THE LINUX FOUNDATION
OPEN SOURCE SUMMIT JAPAN

AI_dev
Open Source GenAI & ML Summit

AUTOMOTIVE LINUX SUMMIT

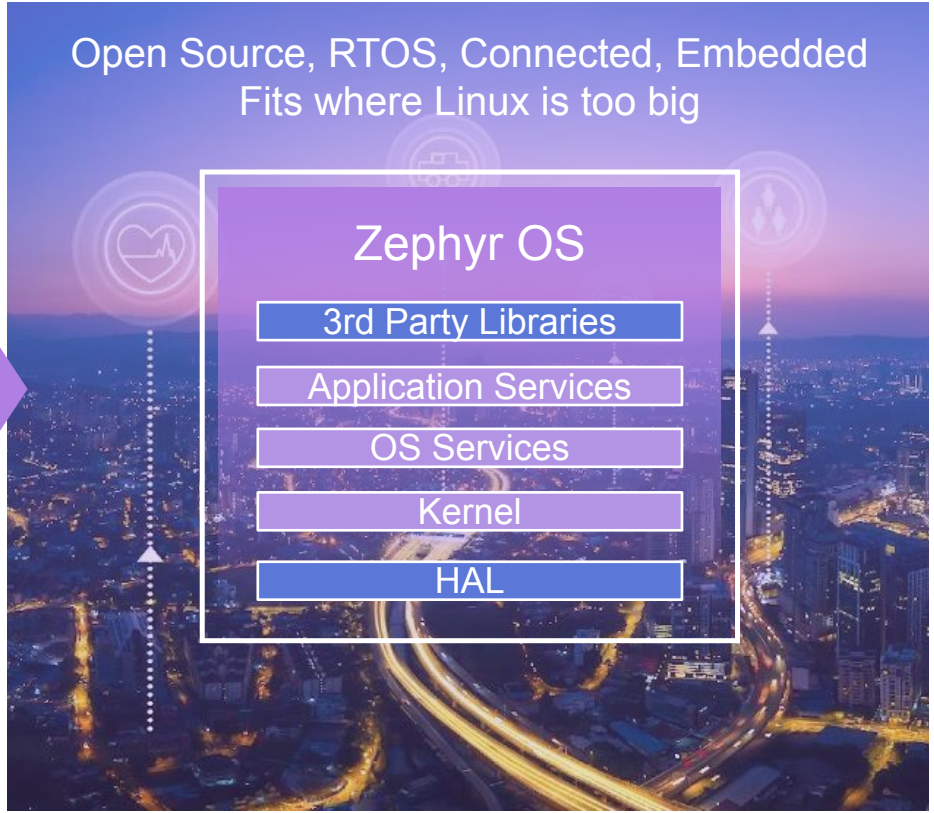
Zephyr: Learnings from Working on Safety Certification in the Open

Kate Stewart,
VP Dependable Embedded Systems,
The Linux Foundation

With thanks to Nicole Pappler, Stephane Parenti, Tobias Kästner, Simon Heim & Anas Nashif

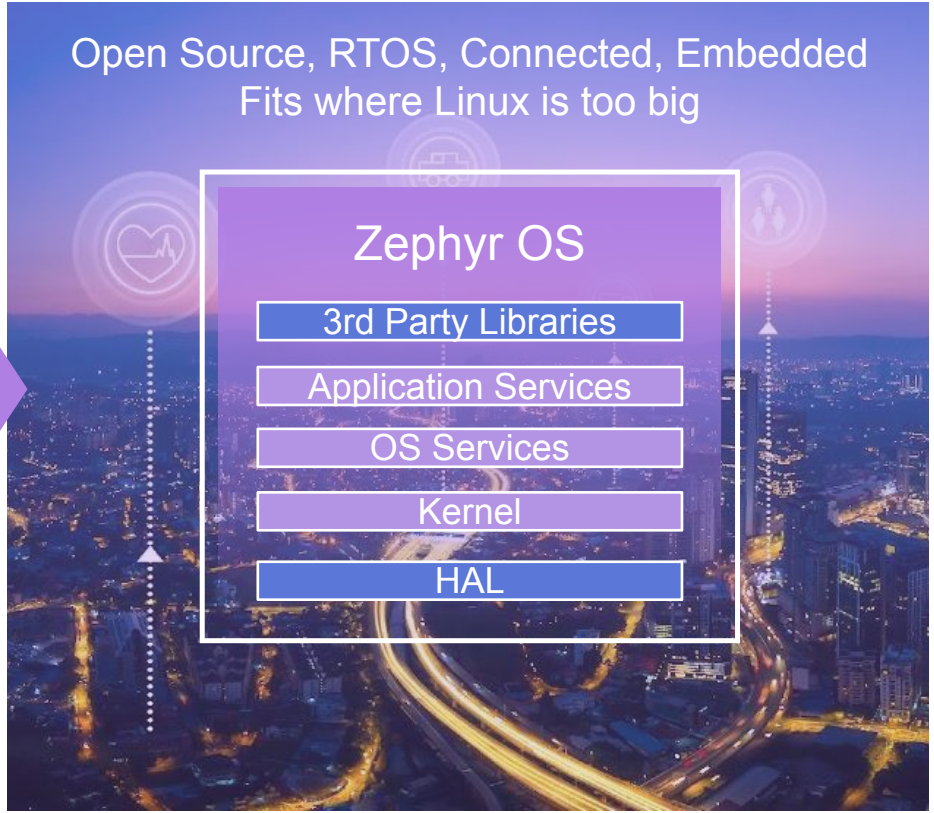
Zephyr Project

- **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**



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**



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.


ARE WE
THERE YET?

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" ?

