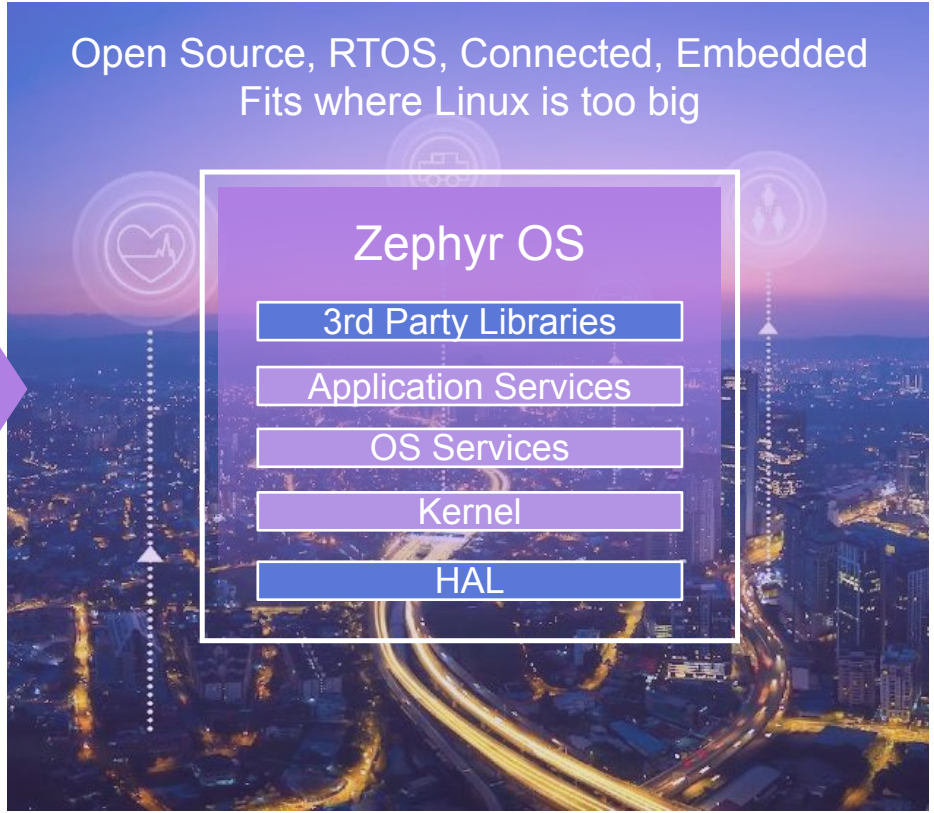
THE LINUX FOUNDATION
MEMBER SUMMIT

Zephyr:
Journey to Safety Certification

Kate Stewart, Zephyr Project Director, The Linux Foundation
Nicole Pappler, Zephyr Functional Safety Manager, AlektoMetis

Zephyr Project

- **Open source** real time operating system
- **Developer friendly** with vibrant community participation
- Built with **safety and security** in mind
- **Broad SoC, board and sensor support**.
- **Vendor Neutral** governance
- **Permissively licensed** - Apache 2.0
- **Complete**, fully integrated, highly configurable, **modular** for **flexibility**
- **Product** development ready using LTS includes **security updates**



Products based on Zephyr

Oticon More Hearing Aid

Lildog & Lilcat Pet Tracker

Livestock Tracker

Moto Watch 100

Samsung Galaxy Ring

Proglove

Adhoc Smart Waste

Google Chromebook

Framework laptop

Keeb.io BDN9

Hati-ACE

Safety Pod

BLiXT solid state circuit breaker

Aethero Deimos Satellite

PHYTEC Distancer

Laird Connectivity sensors & gateways

BeST pump monitoring

Vestas Wind Turbines

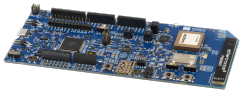
zephyrproject.org/products-running-zephyr

700+ supported boards... and growing

Arduino Portenta H7

ESP32

Sipeed HiFive1



nRF9160 DK

STM32F746G Disco

M5StickC PLUS

TDK RoboKit 1

BBC micro:bit v2

Blue Wireless Swan

Arduino Nano 33 BLE

Intel UP Squared

Dragino LSN50 LoRA Sensor Node

Microchip SAM E54 Xplained Pro Evaluation Kit

Raspberry Pi Pico

Altera MAX10

NXP i.MX8MP EVK

Adafruit Feather M0 LoRa

u-blox EVK-NINA-B3

docs.zephyrproject.org/latest/boards

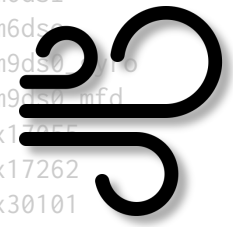
220+ Sensors Already Integrated

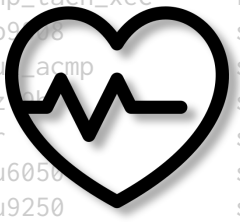
adt7420
adxl345
adxl362
adxl372
ak8975
amg88xx
ams_as5600
ams_iAQcore
apds9960
bma280
bmc150_magn
bme280
bme680
bmg160
bmi160
bmi270
bmm150
bmp388
bq274xx
ccs811

dht
dps310
ds1[illegible]
ens[illegible]
esp[illegible]
fdc[illegible]
fx[illegible]
fx[illegible]
grove
grow_r502a
hmc5883l
hp206c
ht[illegible]221
i[illegible]250d
i[illegible]2605
i[illegible]2670
i[illegible]
icp[illegible]125
iis2dh
iis2dlpc

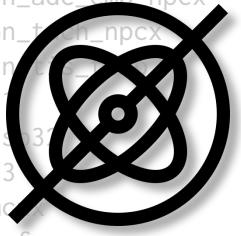
iis2iclx
iis2mdc
iis3dhhc
ina219
ina23[illegible]
isl2[illegible]35
ism3[illegible]
ite_ta[illegible]x2
ite_vcmp_it8xxx2
lis2dh
lis2ds12
lis2dw12
lis2[illegible]
lis3[illegible]
lm75[illegible]
lm77[illegible]
lps22[illegible]
lps22hh
lps25hb
lsm303dlhc_magn

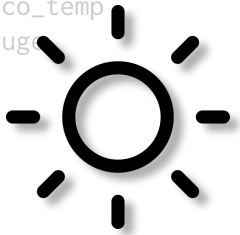
lsm6ds0
lsm6dsl
lsm6ds[illegible]
lsm9ds0[illegible]
lsm9[illegible]mfd
max1[illegible]
max17262
max30101
max31875
max44009
max6675
mchp_tach_xec
mcp9[illegible]
mcu_[illegible]acmp
mhz[illegible]
mpr
mpu6050
mpu9250
ms5607
ms5837





nrf5
nuvoton_adc_cmp_npcx
nuvoton_t[illegible]n_npcx
nxp_kin[illegible]
opt300[illegible]
pcnt_es[illegible]
pms7003
qdec_mc[illegible]
qdec_nrfx
qdec_sam
qdec_stm32
rpi_pico_temp
sbs_gaug[illegible]
sgp40
sht3xd
sht4x
shtcx
si7006
si7055
si7060





si7210
sm351lt
stm32_temp
stm32_vbat
stmemsc
stts751
sx9500
th02
ti_hdc
ti_hdc20xx
tmp007
tmp108
tmp112
tmp116
vcnl4040
vl53l0x
wsen_hids
wsen_itds

