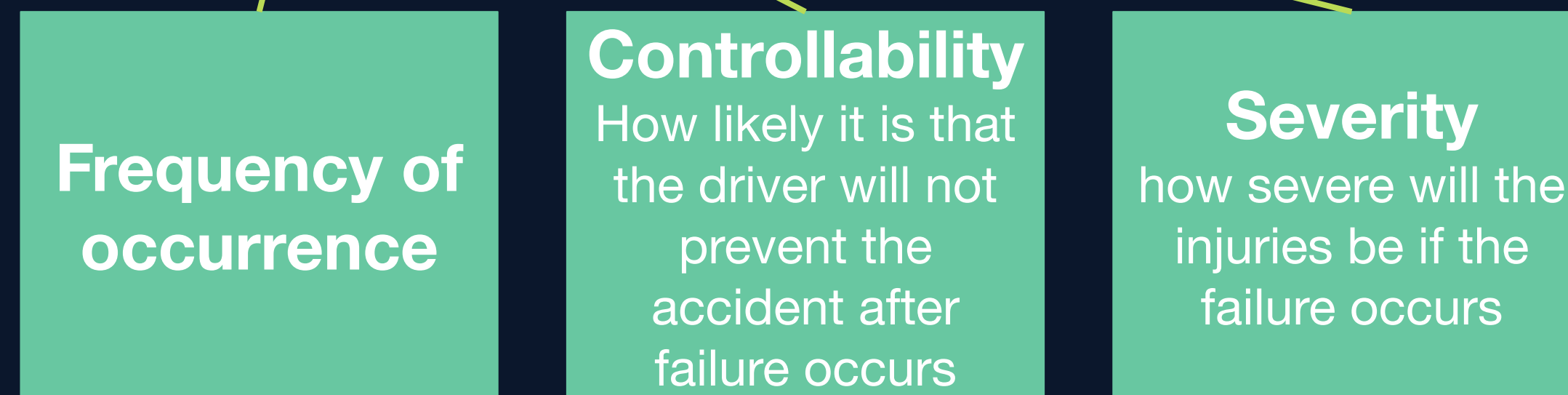
## ISO 26262 Functional-Safety Standard for Road Vehicles

- What does it contain?
  - Processes to be followed to ensure safety
  - Formal definition for errors that might happen during the nominal operation of a car
  - Formal definition of risk with relation to the possible errors and ways of risk assessment and mitigation
  - Enforces an independent safety assessment
- What is ASIL?
  - Automotive safety integrity level (ASIL)
    - QM (quality management), e.g. infotainment
    - A, e.g. rain sensor
    - B, e.g. lane keep assist
    - C, e.g. suspension
    - D, e.g. automated driving

# Safety and Open Source, Oh My?

## ISO 26262 Functional-Safety Standard for Road Vehicles

- Safety  $\equiv$  Absence of unreasonable risk
- Risk = F(f, C, S)



- Example
  - Breaking system
    - Exposure: high probability
    - Controllability: difficult to control
    - Severity: fatal injuries are likely