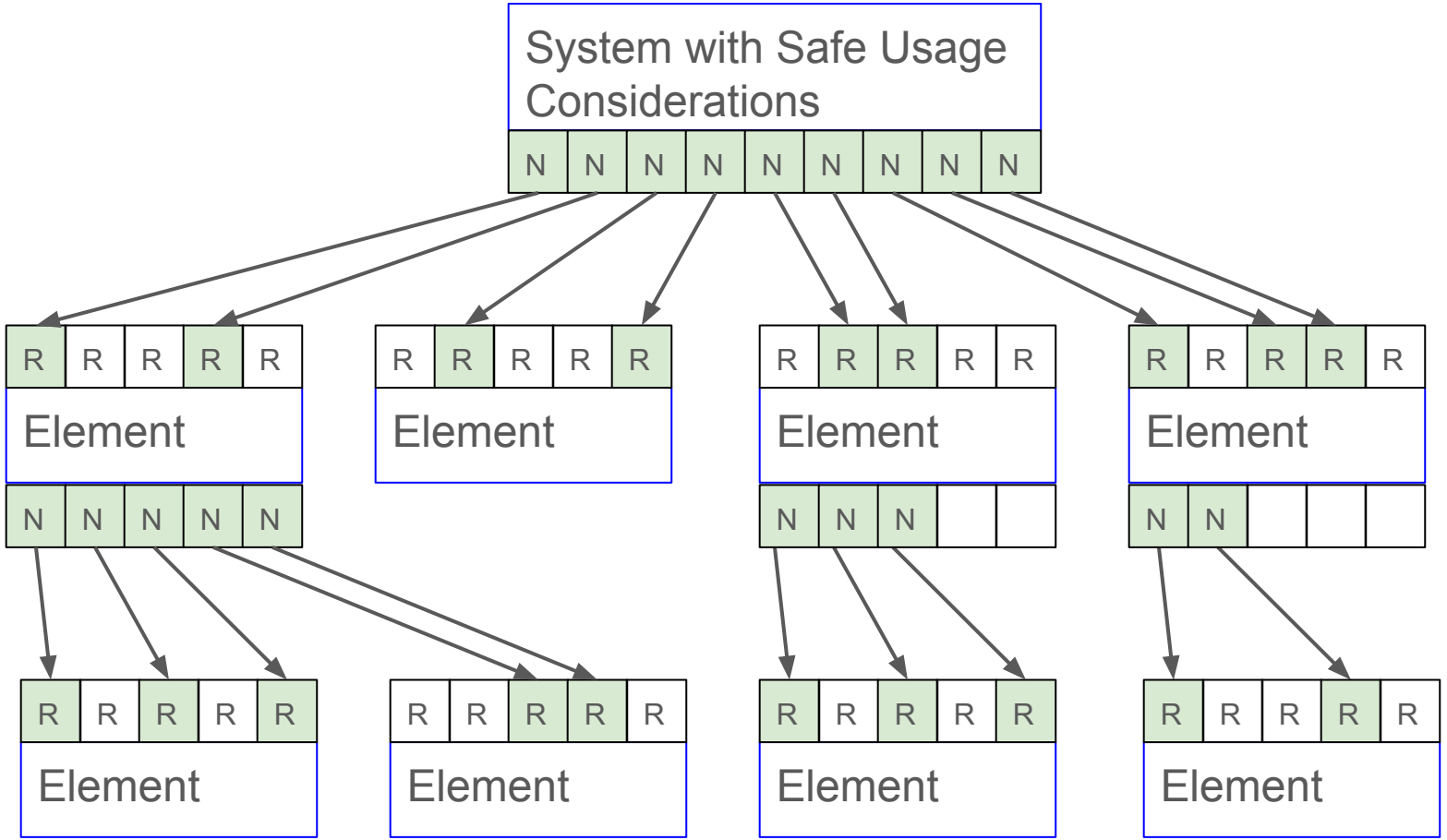
Zephyr®

Systematic capability is the general assumption, that

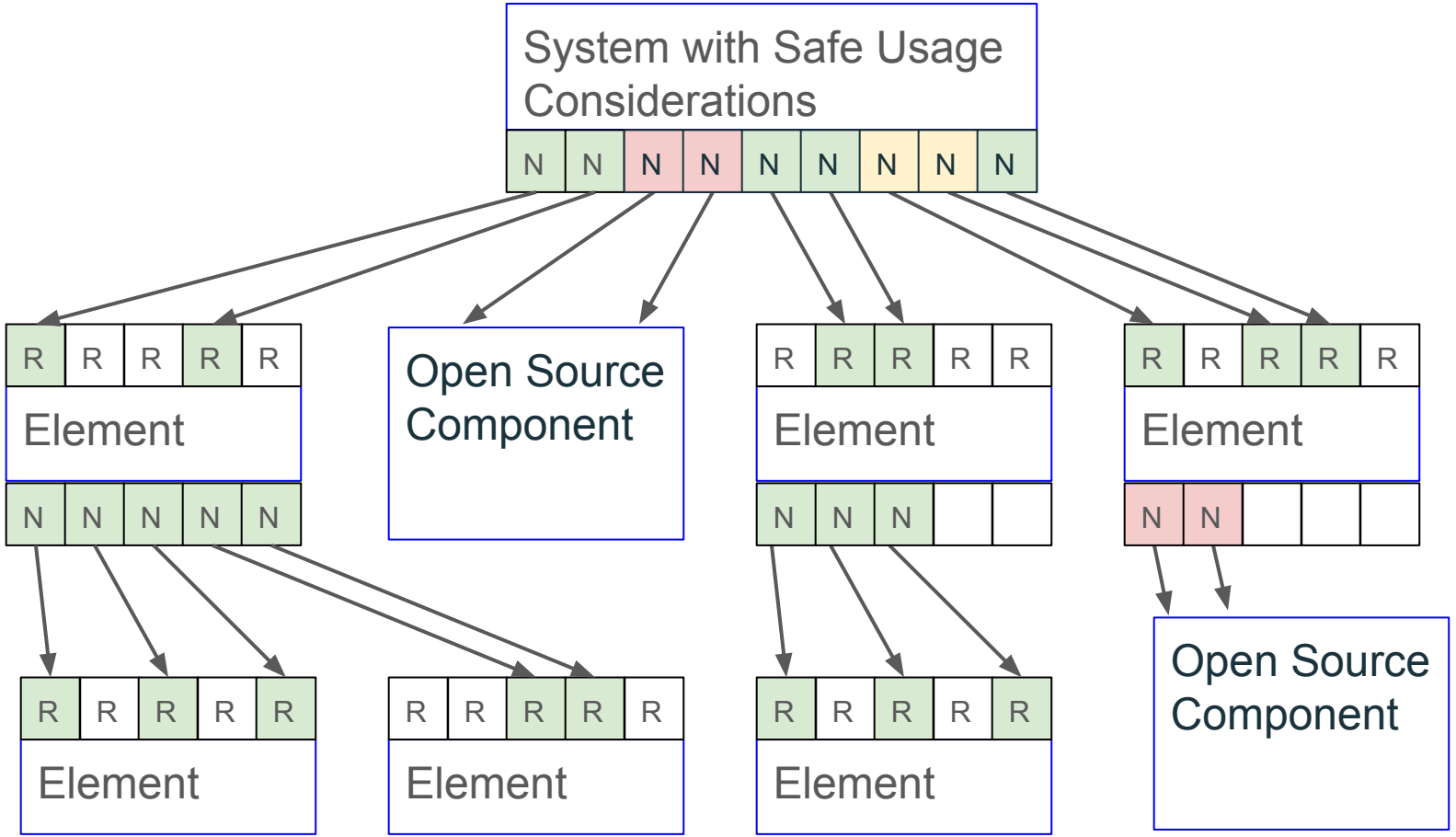
- if development, test and deployment of a system follow a specific set of tasks and
- there is evidence for adherence to these tasks
- (and under the assumption that the system architecture supports safety)

⇒ Software is capable of performing as intended

Analyzing a System for Safe Usage

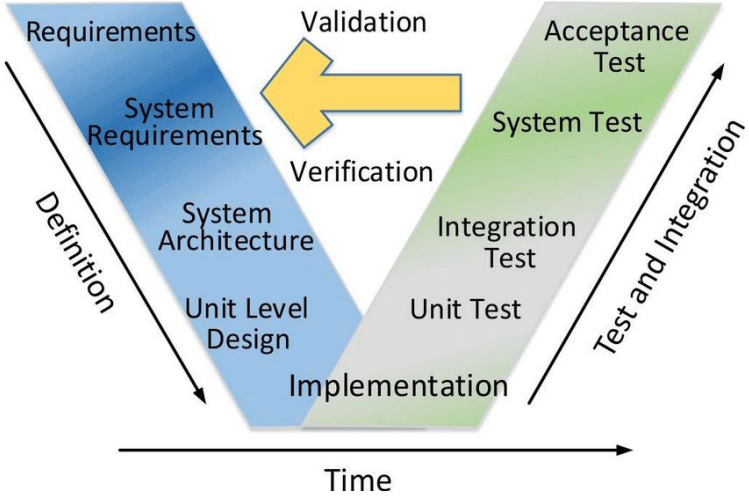


Analyzing a System with Open Source?