github.com/zephyrproject-rtos/zephyr/tree/main/drivers/sensor

Supported Hardware Architectures

Zephyr®

ARC®

arm

Cortex-M, Cortex-R
& Cortex-A

intel®

x86 & x86_64

MIPS

Nios® II
Processor

RISC-V®

32 & 64 bit

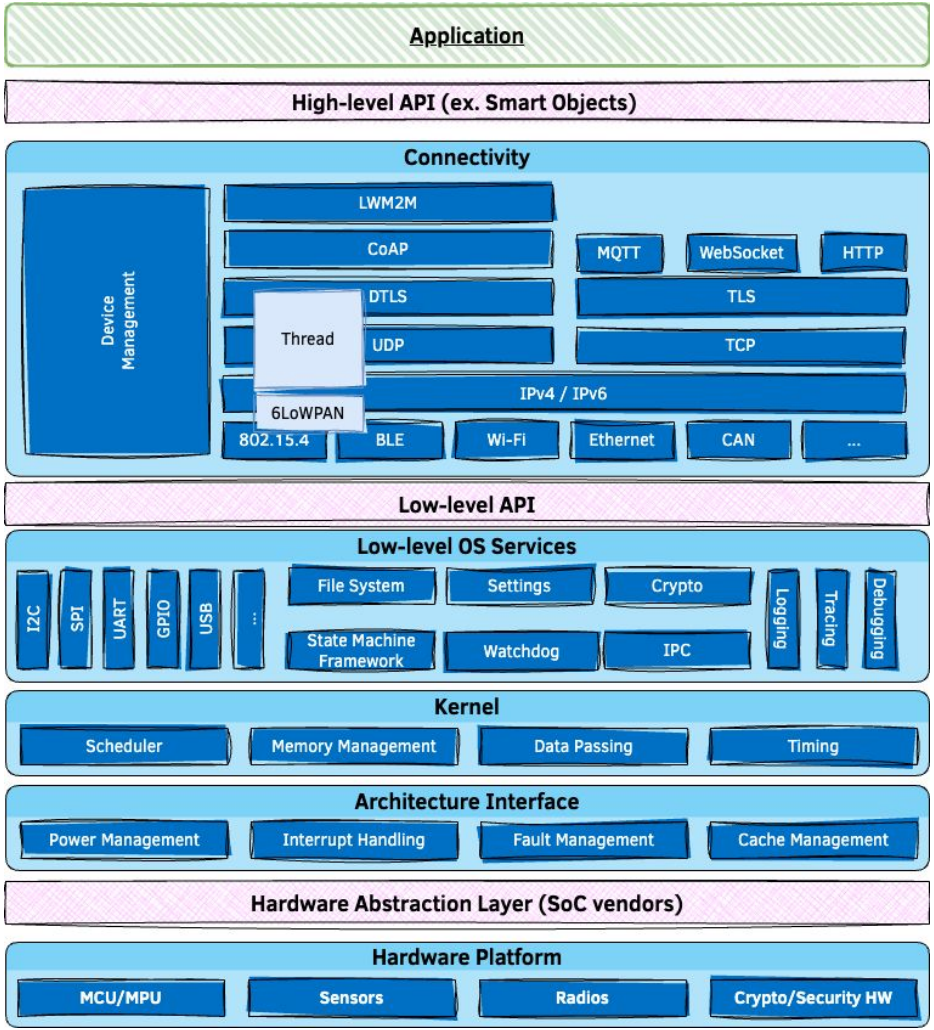
SPARC

tensilica

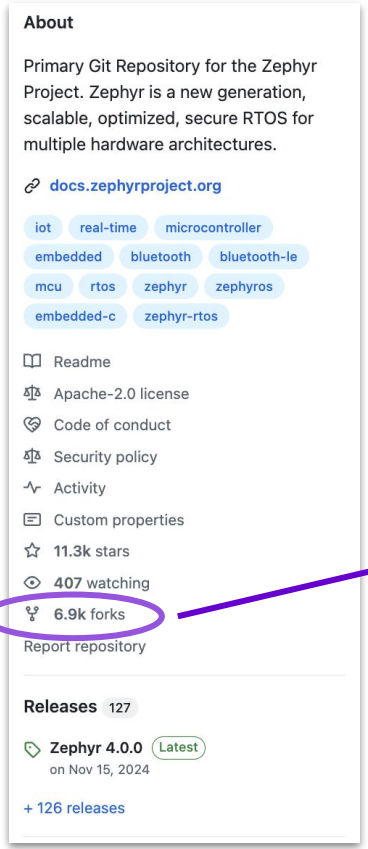
Xtensa

docs.zephyrproject.org/latest/hardware/index.html#hardware-support

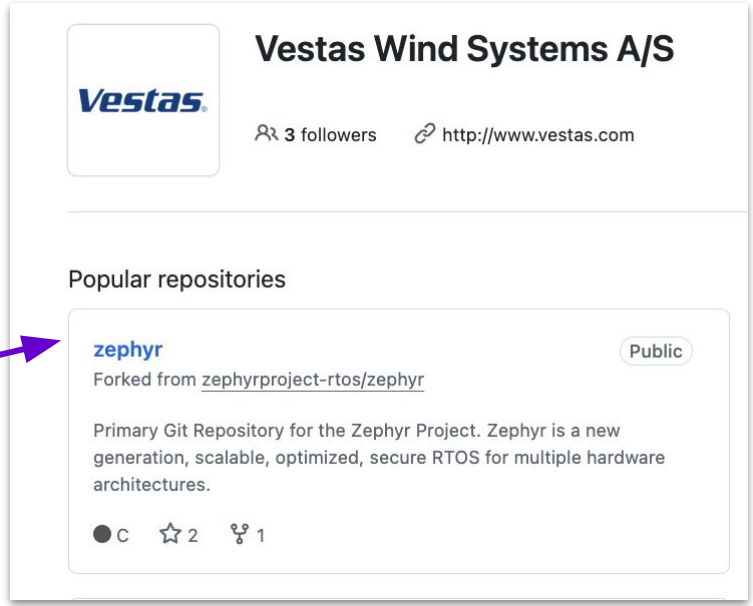
Software Architecture



Zephyr in the wild… 6.9K Forks!



Source: https://github.com/zephyrproject-rtos/zephyr



Source: https://github.com/vestas-wind-systems

Building on POSIX

- **Zephyr apps can run as native Linux applications**
  - Easier to debug/profile with native tools
  - Connect to real devices using TCP/IP, Bluetooth, CAN
  - Helps minimize hardware dependencies during the development phase
- **Re-use existing code & libraries by accessing Zephyr services through POSIX API**
  - Easier for non-embedded programmers
  - Implementation is optimized for constrained systems
  - Supported POSIX subsets: PSE51, PSE52, and BSD sockets

docs.zephyrproject.org/latest/guides/portability/posix.html

Safety & Security Focus From the Start

Exhibit B

Zephyr Project Charter (the "Charter")
The Linux Foundation
Updated August 21, 2023

1. Mission of the Zephyr Project ("Zephyr," or, alternatively, the "Project").

The mission of the Project is to:

a. deliver the best-in-class RTOS for connected resource-constrained devices, built to be secure and safe.

b. maintain an auditable code base, while taking advantage of community participation; this auditable code base is open source;

c. include participation of leading members of this ecosystem, including micro-controller manufacturers, hardware developers, software developers and other members of the ecosystem; and

d. host the infrastructure for the open source Project and sub-projects, establishing a neutral home for community meetings, events and collaborative discussions and providing structure around the business and technical governance of the Project.

Source:
https://zephyrproject.org/wp-content/uploads/2023/08/LF-Zephyr-Charter-2023.08.21.pdf

Safety and Security Focus From the Start

Exhibit B

Zephyr Project Charter (the "Charter")
The Lin
Updated

1. Mission of the Zephyr Project ("Ze

The mission of the Project is to:

a. deliver the best-in-class RTOS
to be secure and safe.

b. maintain an auditable code ba
participation; this auditable c

c. include participation of leading
controller manufacturers, hard
members of the ecosystem; an

d. host the infrastructure for the op
neutral home for community r
providing structure around the

7. Safety Committee

a. Composition – the Safety Committee members shall consist of:

i. one appointed voting representative from each Platinum Member, plus

ii. non-voting Silver Member representatives who shall not count towards quorum.

b. Responsibilities – the Safety Committee shall be responsible for:

i. the definition of the processes to ensure an auditable code base, as well as any associated certification artifacts ("Safety Artifacts");

ii. annually elect a Representative on the Safety Committee to serve as chair of the Safety Committee; and

iii. annually elect a safety architect (the "Safety Architect") , who may be different from the chair of the Safety Committee.

So what does Zephyr mean by Safety?

Safety – the freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment

So what does Zephyr mean by Safety?

Safety – the freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment

Zephyr will focus on "Functional Safety"

- the part of safety that depends on a system or equipment operating correctly in response to its inputs
- Detecting potentially dangerous conditions, resulting either in the activation of a protective or corrective device or mechanisms to prevent hazardous events or in providing mitigation measures to reduce the consequences of the hazardous event.

So what does Zephyr mean by Safety?

Safety – the freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment

Zephyr will focus on "Functional Safety"

- the part of safety that depends on a system or equipment operating correctly in response to its inputs
- Detecting potentially dangerous conditions, resulting either in the activation of a protective or corrective device or mechanisms to prevent hazardous events or in providing mitigation measures to reduce the consequences of the hazardous event.

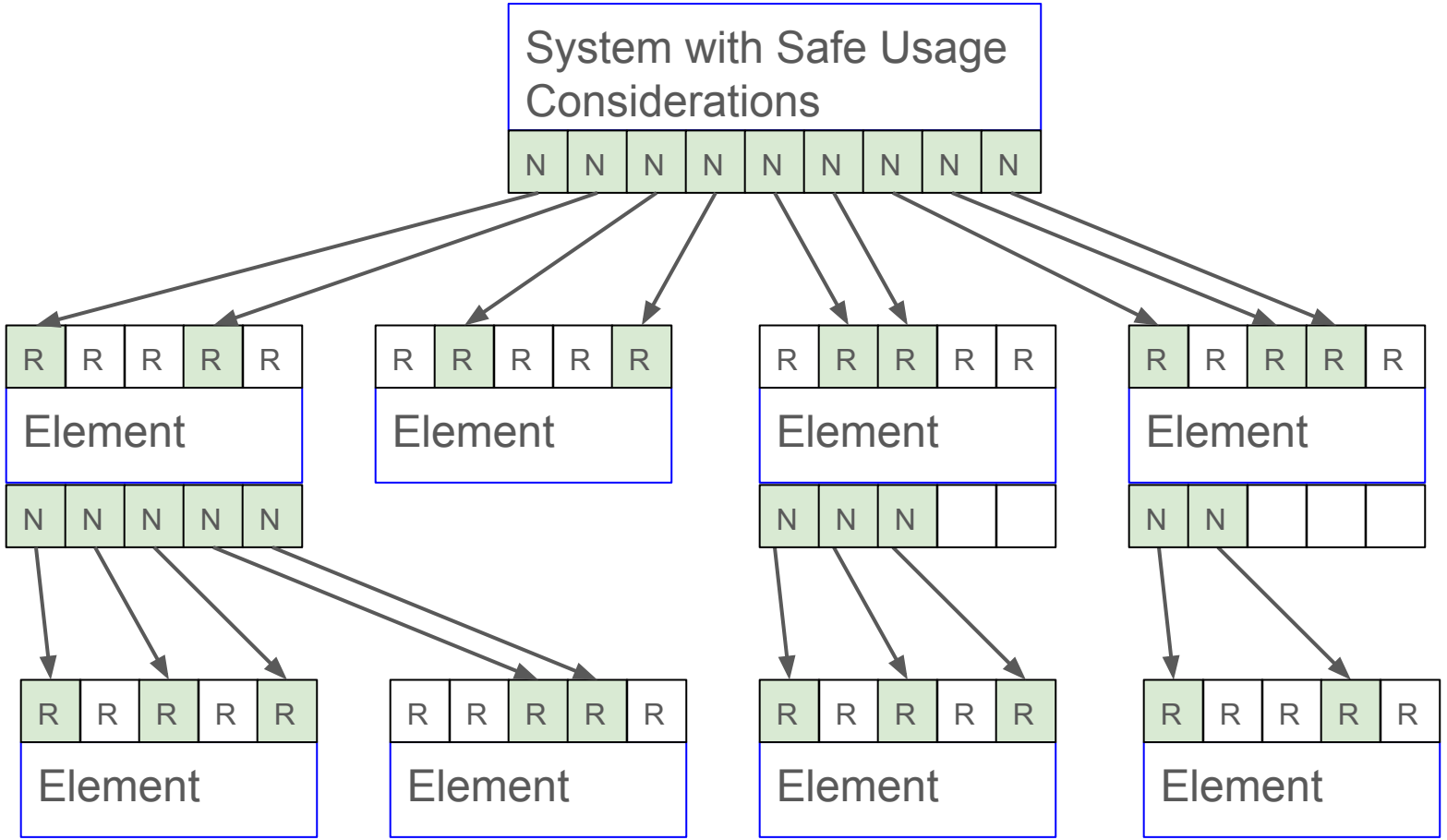
"depends on a system" ?