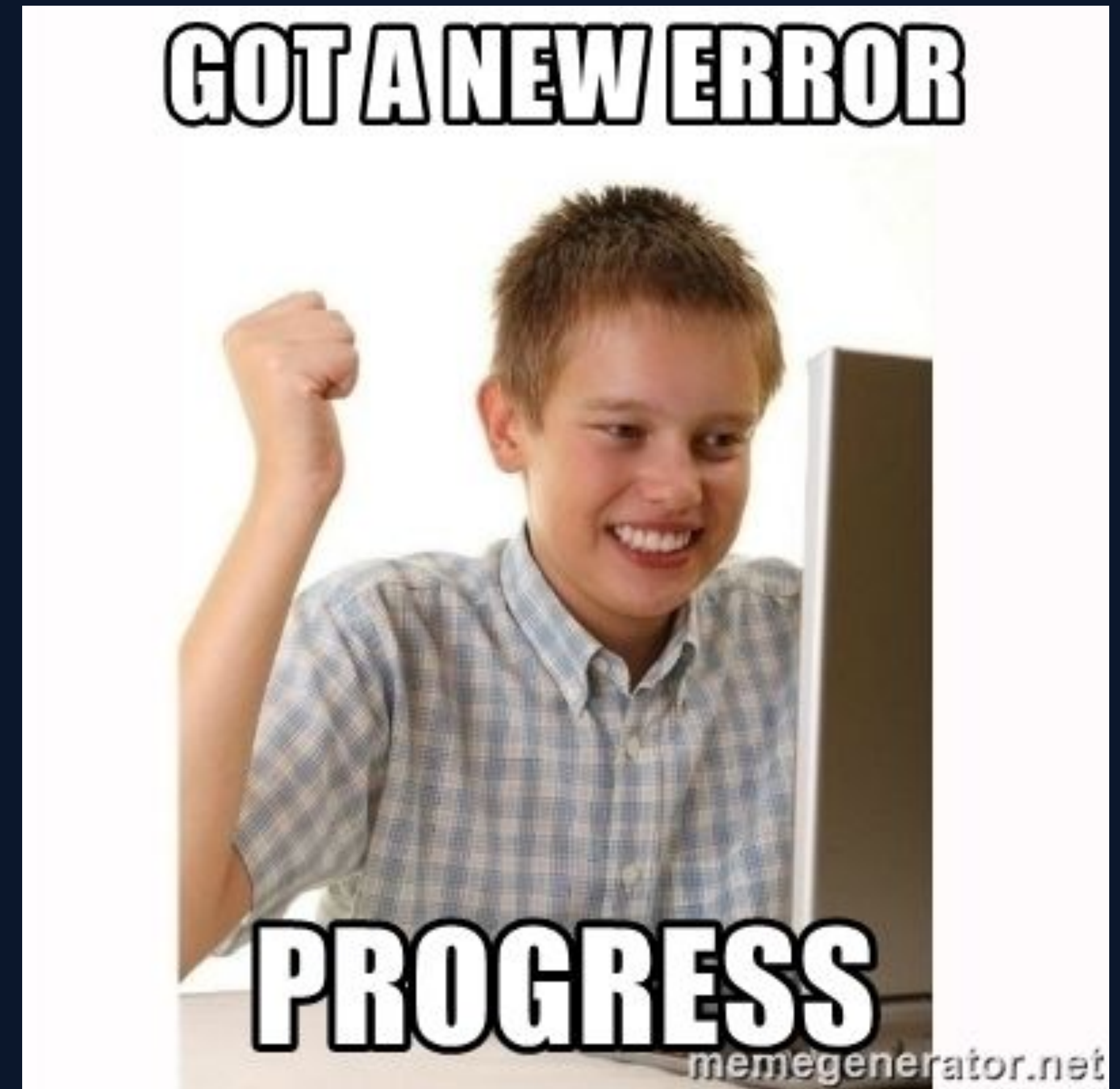
- *Frequency* has two parts
  - Exposure
    - How often is the car in a situation where a safety hazard can occur
  - Failure rate
    - Probability of system to fail
    - Not considered in risk assessment, ASIL instead



# Safety and Open Source, Oh My?

## Goals

- Make developers happy
- Transparency for the community
- Be helpful to newbies
- Encourage knowledge sharing and make life easy for external contributors
- Work as much as possible in the open
  - Pull request reviews
  - Discussions and planning in GitHub issues
  - Gitter.im chat
- Shape workflow after established guidelines (e.g. Bosch or Apex.AI)



# Safety and Open Source, Oh My?

Many Tools, So Safety

Design in e.g. Markdown  
and PlantUML



Code



Unit tests

# Safety and Open Source, Oh My?

Many Tools, So Safety



Requirements in  
commercial tool



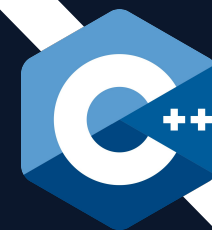
Design in e.g. Markdown  
and PlantUML



Safety manual



Code



Unit tests, written with GTest,  
run in VectorCast



Static code analysis with  
Perforce Helix QAC



Work-in-progress



LTT-ng tracing

**FRIDA**

Integration tests

Performance tests

More to consider, good read up is [ROS quality levels](#)

# Safety and Open Source, Oh My?

## Review of Tools & Processes - Not Enough To Be Safe!



- Relying on safety processes is not enough.
  - Not every programmer is aware of all the rules.
  - Static code analyser do not catch everything.
  - Corner cases can always slip through test cases
- Creating programming paradigms like `and_then/or_else` further reduces errors
- Implementing STL constructs (like `vector`, `list`) is needed to avoid
  - usage of heap
  - exceptions
  - undefined behavior
  - enforce boundary checks
- Extended test strategies which go beyond functional safety standards



# Safety and Open Source, Oh My?

## Review of Tools & Processes - Code Example



```
void Receiver() {  
  
    std::optional<vec3> currentPosition = ReceivePosition();  
    //...  
  
    float travelDistance =  
        distance(startPosition, *currentPosition);  
    //...  
}
```

# Safety and Open Source, Oh My?

## Review of Tools & Processes - Code Example



```
float travelDistance = distance(startPosition, *currentPosition);
```

What if no position was received?