Safety Engineering 101: the "V-Model"

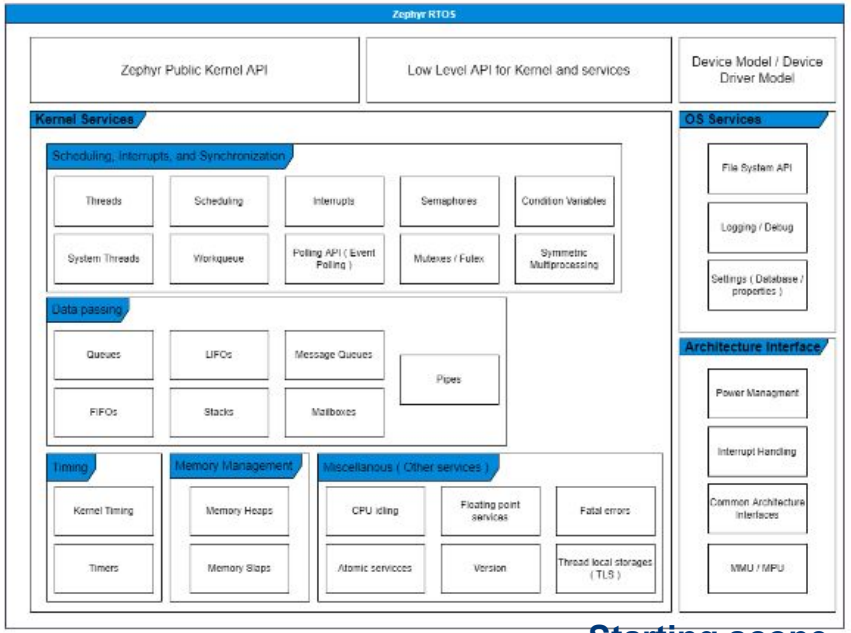
- The V-model for functional safety development originated from systems engineering practices to structure the process of designing and validating complex systems.
- It was later adapted and widely adopted in the automotive industry and other safety-critical sectors as a framework for ensuring the systematic 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

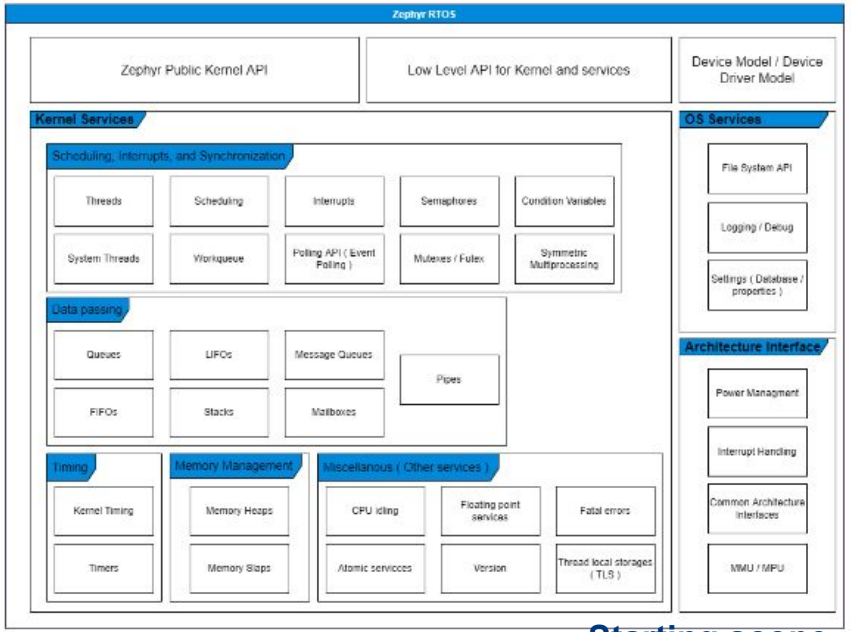


Starting scope

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

Zephyr Initial Certification Focus

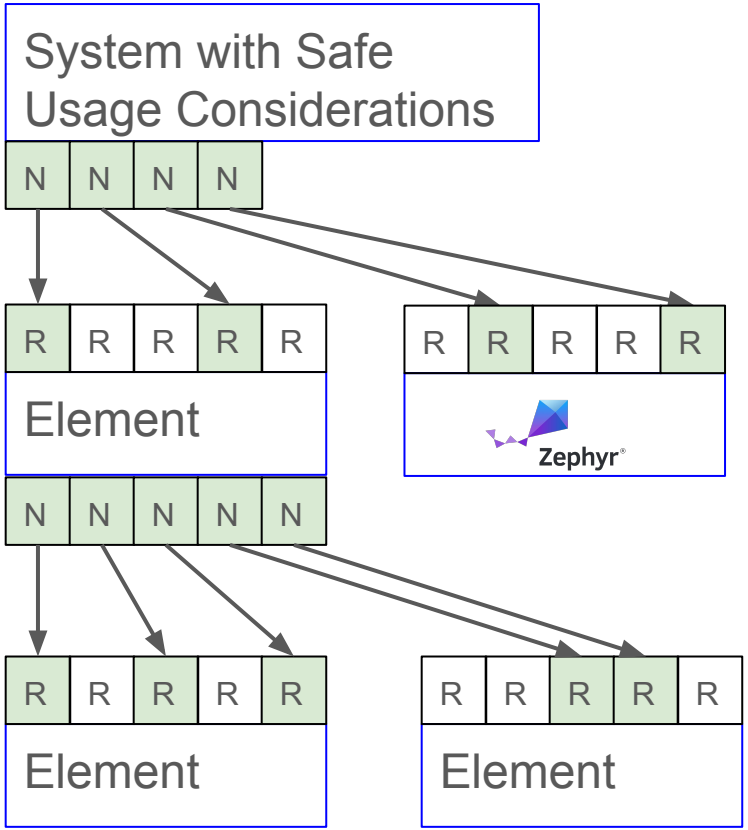
- Start with a limited scope of kernel and interfaces
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s)
- Option for 26262 certification has been included in contract with certification authority should there be sufficient member interest