Systematic capability is the general assumption, that

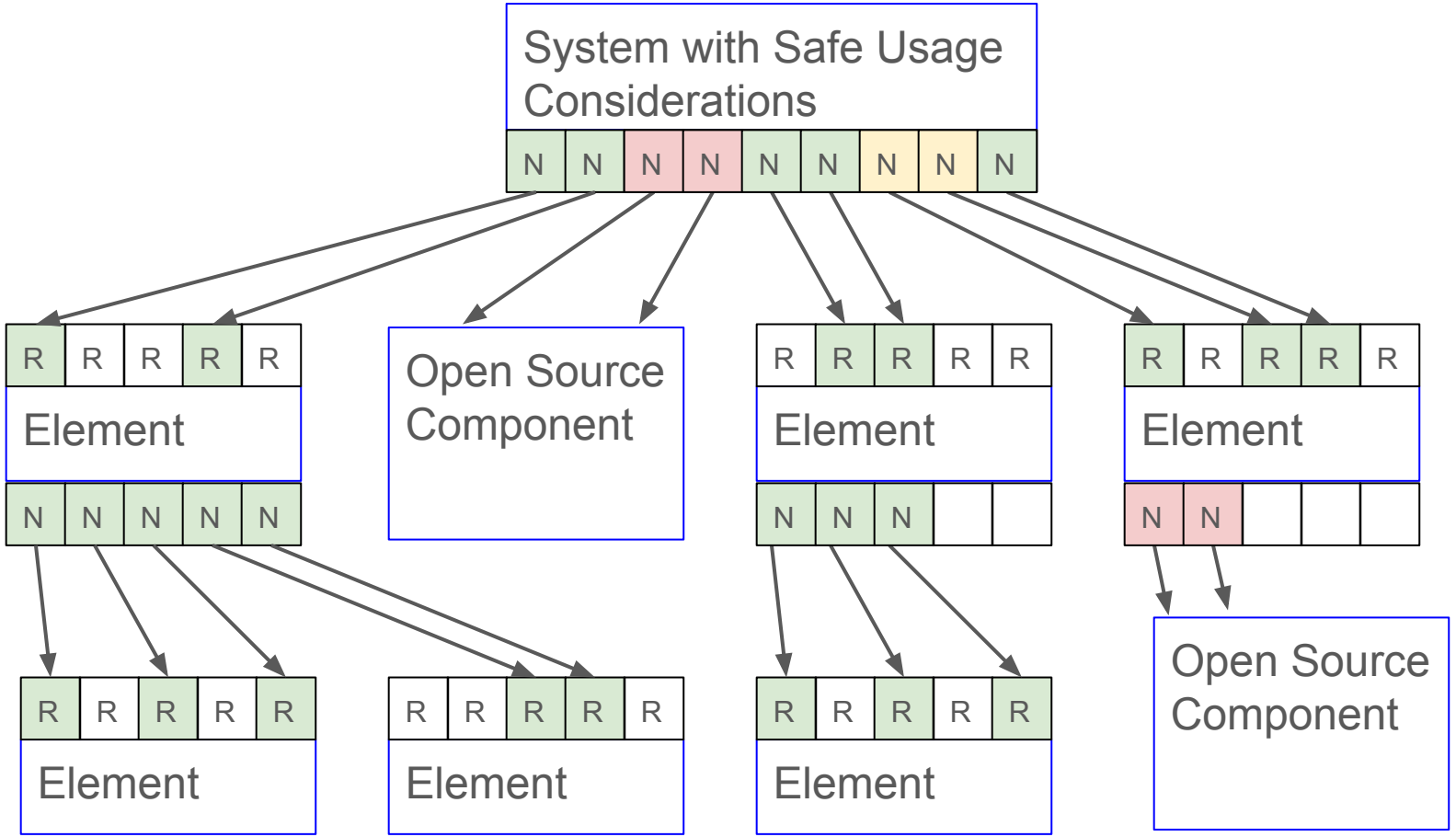
- if development, test and deployment of a system follow a specific set of tasks and
- there is evidence for adherence to these tasks
- (and under the assumption that the system architecture supports safety)

⇒ Software is capable of performing as intended

Analyzing a System for Safe Usage

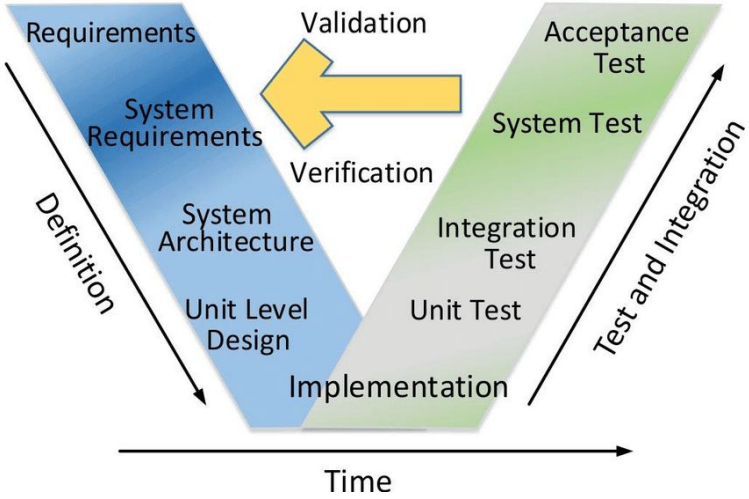


Analyzing a System with Open Source?



Safety Engineering 101: the "V-Model"

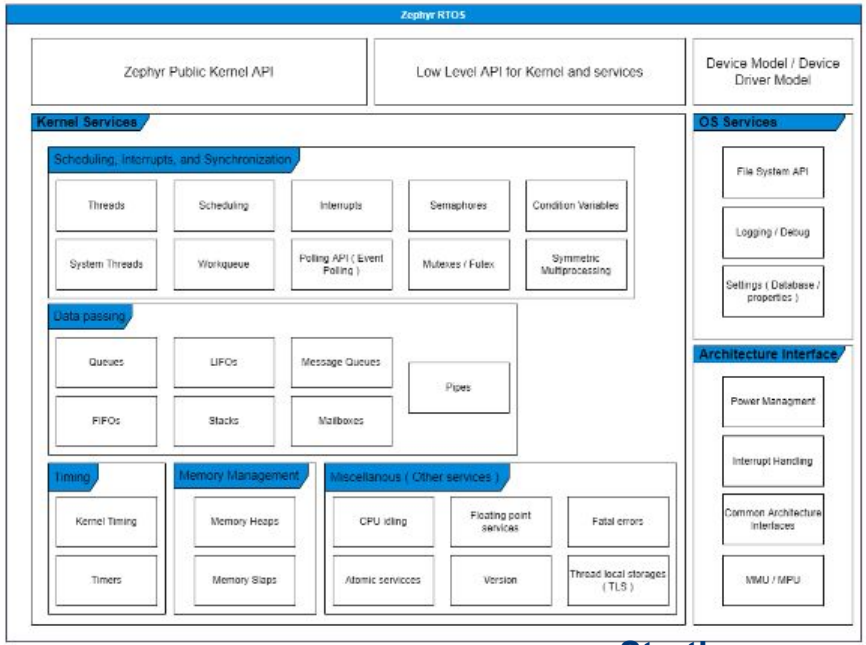
- The V-model for functional safety development originated from systems engineering practices to structure the process of designing and validating complex systems.
- It was later adapted and widely adopted in the automotive industry and other safety-critical sectors as a framework for ensuring the systematic integration of safety requirements and processes throughout each stage of the development lifecycle.
  - **ISO 26262** in the automotive industry
  - **IEC 61508** for industrial systems
  - **DO-178C** in aerospace



Source Image image provided under CC-4.0
https://www.researchgate.net/figure/The-functional-safety-development-via-V-model-14_fig4_362572593

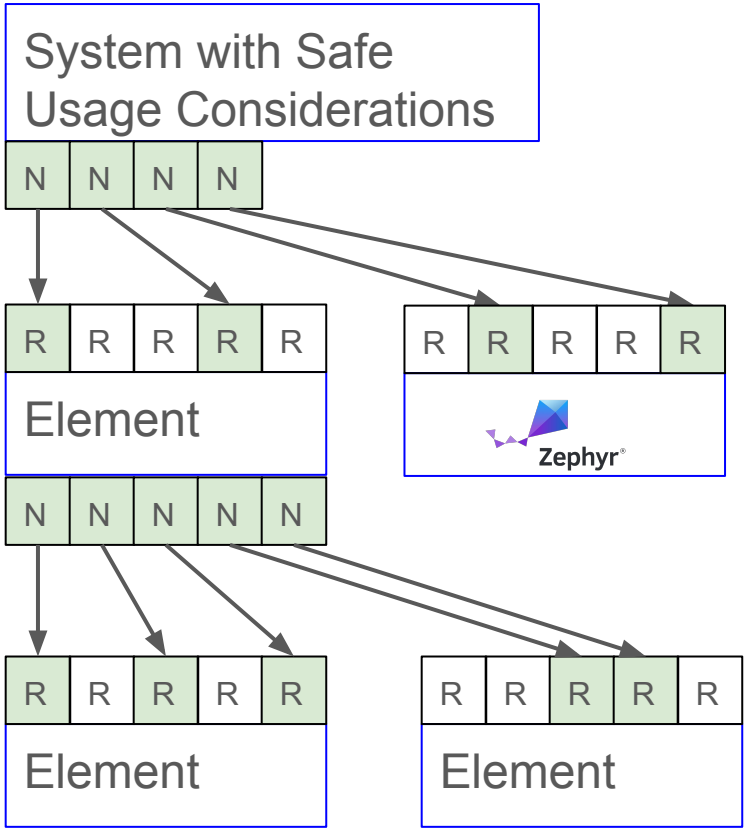
Zephyr Initial Certification Focus

- Start with a limited scope of kernel and interfaces
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s)
- Option for 26262 certification has been included in contract with certification authority should there be sufficient member interest



Scope can be **extended** to include **additional components** with associated **requirements** and **traceability** as determined by the safety committee

Safety Element out of Context - SEooC



Development and verification independent of a specific context or application

Provides integration and operation information for safe system integration

Comes with sufficient evidence, that it can be integrated to a safety relevant system.