- `currentPosition` does not contain any value.
- Access a valid memory position with arbitrary content.

# Safety and Open Source, Oh My?

## Review of Tools & Processes - Code Example



```
float travelDistance = distance(startPosition, *currentPosition);
```

What if no position was received?

- `currentPosition` does not contain any value.
- Access a valid memory position with arbitrary content.

How can we avoid invalid access?

# Safety and Open Source, Oh My?

## Review of Tools & Processes - a Functional Approach



Our C++14 `std::optional` implementation offers two additional methods:

`and_then`      calls a given lambda when containing a value

`or_else`      if no value available the lambda provided in here is  
called



# Safety and Open Source, Oh My?



## Review of Tools & Processes - a Functional Approach

```
ReceivePosition()  
    .and_then([&](auto &position) {  
        float travelDistance = distance(startPosition, position);  
        // ...  
    })  
    .or_else([]() {  
        std::cout << "no position update received";  
        // ...  
    });
```

# Safety and Open Source, Oh My?



## Review of Tools & Processes - a Functional Approach

```
ReceivePosition()  
    .and_then([&](auto &position) {  
        float travelDistance = distance(startPosition, position);  
        // ...  
    })  
    .or_else([]() {  
        std::cout << "no position update received";  
        // ...  
    });
```

# Safety and Open Source, Oh My?

## Review of Tools & Processes - a Functional Approach



```
ReceivePosition()  
  .and_then([&](auto &position) {  
    float travelDistance =  
      distance(startPosition, position);  
    // ...  
  })  
  .or_else([]() {  
    std::cout << "no position update received";  
    // ...  
  });
```

```
std::optional<vec3> currentPosition = ReceivePosition();  
  
if ( currentPosition ) {  
    float travelDistance =  
      distance(startPosition, *currentPosition);  
    // ...  
} else {  
    std::cout << "no position update received";  
    // ...  
}
```

# Safety and Open Source, Oh My?



## Community Contributions - the Workflow

- Eclipse iceoryx is hosted by the Eclipse Foundation
  - Defines some rules and workflows in a [handbook](#)
- Currently eight Eclipse Committers
  - Apex.AI Inc
  - Robert Bosch GmbH
- Eclipse Committers
  - Can approve pull requests
  - Can merge after two approvals
  - Have the responsibility to fulfil process