Starting scope

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

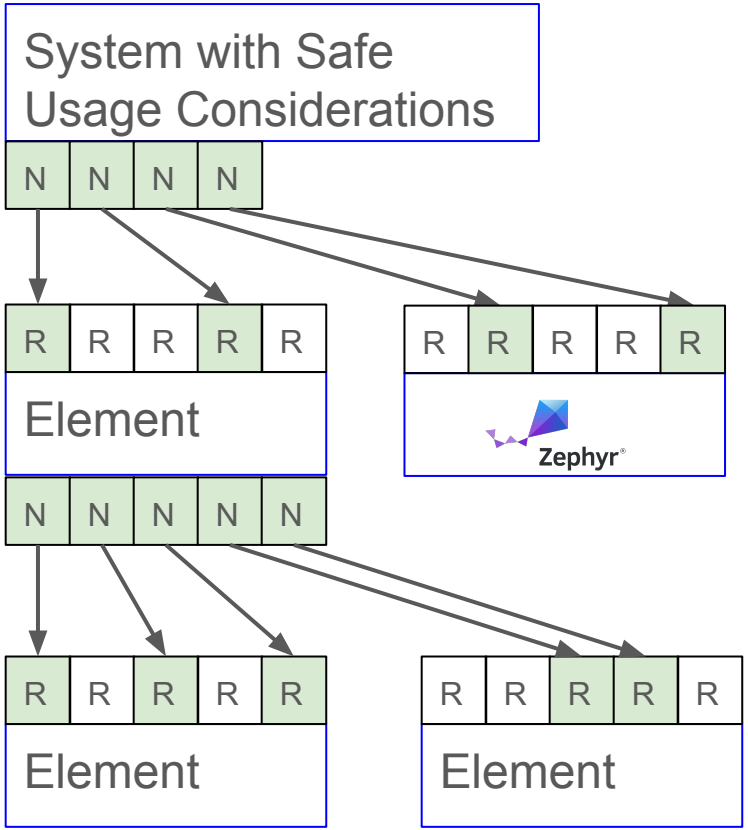


Development and verification independent of a specific context or application

Provides integration and operation information for safe system integration

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

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

Zephyr®

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

Zephyr®

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

Zephyr®

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

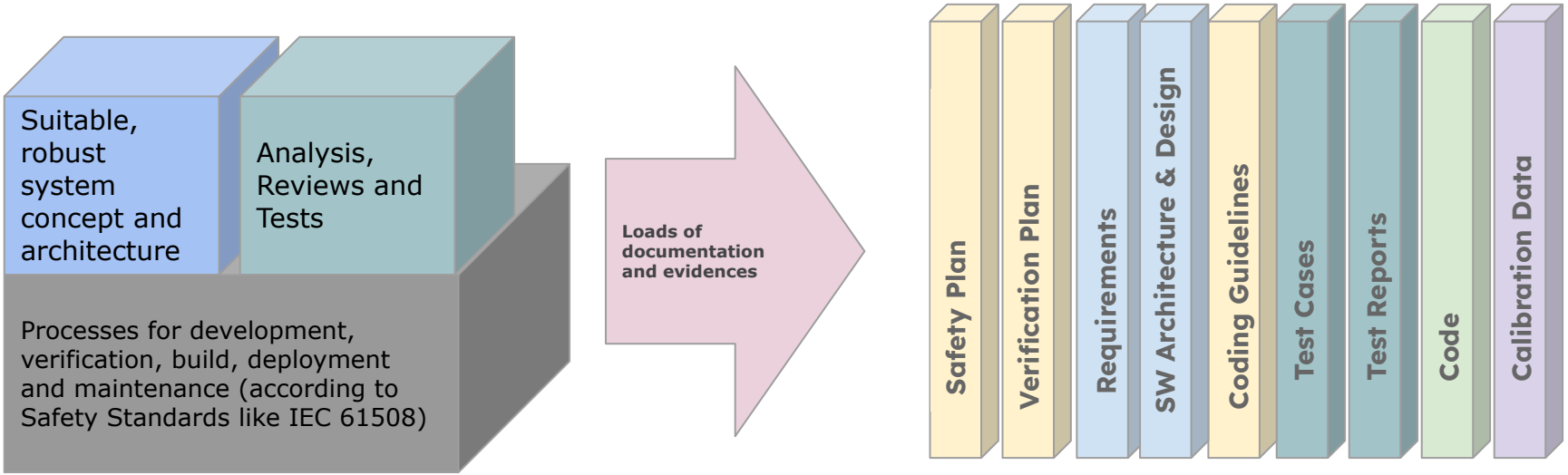
“Where a pre-existing software element is reused to implement all or part of a safety function, the element shall meet both requirements a) and b) below for systematic safety integrity:

b) provides a safety manual [...]

⇒ creation of the Safety Manual

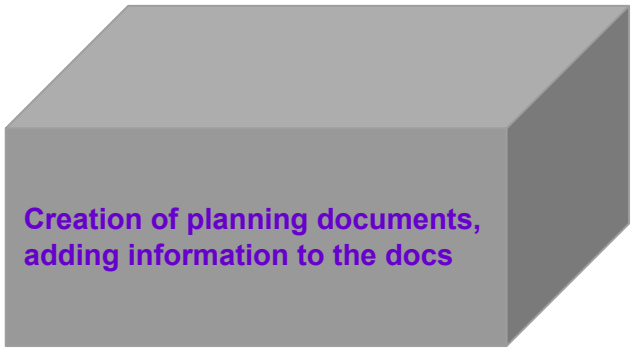
What is Functional Safety aiming for ?

Safety Architecture and Documentation



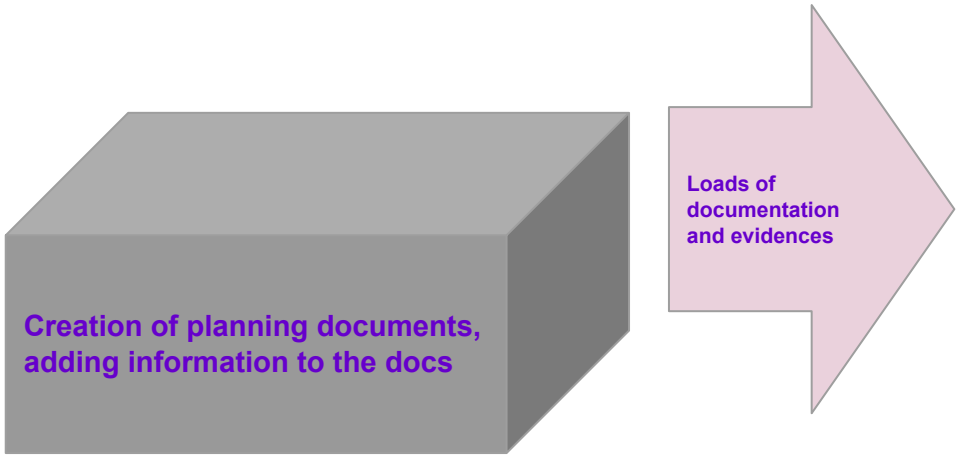
How is the Zephyr Project addressing this?

Following Route 3s by creating the missing work products



How is the Zephyr Project addressing this?