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.12

“Where a pre-existing software element is reused to implement all or part of a safety function, the element shall meet both requirements a) and b) below for systematic safety integrity:

a) Meet the requirements of one of the following compliance routes:
- Route 1s: compliant development. Compliance with the requirements of this standard for the avoidance and control of systematic faults in software;
- Route 2s: proven in use. Provide evidence that the element is proven in use. See 7.4.10 of IEC 61508-2;
- Route 3s: assessment of non-compliant development. Compliance with 7.4.2.13

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.12

“Where a pre-existing software element is reused to implement all or part of a safety function, the element shall meet both requirements a) and b) below for systematic safety integrity:

a) Meet the requirements of one of the following compliance routes:
- Route 1s: compliant development. Compliance with the requirements of this standard for the avoidance and control of systematic faults in software;
- Route 2s: proven in use. Provide evidence that the element is proven in use. See 7.4.10 of IEC 61508-2;
- Route 3s: assessment of non-compliant development. Compliance with 7.4.2.13

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

a) The software safety requirement specification for the element [...] shall be documented to the same degree of precision as would be required by this standard for any safety related element of the same systematic capability [...]

⇒ Creation of System and Software Specification

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

b) The justification for use of a software element shall provide evidence that the desired safety properties specified [...] have been considered [...].

⇒ Definition of the Safety Claims
⇒ Traceability to requirements, code and tests

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

c) The element's design shall be documented to a degree of precision, sufficient to provide evidence of compliance with the requirement specification and the required systematic capability. [...]The justification for use of a software element shall provide evidence that the desired safety properties specified [...] have been considered [...].

⇒ Reuse of the information provided by the docs.zephyrproject.org

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

d) The evidence required in 7.4.2.13 a) and b) shall cover the software's integration with the hardware. [...]

⇒ use the existing tests
⇒ establishing traceability
⇒ enhancing coverage as needed

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

e) There shall be evidence that the element has been subject to verification and validation using a systematic approach with documented testing and review of all parts of the element's design and code [...]

⇒ use the existing tests
⇒ establishing traceability
⇒ enhancing coverage as needed

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

f) Where the software element provides functions which are not required in the safety related system, then evidence shall be provided that the unwanted functions will not prevent the E/EE/EP system from meeting its safety requirements.

⇒ Providing a defined safety scope definition

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

g) There shall be evidence that all credible failure mechanisms of the software element have been identified and that appropriate mitigation measures have been implemented.

⇒ creating requirements
⇒ establishing traceability to code & tests

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

h) The planning for use of the element shall identify the configuration of the software element, the software and hardware run-time environment and if necessary the configuration of the compilation / linking system.

⇒ information that will be discussed in the Safety Manual

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.13

To comply with Route 3s a pre-existing software element shall meet all of the following requirements a) to i)

i) The justification for use of the element shall be valid only those applications which respect the assumptions made in the [...] safety manual [...]

⇒ creation of the Safety Manual

Systemic Capability for Safety

IEC 61508-3, Clause 7.4.2.12

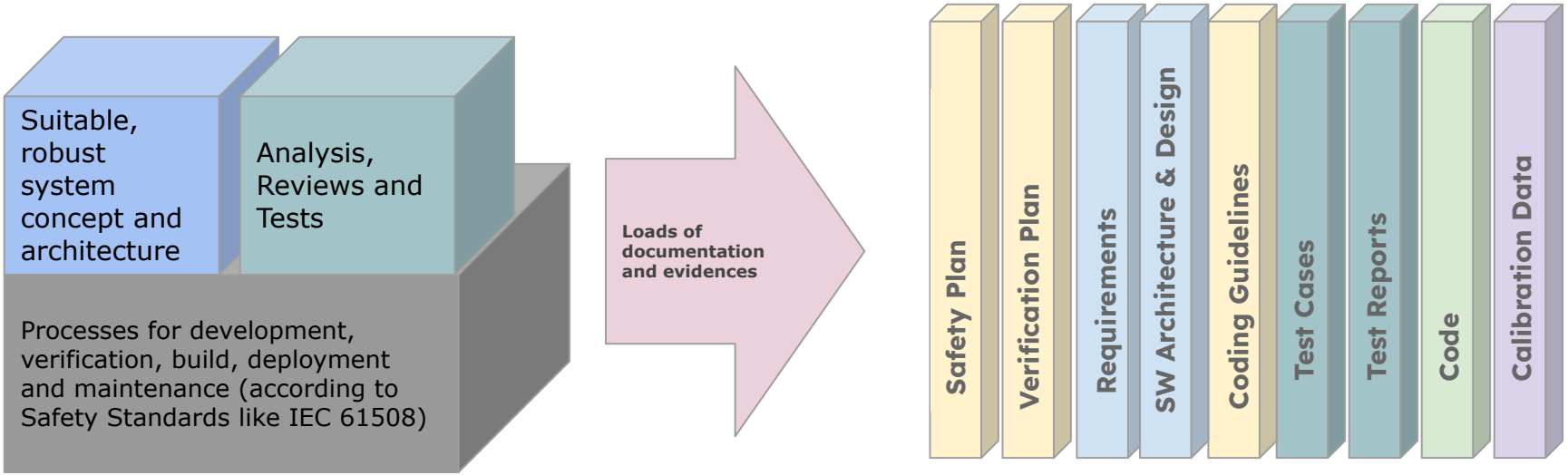
“Where a pre-existing software element is reused to implement all or part of a safety function, the element shall meet both requirements a) and b) below for systematic safety integrity:

b) provides a safety manual [...]

⇒ creation of the Safety Manual

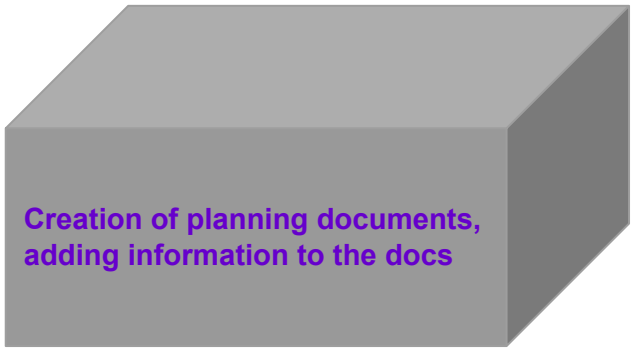
What is Functional Safety aiming for ?

Safety Architecture and Documentation



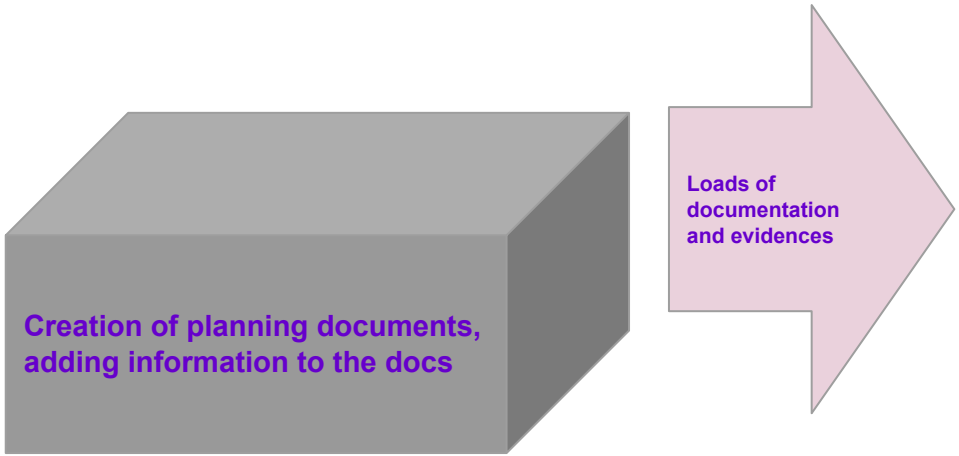
How is the Zephyr Project addressing this?