# Apex.AI

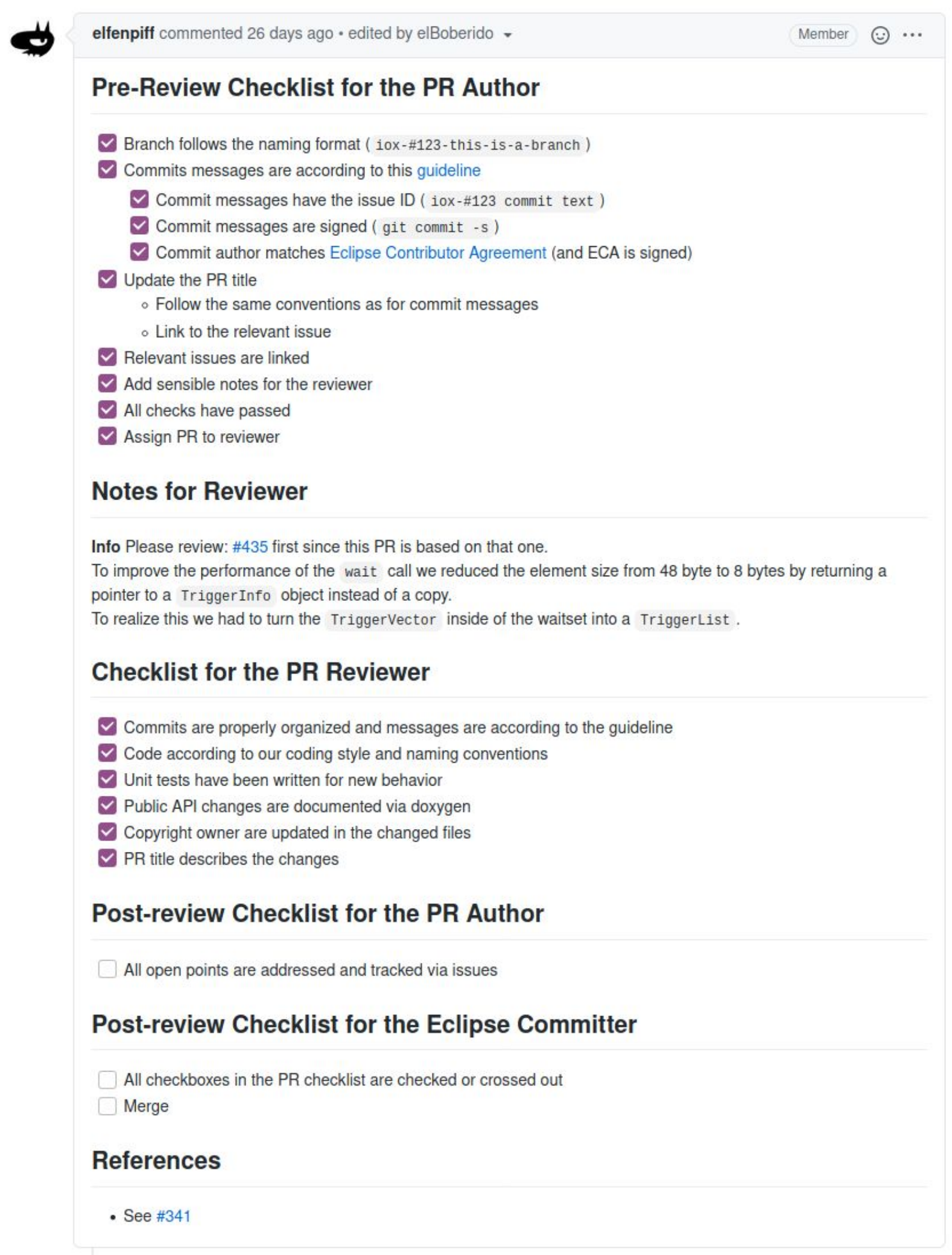




# Safety and Open Source, Oh My?

## Community Contributions - the Workflow

- Static code analysis with Perforce QAC
  - Adaptive Autosar C++ 14
  - MISRA C++ 2008, replaced by MISRA C++ 2020
  - SEI CERT C++ 2016
- Continuous integration via GitHub Actions
  - MacOS, Windows, Ubuntu build (QNX soon)
  - Clang Sanitizer (ASan, UBSan, LSan)
  - -Werror
- Review protocols
  - Stored in GitHub as pull requests
  - Template for checklist automatically added



elfenpiff commented 26 days ago • edited by elBoberido

Member

### Pre-Review Checklist for the PR Author

- ☒ Branch follows the naming format ( `iox-#123-this-is-a-branch` )
- ☒ Commits messages are according to this [guideline](#)
  - ☒ Commit messages have the issue ID ( `iox-#123 commit text` )
  - ☒ Commit messages are signed ( `git commit -s` )
  - ☒ Commit author matches [Eclipse Contributor Agreement](#) (and ECA is signed)
- ☒ Update the PR title
  - Follow the same conventions as for commit messages
  - Link to the relevant issue
- ☒ Relevant issues are linked
- ☒ Add sensible notes for the reviewer
- ☒ All checks have passed
- ☒ Assign PR to reviewer

### Notes for Reviewer

**Info** Please review: [#435](#) first since this PR is based on that one.  
To improve the performance of the `wait` call we reduced the element size from 48 byte to 8 bytes by returning a pointer to a `TriggerInfo` object instead of a copy.  
To realize this we had to turn the `TriggerVector` inside of the waitset into a `TriggerList`.