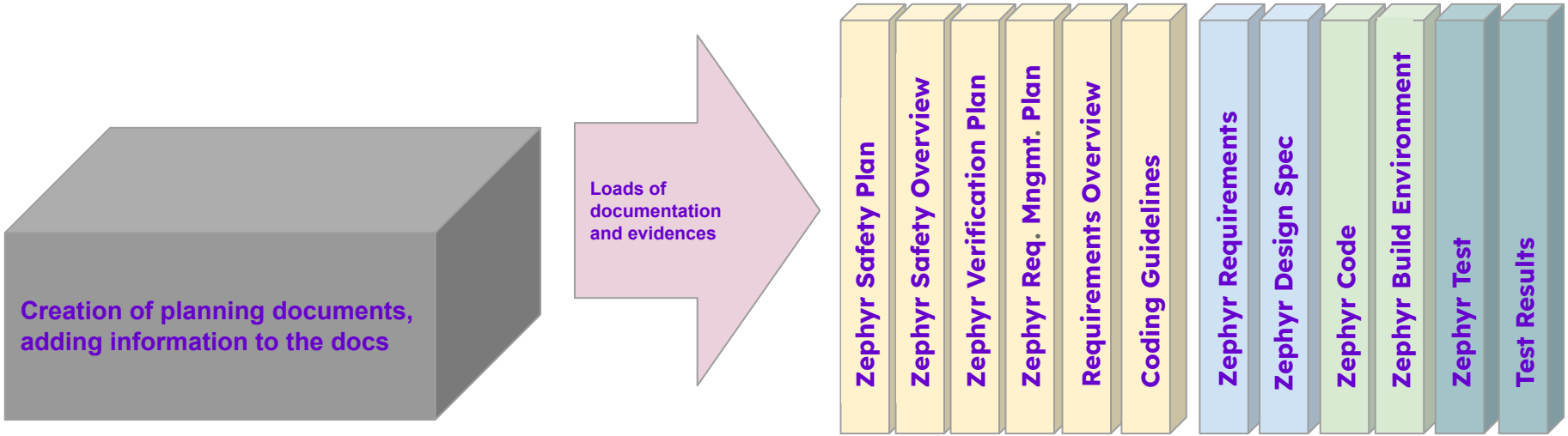
Following Route 3s by creating the missing work products



How is the Zephyr Project addressing this?

Zephyr®

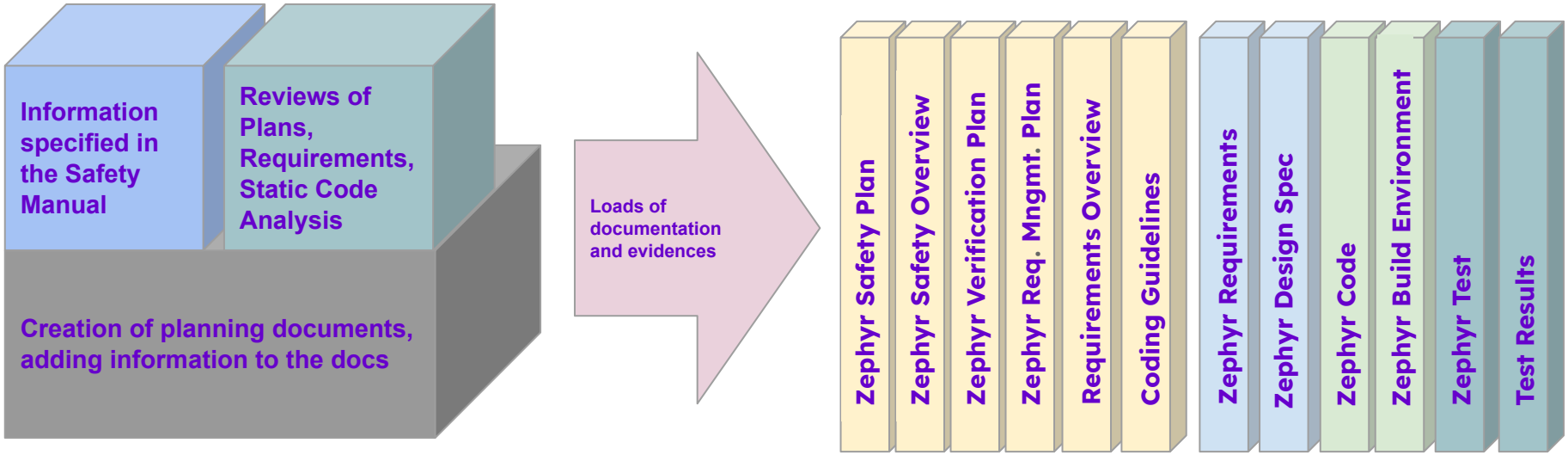
Following Route 3s by creating the missing work products



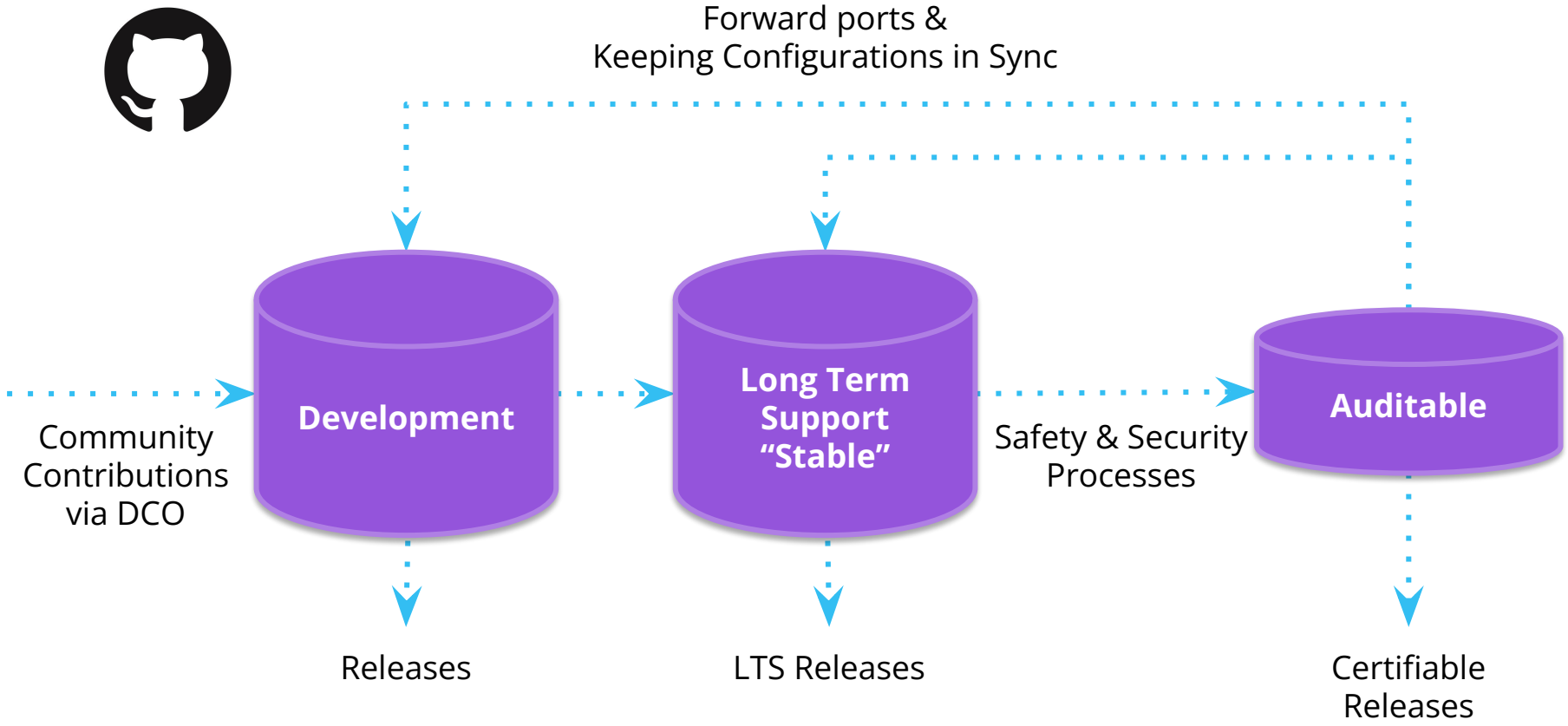
How is the Zephyr Project addressing this?