Following Route 3s by creating the missing work products



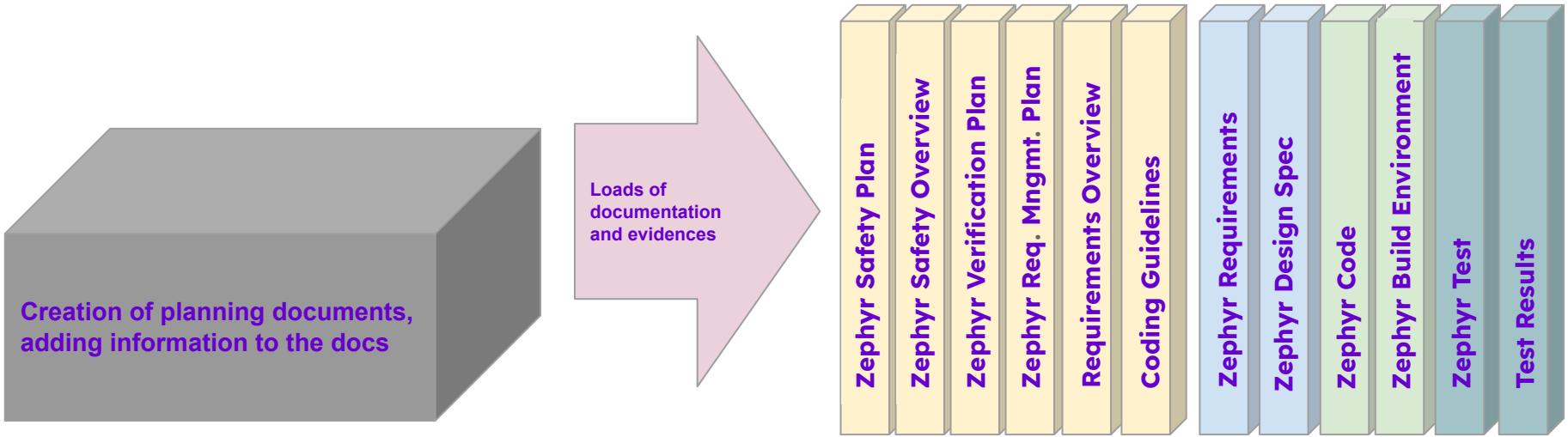
How is the Zephyr Project addressing this?

Following Route 3s by creating the missing work products



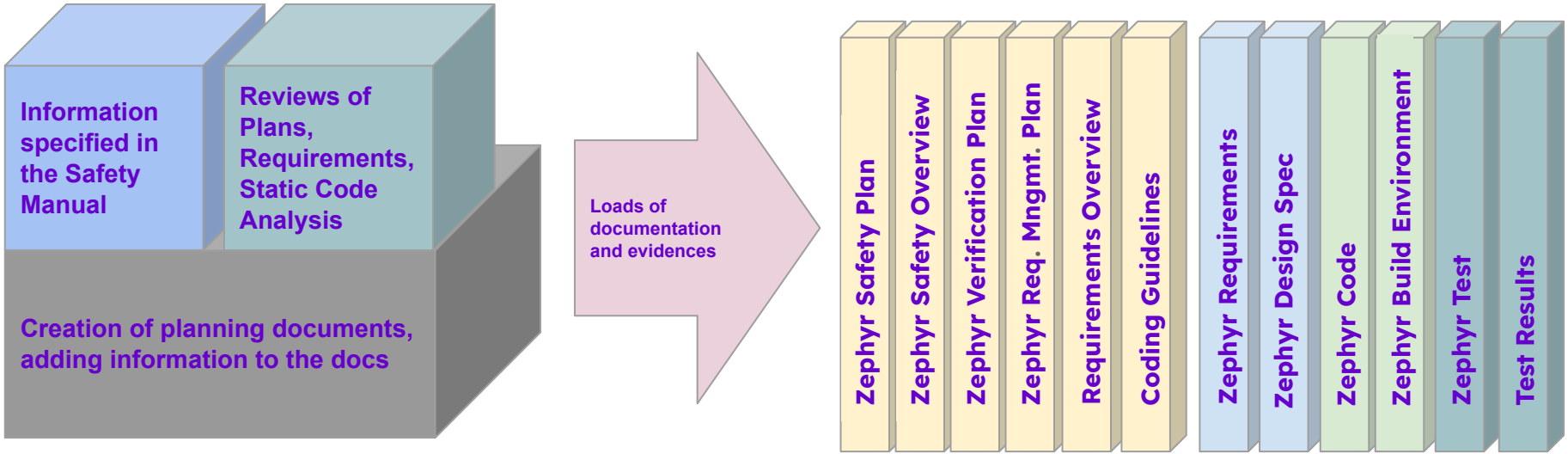
How is the Zephyr Project addressing this?

Following Route 3s by creating the missing work products



How is the Zephyr Project addressing this?

Following Route 3s by creating the missing work products



Creating the Work Products

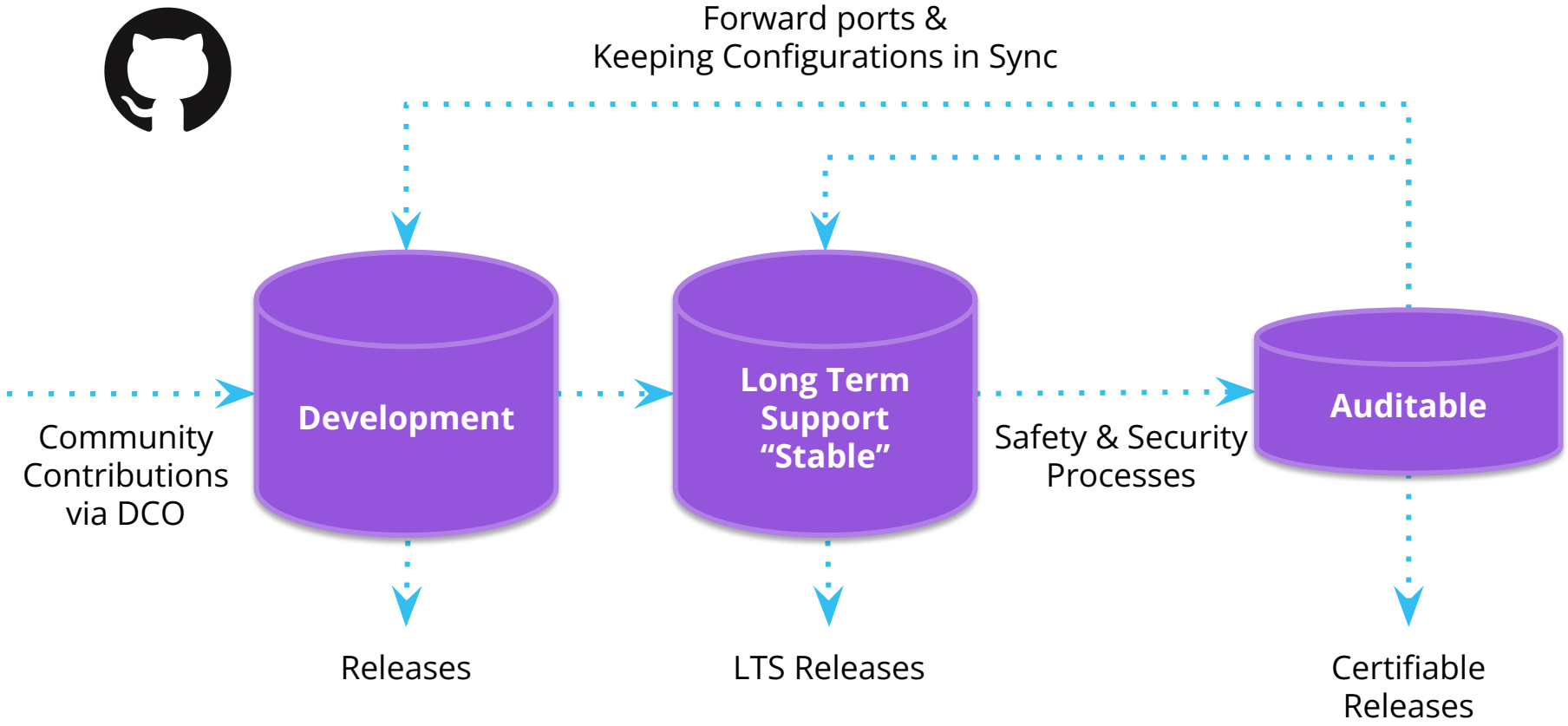
Safety Committee Role

- Safety Certification strategy decisions
  - Scope of certification
  - Certification standards
  - Certification timeline
- Assessment and audit specific tasks
- Owner of certification artefacts
- **Participation limited** to the project's platinum members, the safety architect, the safety chair, functional safety manager and project staff