### Checklist for the PR Reviewer

- ☒ Commits are properly organized and messages are according to the guideline
- ☒ Code according to our coding style and naming conventions
- ☒ Unit tests have been written for new behavior
- ☒ Public API changes are documented via doxygen
- ☒ Copyright owner are updated in the changed files
- ☒ PR title describes the changes

### Post-review Checklist for the PR Author

- ☐ All open points are addressed and tracked via issues

### Post-review Checklist for the Eclipse Committer

- ☐ All checkboxes in the PR checklist are checked or crossed out
- ☐ Merge

### References

- See [#341](#)

# Safety and Open Source, Oh My?

## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```



# Safety and Open Source, Oh My?

## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

100% line coverage

`Calculate(2.0, 2.0);` // =>  $2 + 2 / -1$  =>  $-4$



# Safety and Open Source, Oh My?



## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

### 100% line coverage

```
Calculate(2.0, 2.0); // => 2 + 2 / -1 => -4
```

### Full branch coverage

```
Calculate(-2.0, 2.0); // => -2 + 2 / -1 => 0
```

# Safety and Open Source, Oh My?



## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

### 100% line coverage

`Calculate(2.0, 2.0);` //  $\Rightarrow 2 + 2 / -1 \Rightarrow -4$

### Full branch coverage

`Calculate(-2.0, 2.0);` //  $\Rightarrow -2 + 2 / -1 \Rightarrow 0$

### MC/DC Coverage

Every condition must be executed twice,  
once for true and once for false.

MC/DC = Modified condition / decision coverage



# Safety and Open Source, Oh My?



## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

### 100% line coverage

```
Calculate(2.0, 2.0); // => 2 + 2 / -1 => -4
```

### Full branch coverage

```
Calculate(-2.0, 2.0); // => -2 + 2 / -1 => 0
```

### MC/DC Coverage

Every condition must be executed twice,  
once for true and once for false.

```
Calculate(-2.0, -2.0); // => -2 + -2 / -1 => -4
```

```
Calculate( 2.0, -2.0); // => 2 + -2 / -1 => 0
```

MC/DC = Modified condition / decision coverage

# Safety and Open Source, Oh My?

## Testing - Code Example



```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

There is still a bug.

```
Calculate(1.0, 2.0); // => 1 + 2 / 0  => -???
```

# Safety and Open Source, Oh My?



## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

There is still a bug.

```
Calculate(1.0, 2.0); // => 1 + 2 / 0  => -???
```

We should test also for the following ideas:

- Zero
- One
- Many
- Corner Cases
- Limits

# Safety and Open Source, Oh My?



## Testing - Code Example

```
float Calculate(float a, float b)
{
    float result = 1.0f;

    if ( a > 0.0f && b > 0.0f )
        result = result - a;

    return a + b / result;
}
```

But there is still a bug and undefined behavior.