Zephyr®

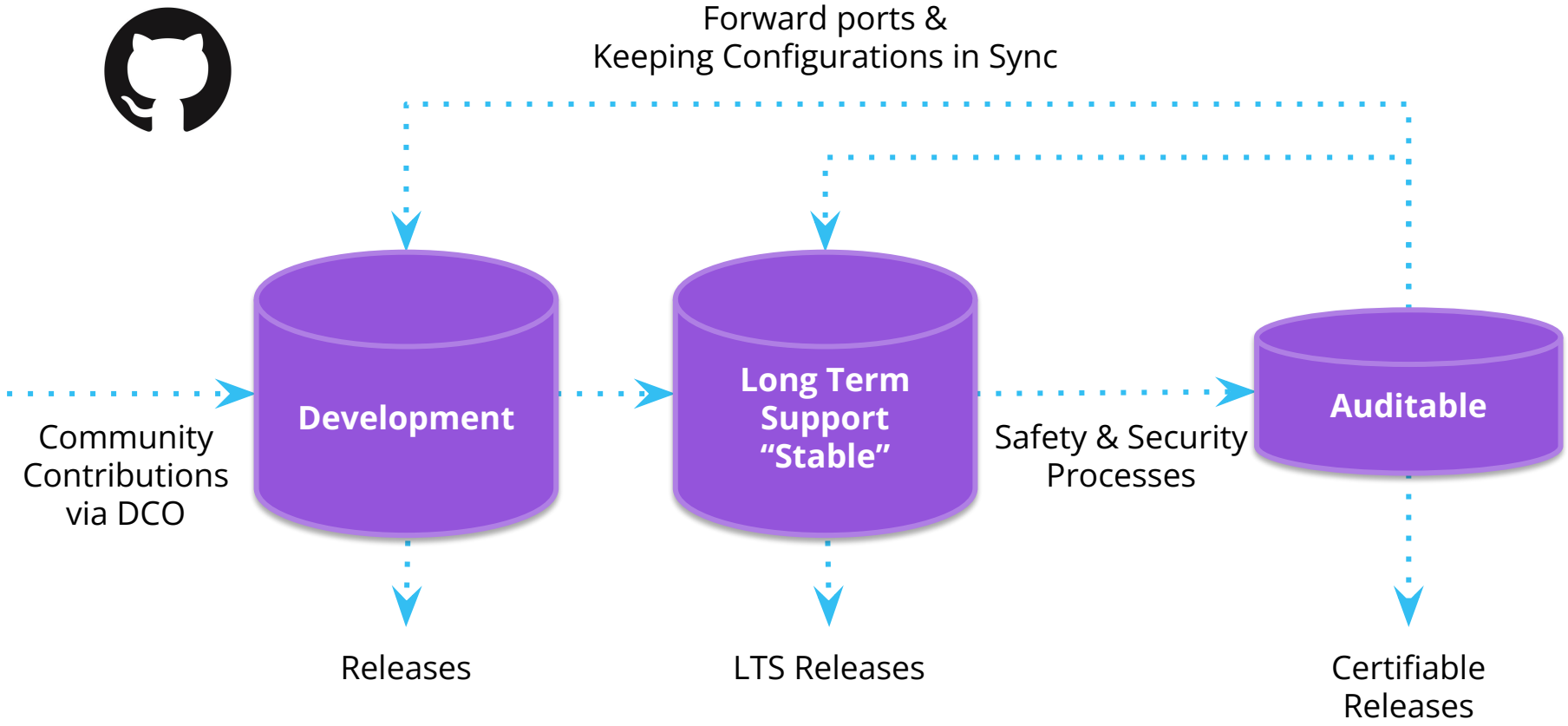
Following Route 3s by creating the missing work products



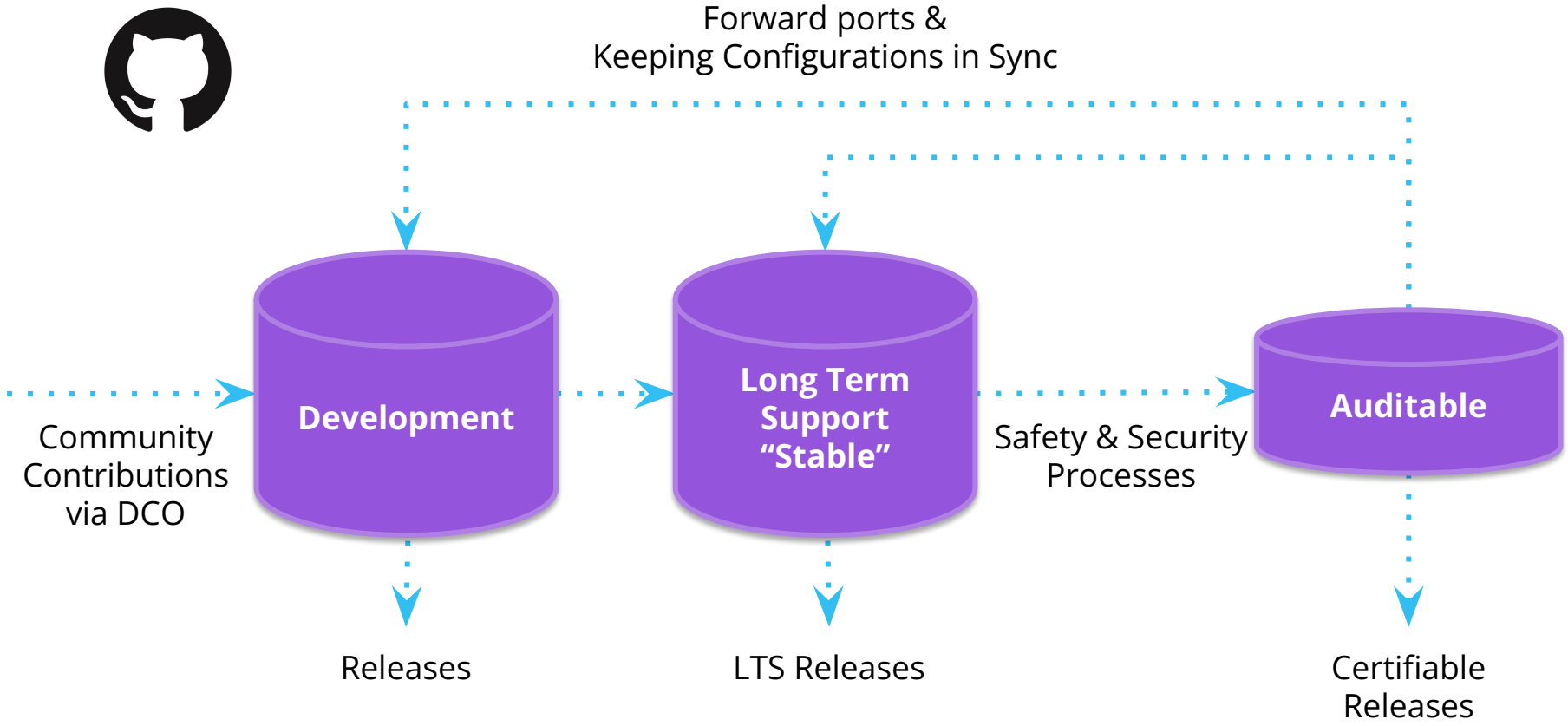
Conceptual Repositories



Introduce Traceability on "Development"