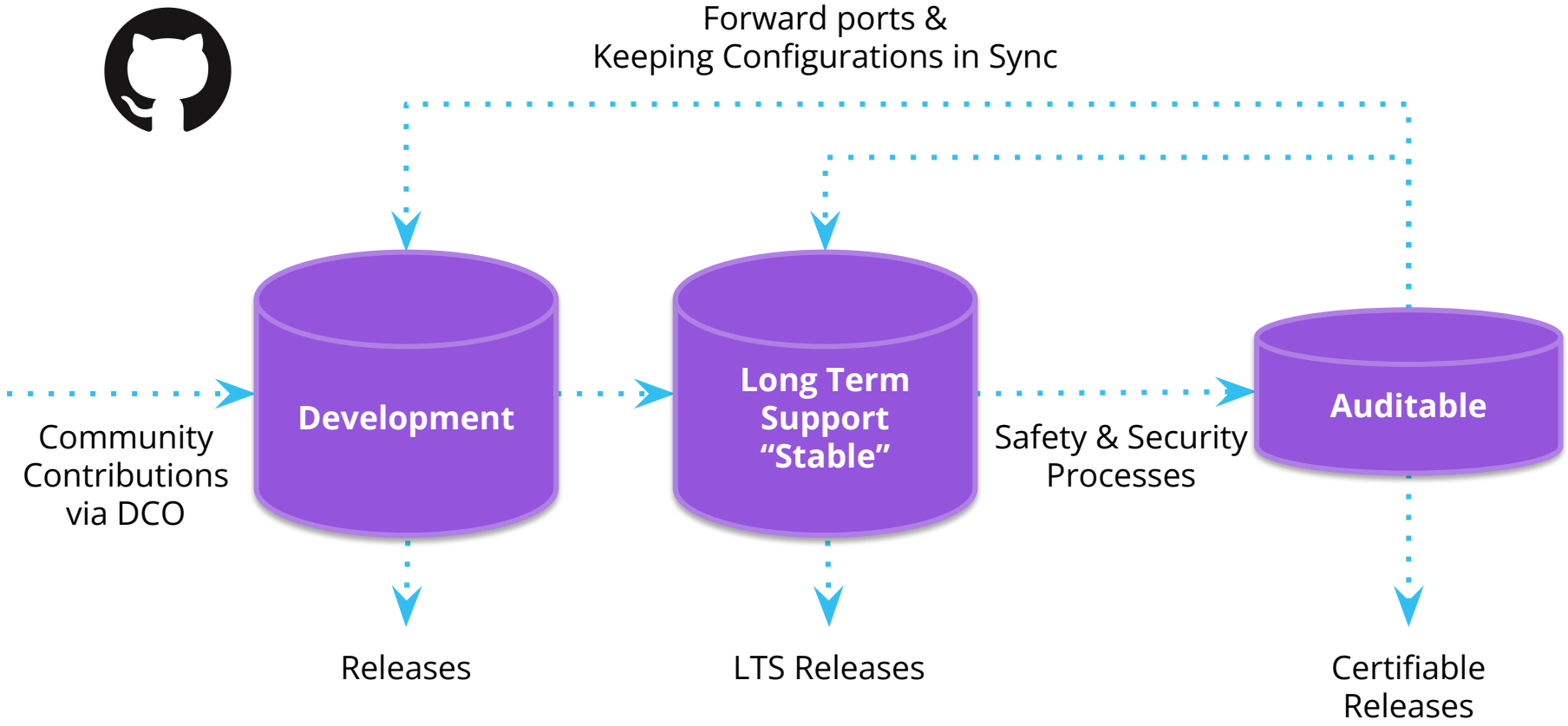
Safety Working Group Role

- Creating/deriving and documenting the requirements for project code
- Establishing traceability between requirements, code and relevant tests
- Extending testing coverage as needed
- Setting up requirements management tooling
- Working on the creation of the required documentation and evidence
- **Open to everyone to participate**

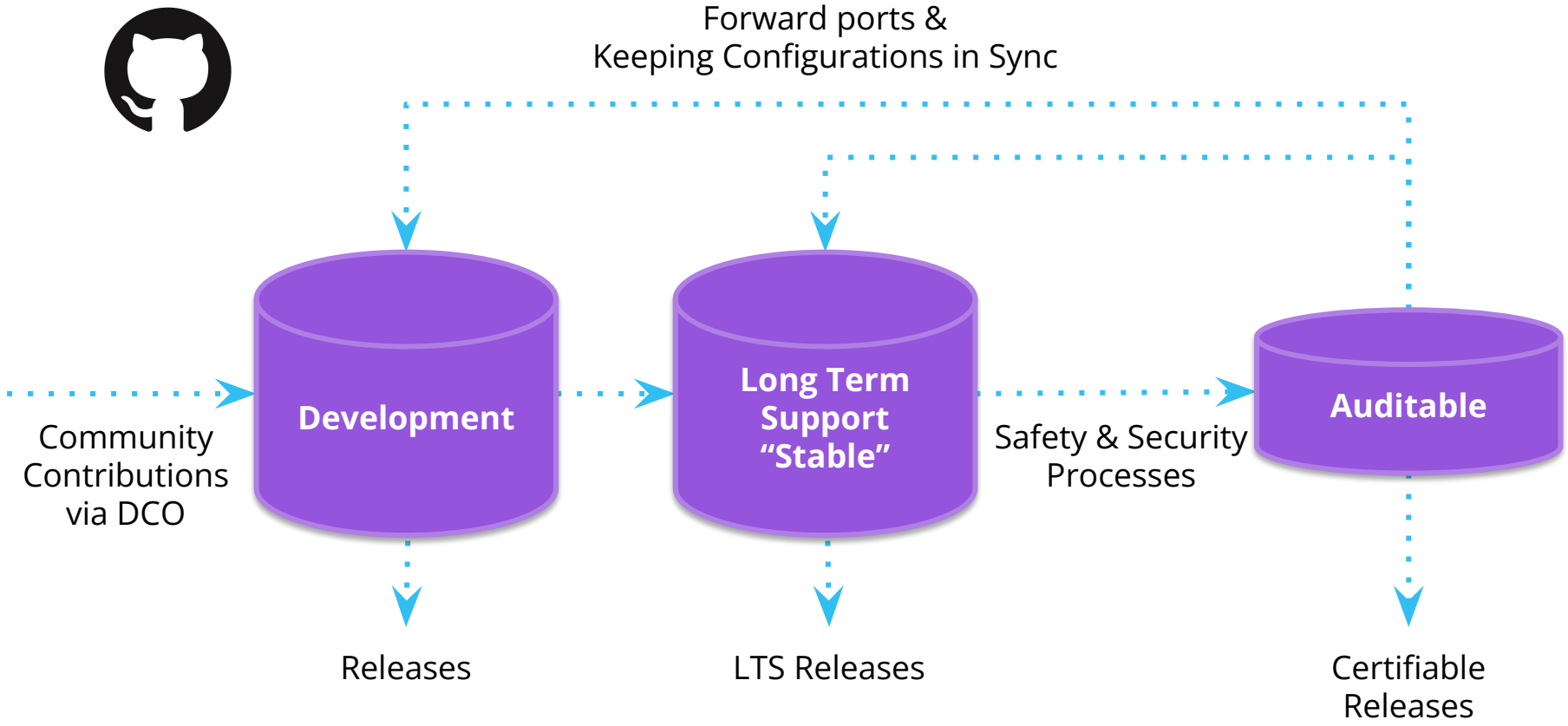
Code Repositories



Traceability on "Development"

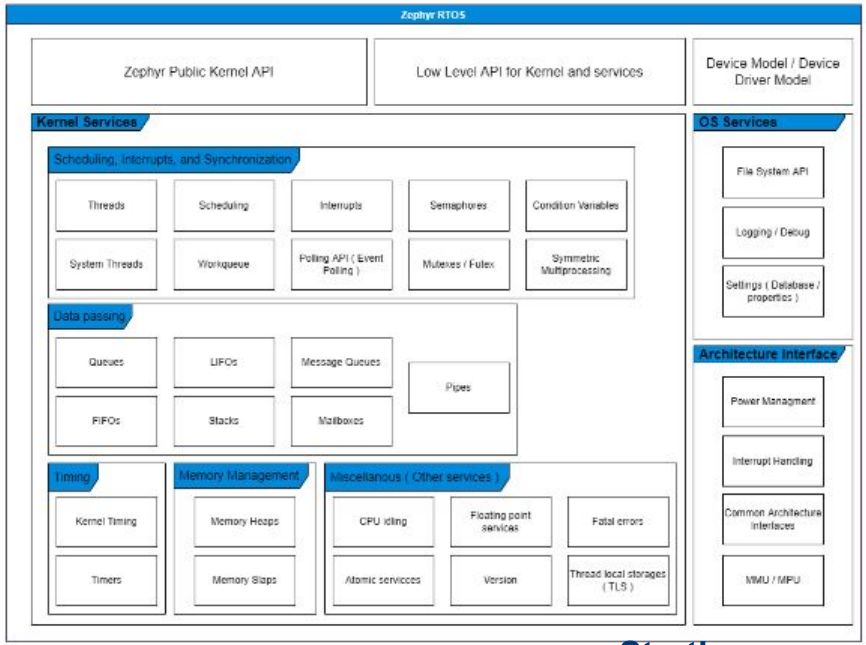


"Auditable" is subset of "Development"



Zephyr Initial Certification Focus

- Start with a limited scope of kernel and interfaces
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s)
- Option for 26262 certification has been included in contract with certification authority should there be sufficient member interest



Scope can be **extended** to include **additional components** with associated **requirements** and **traceability** as determined by the safety committee

Work Product Strategy

Strategy for artefact creation:

- use developer friendly tooling
- use known workflows on GitHub
- reuse as much as we can from the docs.zephyrproject.org

Enhancements to project documentation

- Sphinx in docs.zephyrproject.org

New artefacts ⇒ StrictDoc

- Requirements
- Safety Plan
- Safety Manual
- …

Evidence ⇒ StrictDoc

- Assessments
- Checklists
- …

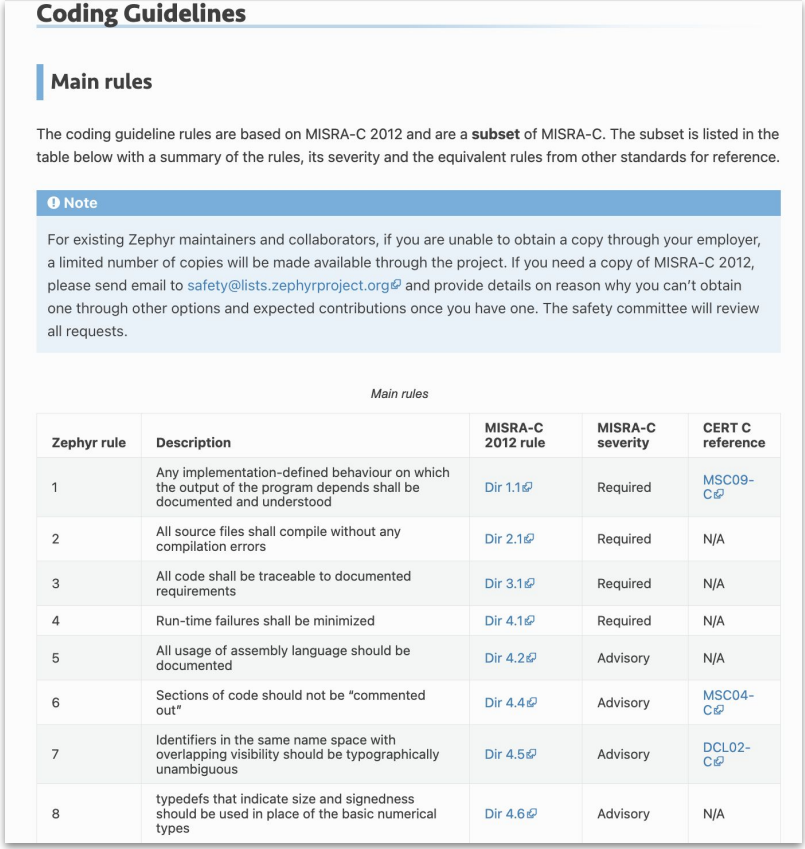
Compliance with Coding Standards

Project already has defined Coding Guidelines derived from MISRA rules and CERT-C references

Identification of Coding Guideline violations and adaption of the code

- Initially done by Bugseng on a separate branch
- Work merged to the main branch of "Development" in 2024

Coming soon: Static Analysis in the CI to check for adherence on each patch



Coding Guidelines

Main rules

The coding guideline rules are based on MISRA-C 2012 and are a **subset** of MISRA-C. The subset is listed in the table below with a summary of the rules, its severity and the equivalent rules from other standards for reference.

