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

© 2021 Apex.AI, Inc.
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)
Why Processes Are Your Friends

```cpp
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  4) false
Safety and Open Source, Oh My?

Why Processes Are Your Friends

```cpp
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?

V-Model / ASPICE®

Software Requirement Analysis
Software Architectural Design
Software Detailed Design and Unit Construction
Software Unit Verification
Software Integration and Integration Test
Software Qualification Test

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 ≡ Absence of unreasonable risk
- Risk = F(f, C, S)

**Frequency of occurrence**

**Controllability**
How likely it is that the driver will not prevent the accident after failure occurs

**Severity**
How severe will the injuries be if the failure occurs

- 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

© 2021 Apex.AI, Inc.
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

LTT-ng tracing

FRIDA

Integration tests

Performance tests

Unit tests, written with GTest, run in VectorCast

Static code analysis with Perforce Helix QAC

More to consider, good read up is ROS quality levels

© 2021 Apex.AI, Inc.
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
void Receiver() {

    std::optional<vec3> currentPosition = ReceivePosition();
    //...

    float travelDistance =
        distance(startPosition, *currentPosition);
    //...
}

Safety and Open Source, Oh My?

Review of Tools & Processes - Code Example
**Safety and Open Source, Oh My?**

**Review of Tools & Processes - Code Example**

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

  .and_then([&](auto &position) {
    float travelDistance = distance(startPosition, position);
    // ...
  })

  .or_else([]() {
    std::cout << "no position update received";
    // ...
  });
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
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
float Calculate(float a, float b)
{
    float result = 1.0f;

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

    return a + b / result;
}
float Calculate(float a, float b) {
    float result = 1.0f;

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

    return a + b / result;
}
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
Safety and Open Source, Oh My?

Testing - Code Example

```c
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.

MC/DC = Modified condition / decision coverage
Safety and Open Source, Oh My?

Testing - Code Example

```c
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
Testing - Code Example

```c
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 => -???
```
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

```c
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.

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

Floating point overflow leads to undefined behavior.

```c
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 practices used in the industry
  - Necessary but not sufficient
Safet y 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
Questions?
Safety and Open Source, Oh My?

List of References

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

```plaintext
commit f1c4b3c154cc77fcco352fc3c0dd5e629a22f6e638 (HEAD -> master, origin/master, origin/HEAD)
Author: dkroenke <391350883@dkroenke@users.noreply.github.com>
Date: Wed Jan 13 15:45:10 2021 +0100
Merge pull request #451 from chiranjevliddl/iox-454-Enhancing-Test-in-Roudl
- iox-454 Added testcases for RoudlProcess class
• commit 424bb2b2b7a0b0e342a088843e3d9e0a4705
  Merge: 87c7154ab61f55972a
  Author: Chiranjeel Maddr (RBEI/EBBI) <chiranjeel.maddl@in.bosch.com>
  Date: Mon Jan 11 10:51:30 2021 +0530
  Merge remote-tracking branch 'upstream/master' into iox-454-Enhancing-Test-in-Roudl
• commit 87c7154ab61f55972a
  Merge: 87c7154ab61f55972a
  Author: Chiranjeel Maddr (RBEI/EBBI) <chiranjeel.maddl@in.bosch.com>
  Date: Mon Jan 11 10:51:12 2021 +0530
  iox-454 Replaced test result number with variables
  Signed-off-by: Chiranjeel Maddr (RBEI/EBBI) <chiranjeel.maddl@in.bosch.com>
• commit 87c7154ab61f55972a
  Merge: 87c7154ab61f55972a
  Author: Chiranjeel Maddr (RBEI/EBBI) <chiranjeel.maddl@in.bosch.com>
  Date: Fri Jan 8 17:55:48 2021 +0530
  Merge remote-tracking branch 'upstream/master' into iox-454-Enhancing-Test-in-Roudl
• commit 87c7154ab61f55972a
  Merge: 87c7154ab61f55972a
  Author: Chiranjeel Maddr (RBEI/EBBI) <chiranjeel.maddl@in.bosch.com>
  Date: Fri Jan 8 17:55:23 2021 +0530
  iox-454 updated the sendMQ fail test function
  Signed-off-by: Chiranjeel Maddr (RBEI/EBBI) <chiranjeel.maddl@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
void Sender() {
    if (NewCameraImageAvailable()) {
        Image *image = new Image();
        AcquireCameraImage(image);
        SendImage(image);
    }
}

vec3 *position = new vec3();
AcquireCarPosition(position);
SendPosition(position);

positionHistory.push_back(position);
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.

```c
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.

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

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

```bash
$ 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.
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?