```
Calculate(
    340282346638528859811704183484516925440.0000000,
    2.0);
```

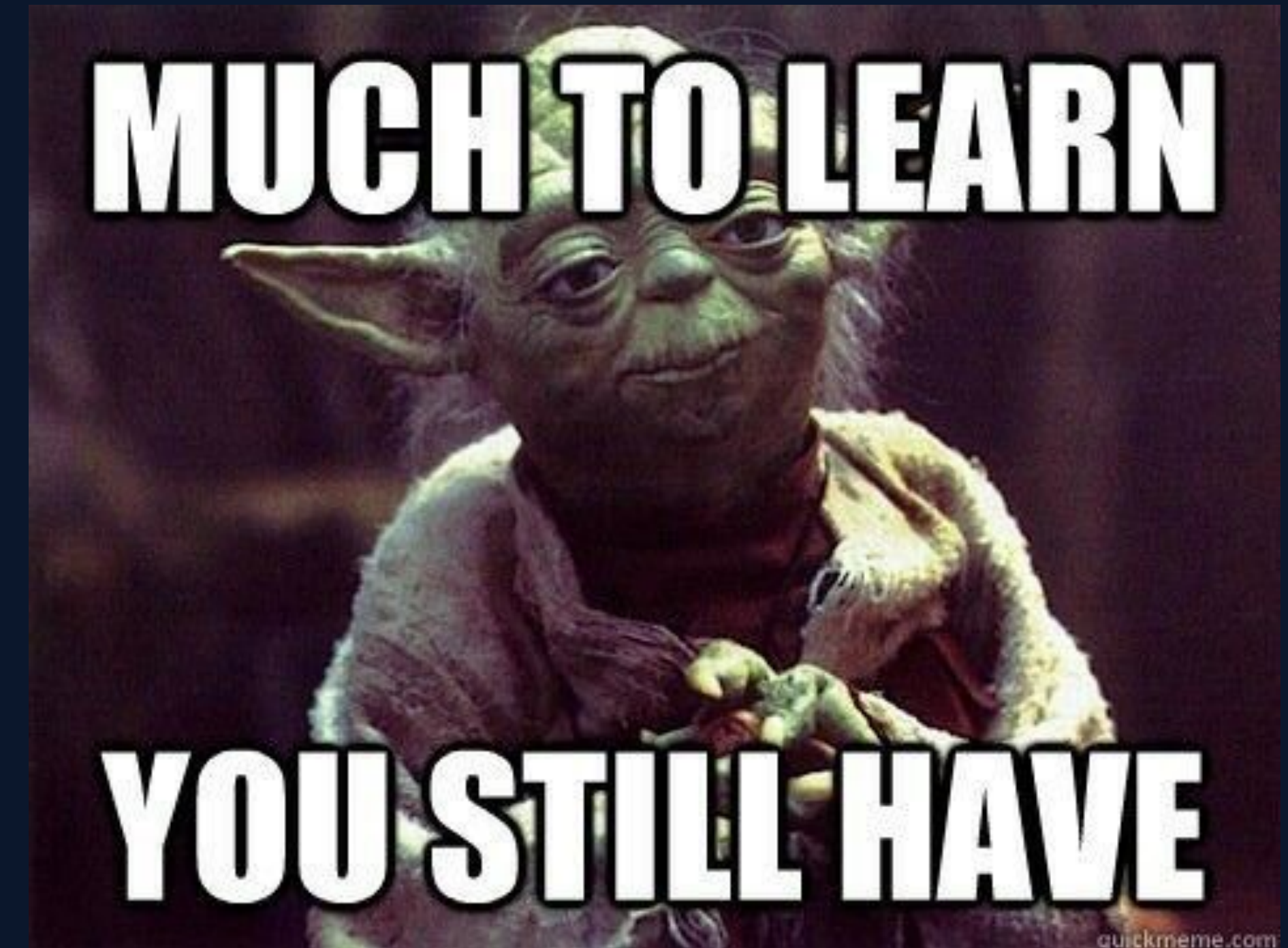
Floating point overflow leads to undefined behavior.

```
return a + b / result;
```

# Safety and Open Source, Oh My?

## Lessons Learned

- FOSS for safety-critical automotive vehicle computers has big potential
  - Possibility to reduce costs
  - Creating better software quality for customers and users through transparency
- Developing safety software in the open is no rocket science, just do it
  - Vendors of special tools are often supportive to advertise their tool
- Certified != safe
  - Best practises used in the industry
  - Necessary but not sufficient





# Safety and Open Source, Oh My?

## Outlook for Eclipse iceoryx

- Release v1.0 planned for early Q2 2021
  - n:m communication
  - New functional C++ API
  - C API
  - MacOS support
- Integration into Cyclone DDS (ROS2 default middleware)
- ISO 26262 certified release planned for early 2022

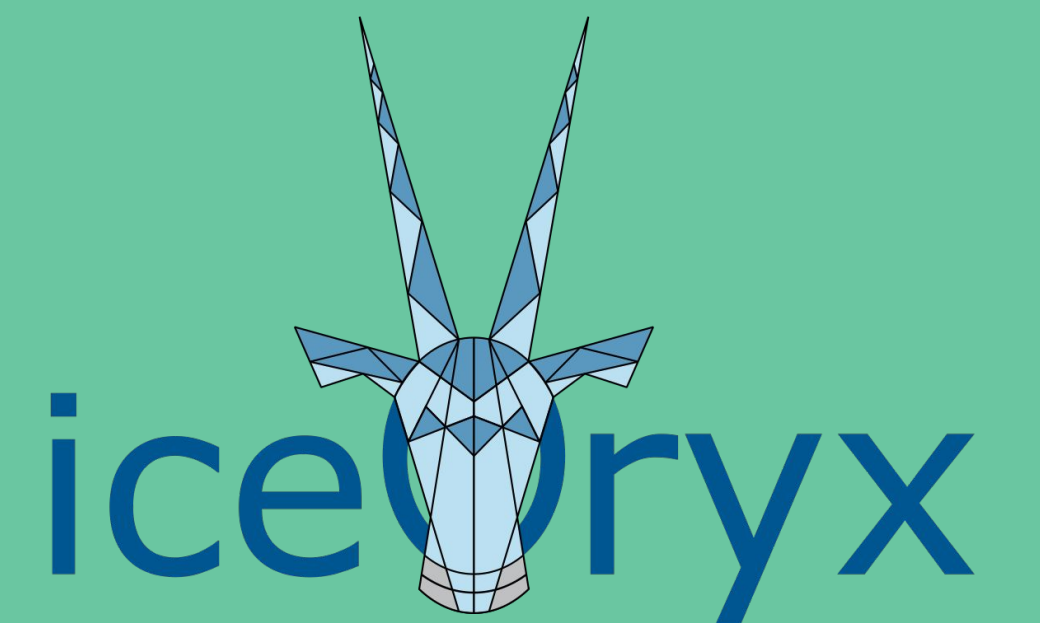


Write us:

[iceoryx-dev@eclipse.org](mailto:iceoryx-dev@eclipse.org)

Questions?

Apex.AI



# Safety and Open Source, Oh My?

## List of References

- <https://www.pinterest.com/pin/183310647311583832/>
- Automotive SPICE® Pocket Guide, Method Park AG
- <https://beza1e1.tuxen.de/aspice.html>
- <https://memegenerator.net/instance/31804326/computer-kid-got-a-new-error-progress>
- <https://i.morioh.com/2019/11/06/362222e42d54.jpg>
- <https://github.com/isocpp/logos>
- <https://refactorsaurusrex.com/posts/unit-testing-hostingenvironment.mapproach/>
- <https://i.pinimg.com/originals/8d/ec/f9/8decf9caed777b8d0d698e01270ce308.png>
- <http://www.quickmeme.com/img/e9/e9b82533f50538f4d36656f24bf2afb39642223033cd19d52ef1eea5b03ab1bf.jpg>



# Safety and Open Source, Oh My?



## Backup: Traceability

- Traceability of code
  - Naming convention for branches and commits
  - Trace every line of code to a GitHub issue and requirement