Note

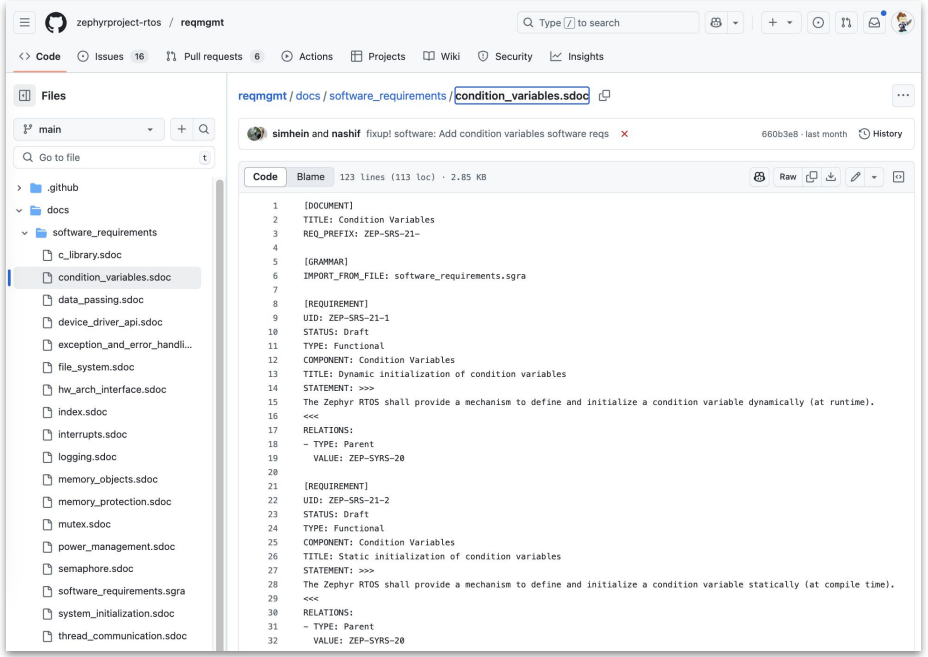
For existing Zephyr maintainers and collaborators, if you are unable to obtain a copy through your employer, a limited number of copies will be made available through the project. If you need a copy of MISRA-C 2012, please send email to safety@lists.zephyrproject.org and provide details on reason why you can't obtain one through other options and expected contributions once you have one. The safety committee will review all requests.

Main rules

Zephyr rule	Description	MISRA-C 2012 rule	MISRA-C severity	CERT C reference
1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood	Dir 1.1	Required	MSC09-C
2	All source files shall compile without any compilation errors	Dir 2.1	Required	N/A
3	All code shall be traceable to documented requirements	Dir 3.1	Required	N/A
4	Run-time failures shall be minimized	Dir 4.1	Required	N/A
5	All usage of assembly language should be documented	Dir 4.2	Advisory	N/A
6	Sections of code should not be "commented out"	Dir 4.4	Advisory	MSC04-C
7	Identifiers in the same name space with overlapping visibility should be typographically unambiguous	Dir 4.5	Advisory	DCL02-C
8	typedefs that indicate size and signedness should be used in place of the basic numerical types	Dir 4.6	Advisory	N/A

Critical Path: Requirement & Traceability

- Established hierarchical structure of requirements
- Capturing the requirements in StrictDoc which is working towards import/export of SPDX
- Capturing plans in StrictDoc where each planning item (like Safety Plan) is tracked as a requirement
- Capturing Assessment checklist in StrictDoc where each checkpoint is a requirement, tracing to the Zephyr's evidences



Safety Assessment & Certification Plan

Phase 1 - Concept Phase

Approval of the overall plans and strategy, scope and high level requirements & architecture

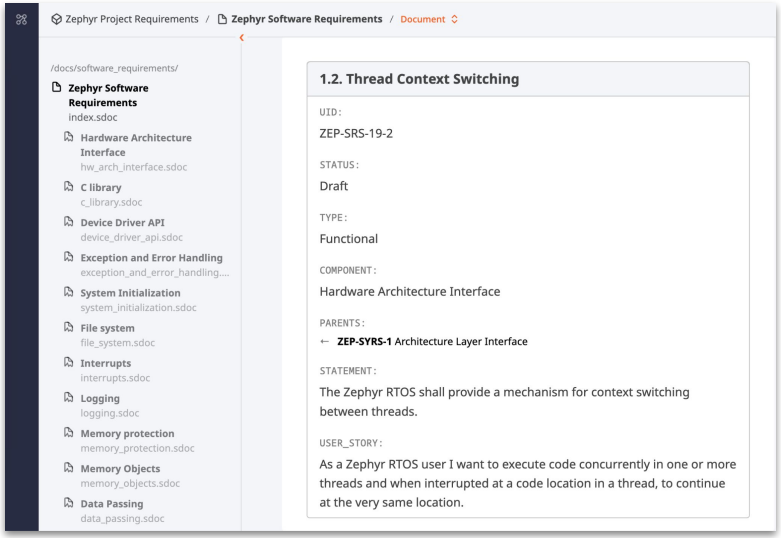
Phase 2 - Detailed Phase

Filling the gaps:

- Completeness of requirements
- Safety Analysis
- Traceability
- Verification & Test Coverage

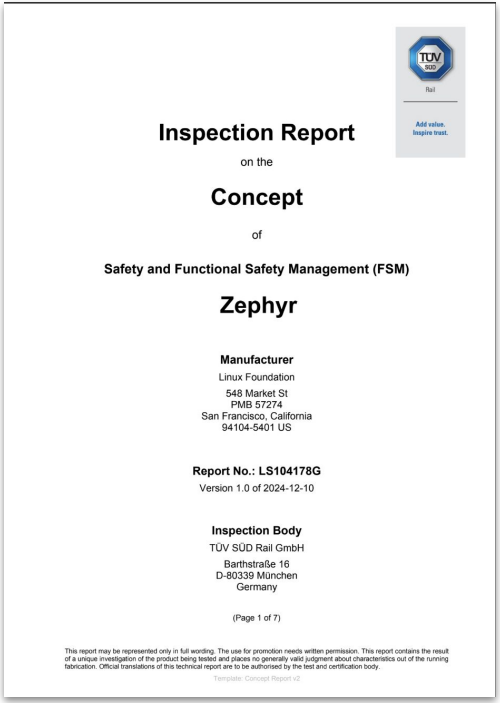
Status Today

- Coding Guidelines established based on MISRA rules and applied
- Static Analysis tooling to check for adherence to Guidelines selected for future contributions
- Reference Requirements and Traceability started in the code base
- Automatic human readable documentation of requirements from StrictDoc rules is available



Status Today

- Coding Guidelines established based on MISRA rules and applied
- Static Analysis tooling to check for adherence to Guidelines selected for future contributions
- Reference Requirements and Traceability started in the code base
- Automatic human readable documentation of requirements from StrictDoc rules is available
- **Formal concept approval and Phase 1 is complete … on to Phase 2**



Inspection Report

on the

Concept

of

Safety and Functional Safety Management (FSM)

Zephyr

Manufacturer

Linux Foundation

548 Market St
PMB 57274
San Francisco, California
94104-5401 US

Report No.: LS104178G

Version 1.0 of 2024-12-10

Inspection Body

TÜV SÜD Rail GmbH

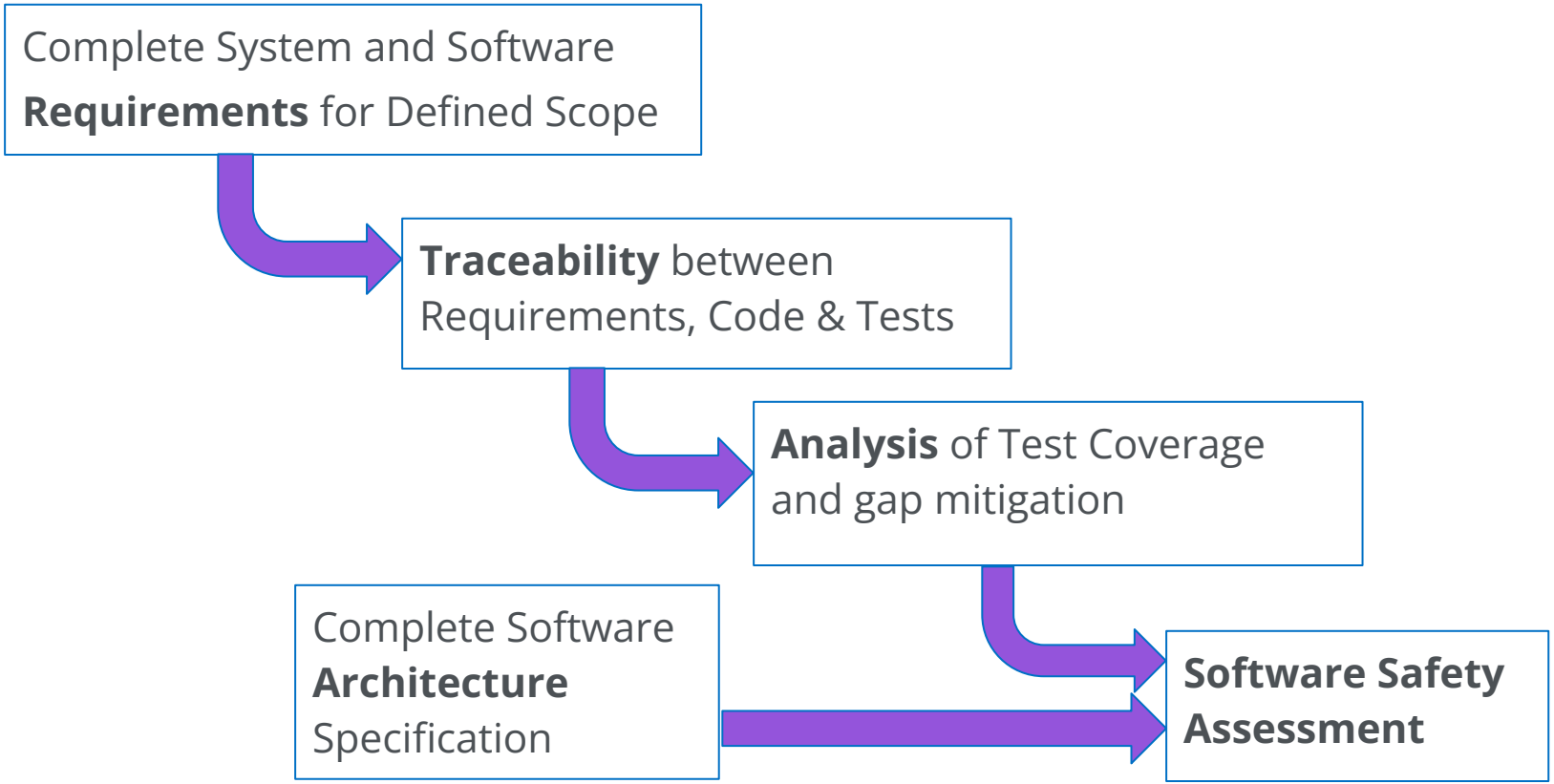
Barthstraße 16
D-80339 München
Germany

(Page 1 of 7)

This report may be represented only in full wording. The use for promotion needs written permission. This report contains the result of a unique investigation of the product being tested and places no generally valid judgment about characteristics out of the running fabrication. Official translations of this technical report are to be authorised by the test and certification body.

Template: Concept Report v2

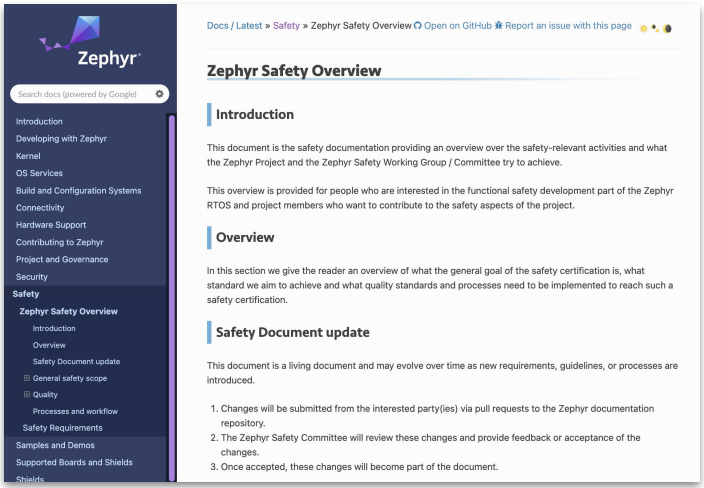
Critical Path to Phase 2



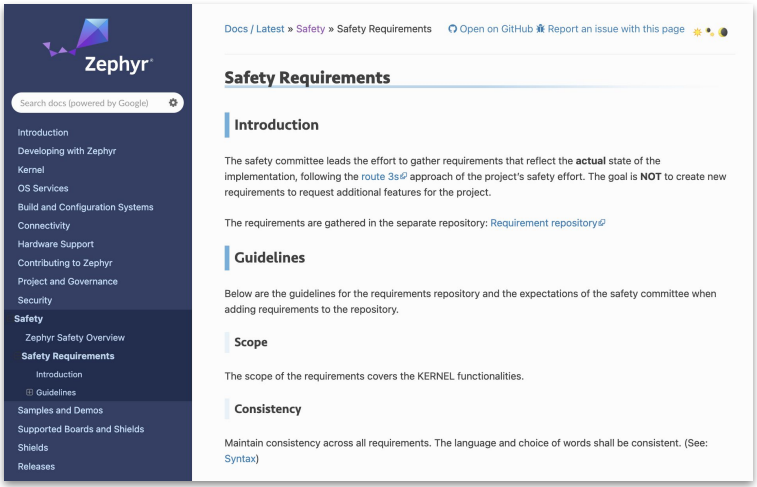
Want to help make it happen faster?

Read the docs to start …

Safety Overview



Requirements Guideline

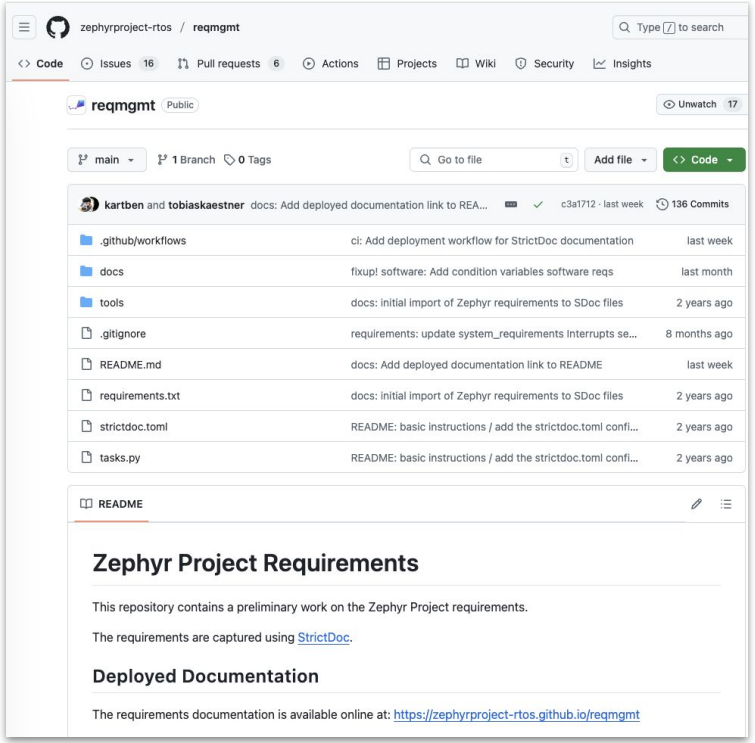


Want to help make it happen faster?

Contribute in our repos:

Requirements Repository

- Grab a PR and give some feedback
- Read through the existing requirements and submit an issue if needed
- Get familiar with StrictDoc and start creating new requirements :-)



https://github.com/zephyrproject-rtos/reqmgmt

Join us!

Zephyr®

Participate in the Working Group:

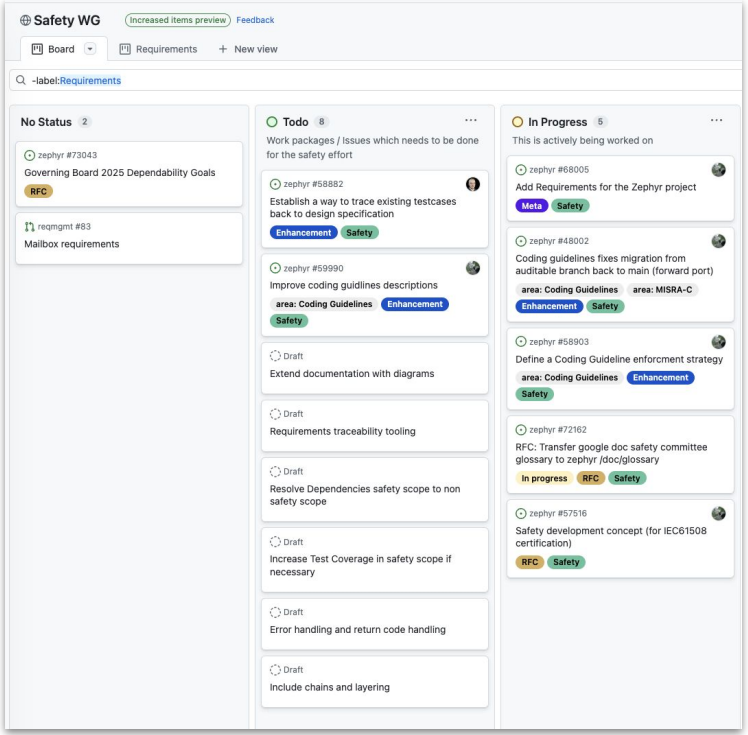
[Safety Working Group Project]

- Have a look at the tasks
- Grab an existing task
- Or submit a new task

https://lists.zephyrproject.org/g/safety-wg

- Biweekly meeting calendar information and mailing list subscription.

https://discord.gg/mgZkSmq2



Source: https://github.com/orgs/zephyrproject-rtos/projects/23/views/1