```
* commit f1c4b31c54cc7fccb352fc3cdda562d9a22fe638 (HEAD -> master, origin/master, origin/HEAD)
Merge: 6e31bc4f 424bbb82
Author: dkroenke <38155883+dkroenke@users.noreply.github.com>
Date: Wed Jan 13 15:45:18 2021 +0100

    Merge pull request #463 from chiranjeemaddi/iox-#454-Enhancing-Test-in-RouDi

    iox-#454 Added testcases for RoudiProcess class

* commit 424bbb8270ab009ee3a42a6888483e5dc90a4705
Merge: 87c71548 61f59f2a
Author: Chiranjevi Maddi (RBEI/EBB1) <chiranjevi.maddi@in.bosch.com>
Date: Mon Jan 11 10:51:30 2021 +0530

    Merge remote-tracking branch 'upstream/master' into iox-#454-Enhancing-Test-in-RouDi

* commit 87c7154857f06a411633a2af2d198fed5a033f7d
Author: Chiranjevi Maddi (RBEI/EBB1) <chiranjevi.maddi@in.bosch.com>
Date: Mon Jan 11 10:51:12 2021 +0530

    iox-#454 Replaced test result number with variables

    Signed-off-by: Chiranjevi Maddi (RBEI/EBB1) <chiranjevi.maddi@in.bosch.com>

* commit c05d6e05bad31bf201bbae50cf240efcf7c98d07
Merge: 4437cf01 64fa3893
Author: Chiranjevi Maddi (RBEI/EBB1) <chiranjevi.maddi@in.bosch.com>
Date: Fri Jan 8 17:55:48 2021 +0530

    Merge remote-tracking branch 'upstream/master' into iox-#454-Enhancing-Test-in-RouDi

* commit 4437cf010bdfc6bf22bb0355079495e12de7d06b
Author: Chiranjevi Maddi (RBEI/EBB1) <chiranjevi.maddi@in.bosch.com>
Date: Fri Jan 8 17:55:23 2021 +0530

    iox-#454 updated the sendMQ fail test function

    Signed-off-by: Chiranjevi Maddi (RBEI/EBB1) <chiranjevi.maddi@in.bosch.com>
```

# Safety and Open Source, Oh My?

## Backup: Why Eclipse?

- Eclipse Foundation provides framework and infrastructure to maintain FOSS projects
- Benefit from open source knowhow
- IP questions taken care by Eclipse Foundation
- Responsible disclosure possibility provided



# Safety and Open Source, Oh My?

## Backup: Community Contributions - Code Example



```
void Sender() {  
    if ( NewCameraImageAvailable() ) {  
        Image *image = new Image();  
        AcquireCameraImage(image);  
        SendImage(image);  
    }  
  
    vec3 *position = new vec3();  
    AcquireCarPosition(position);  
    SendPosition(position);  
  
    positionHistory.push_back(position);  
}
```

# Safety and Open Source, Oh My?

## Backup: Community Contributions - Code Example

```
void Sender() {  
    if ( NewCameraImageAvailable() ) {  
        Image *image = new Image();  
        AcquireCameraImage(image);  
        SendImage(image);  
    }  
  
    vec3 *position = new vec3();  
    AcquireCarPosition(position);  
    SendPosition(position);  
}
```



# Safety and Open Source, Oh My?

## Backup: Community Contributions - Bug in Memory Allocation



Sometimes your application crashes while allocating memory for the camera image.

```
Image *image = new Image();
```

# Safety and Open Source, Oh My?

## Community Contributions - Bug in Memory Allocation



Sometimes your application crashes while allocating memory for the camera image.

```
Image *image = new Image();
```

But a quick inspection of your system reveals that enough memory is available.

```
> cat /proc/meminfo
MemTotal:        65841192 kB
MemFree:         54992572 kB
MemAvailable:    61002968 kB
```

## What happened?

# Safety and Open Source, Oh My?

## Backup: Community Contributions - Memory Fragmentation



Let's start with 16 cells of fresh and clean memory.





# Safety and Open Source, Oh My?



## Backup: Community Contributions - Memory Fragmentation

Let's start with 16 cells of fresh and clean memory.



We send the car **position** and reserve a memory cell.



# Safety and Open Source, Oh My?



## Backup: Community Contributions - Memory Fragmentation

Let's start with 16 cells of fresh and clean memory.



We send the car **position** and reserve a memory cell.



Reserving 6 cells of memory for the **image**



# Safety and Open Source, Oh My?



## Backup: Community Contributions - Memory Fragmentation

Let's start with 16 cells of fresh and clean memory.



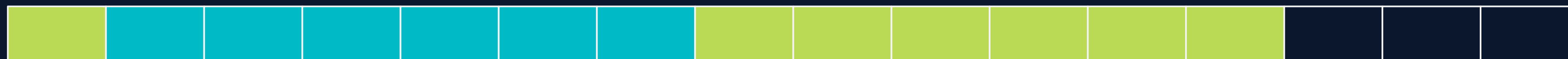
We send the car **position** and reserve a memory cell.



Reserving 6 cells of memory for the **image**



We update the car **position** more often and require additional cells.

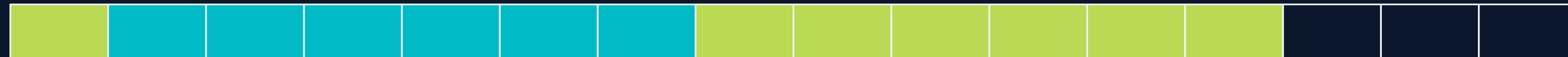


# Safety and Open Source, Oh My?



## Backup: Community Contributions - Memory Fragmentation

We update the car **position** more often and require additional cells.



Some of the **position** informations are not required by the user - let's release them.

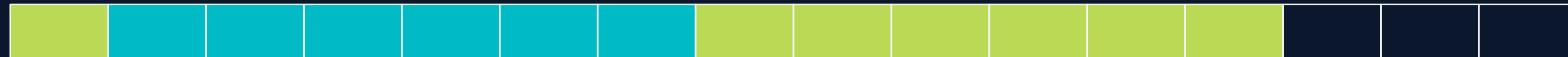


# Safety and Open Source, Oh My?



## Backup: Community Contributions - Memory Fragmentation

We update the car **position** more often and require additional cells.



Some of the **position** informations are not required by the user - let's release them.



State:

- 8 free memory cells
- no space left for an **image** since we require 6 consecutive cells

# Safety and Open Source, Oh My?

## Backup: Community Contributions - Memory Management



In safety critical systems you have to guarantee that you never go out of memory!

Our solutions are:

- To never use heap allocations.
- A bucket memory pool which guarantees that enough memory is available.



# Safety and Open Source, Oh My?

## Backup: Community Contributions - Memory Management



In safety critical systems you have to guarantee that you never go out of memory!

Our solutions are:

- To never use heap allocations.
- A bucket memory pool which guarantees that enough memory is available.

How can we make sure no one uses the heap?