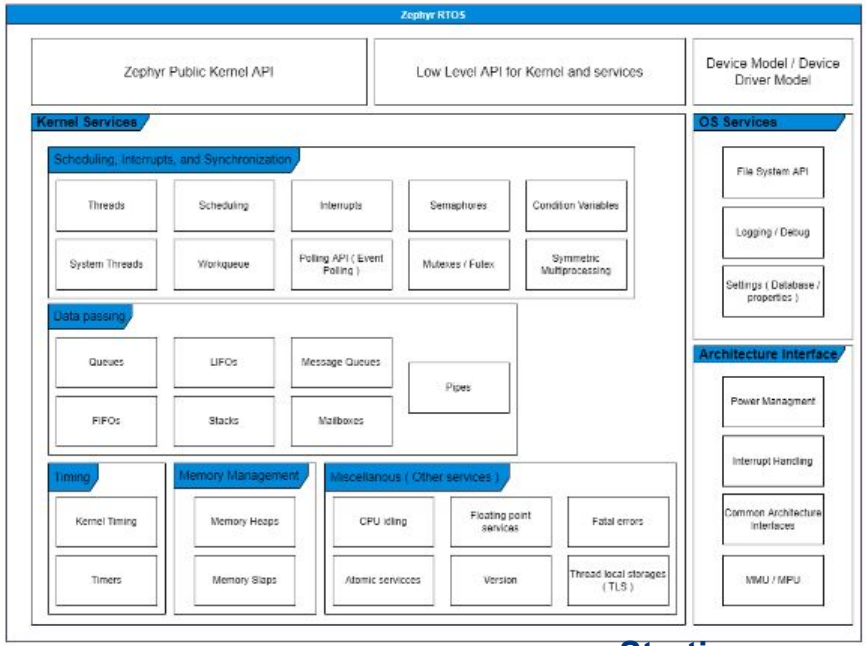


"Auditable" subset of "Development"



Zephyr Initial Certification Focus

- Start with a limited scope of kernel and interfaces - aka "**Auditable**"
- Need to be able to maintain it with security fixes over time - "**Auditable**" set expected to be associated with a specific "**Long Term Support**" release



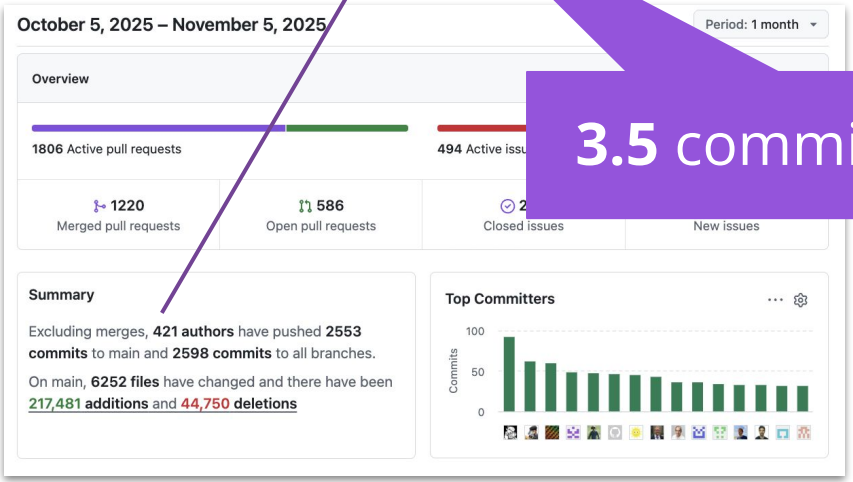
Starting scope

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

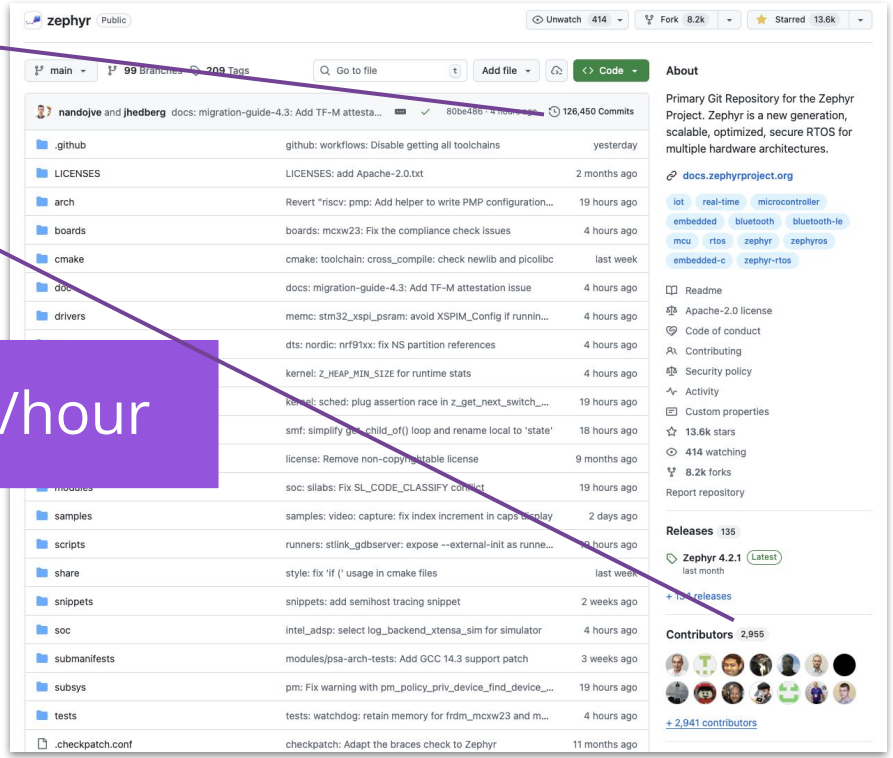
Zephyr Development: Rate of Change

https://github.com/zephyrproject-rtos/zephyr
- Total commits: 126,450
- Total contributors: 2,955

https://github.com/zephyrproject-rtos/zephyr/pulse/monthly
- Monthly contributors: 421
- Monthly commits: 2,598



3.5 commits/hour



Lessons Learned from First Attempt

- **All changes and enhancements need to go to Upstream Development**
  - Initial attempt at alignment with Coding Guidelines (MISRA, etc.) was on an LTS.
  - Redoing each release wasn't practical, so recognized need to be upstream.
- **Need buy in from Upstream Maintainers on Methodology & Processes**
  - TSC, Process and Testing Workgroups are key stakeholders and need to participate to shape a practical path that the project can adhere to.
- **Need Open Community Participating to Scale**
  - Committee sets scope to apply funding to
  - Others should be able to apply same methodology on different scope (and hopefully contribute back to project upstream)

Open Up Creating the Work Products

Zephyr®

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**

Work Product Strategy

Zephyr®

Strategy for artefact creation:

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

New artefacts ⇒ StrictDoc

- Requirements
- Safety Plan
- Safety Manual
- …

Evidence ⇒ StrictDoc

- Assessments
- Checklists
- …

Enhancements to project documentation

- Sphinx in docs.zephyrproject.org

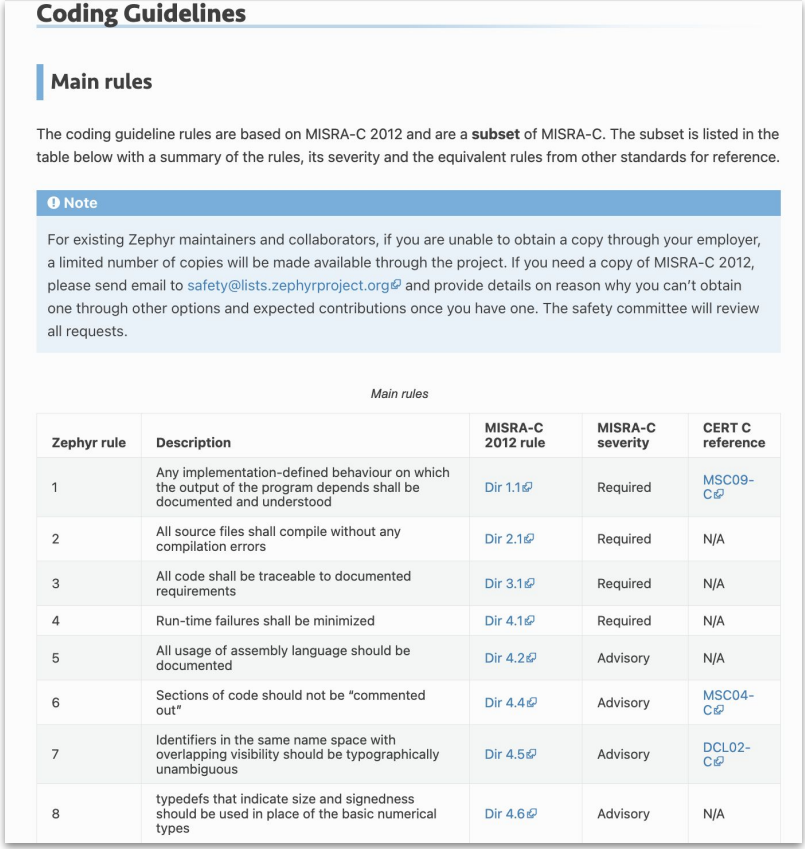
Compliance with Coding Standards

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

Static Analysis in the CI to check for adherence on each patch integrated into workflow via commercial tool "ECLAIR" from Bugseng



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

Lessons Learned in Current Path

- Work with those interested in collaborating with you.
  - **Open Source**:
    - StrictDoc maintainer engaged in evolving tool to work with our needs. Working regularly with Safety Manager and Safety Architect on including project needs into tool requirements.
    - SPDX community willing to evolve specification for required metadata for continuous compliance via Safety Profile
  - **Commercial**:
    - Bugseng willing to prototype with us that tool could be integrated into workflow via trial licenses, before project signed contract in late 2025. Safety Architect & TSC Steering Committee Chair active in proving prototype.

Systematic Capability for Safety

To achieve Route 3s for an SEooC:

- Create and establish enough evidence for planning, process and adherence to defined techniques (Functional Safety Management: plans, guidelines, tooling application)
- Create accompanying documentation to enable usage of the safety scope (Safety Manual)
- Provide code that is
  - Compliant with the coding guidelines of the project
  - Described by requirements, architecture and design
  - Sufficiently tested
- Achieve a 3rd party certification for IEC 61508 SIL 3

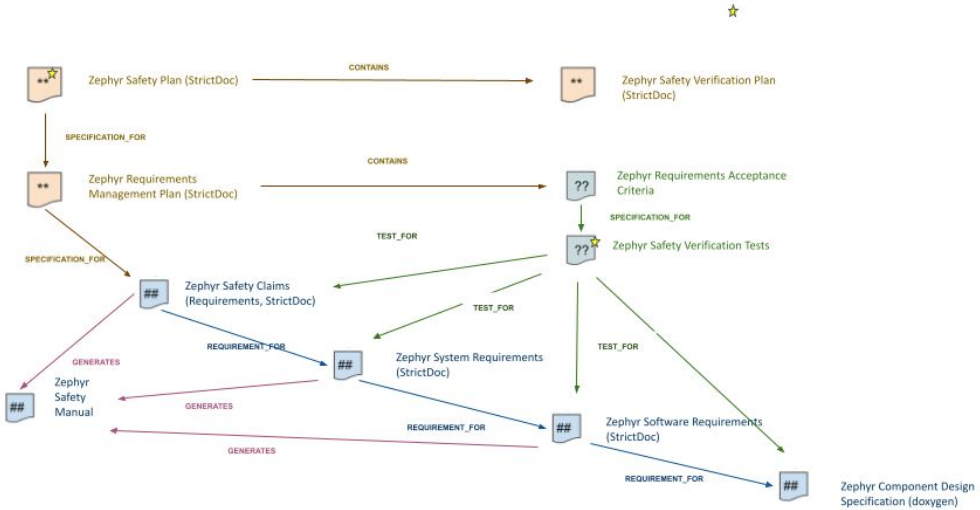
Zephyr Approach for route 3s:

- Creation of plans and process documentation
- Define the safety scope
- Definition of safety claims, traceability to requirements, code and tests
- Reusing as much of the existing documentation as possible
- Use existing tests, create traceability, analysis and close coverage gaps
- Create a Safety Manual

Zephyr Work Product Structure

Principles of creating the documentation:

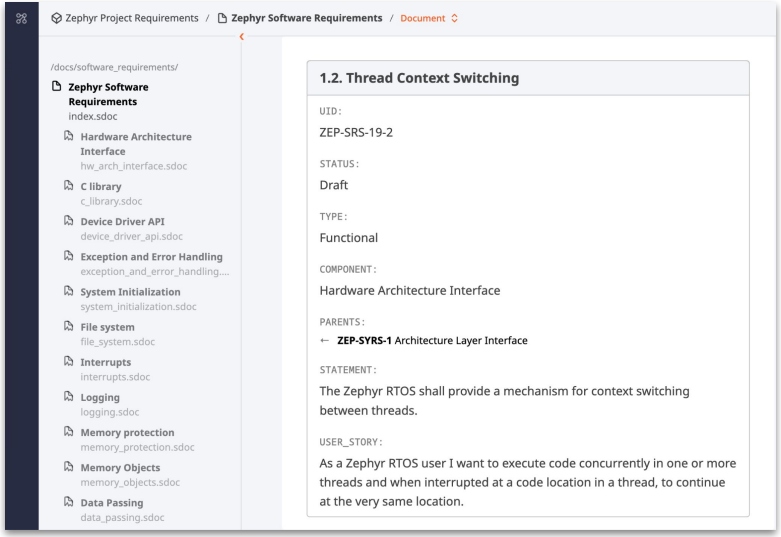
- Use developer friendly tooling
- Use known workflows on GitHub
- Reuse as much as we can from the docs



- New documents like Safety Plan, Safety Manual, Requirements: StrictDoc
- Enhancement of the community documentation in the Docs: Sphinx
- Assessment evidences & checklist: StrictDoc

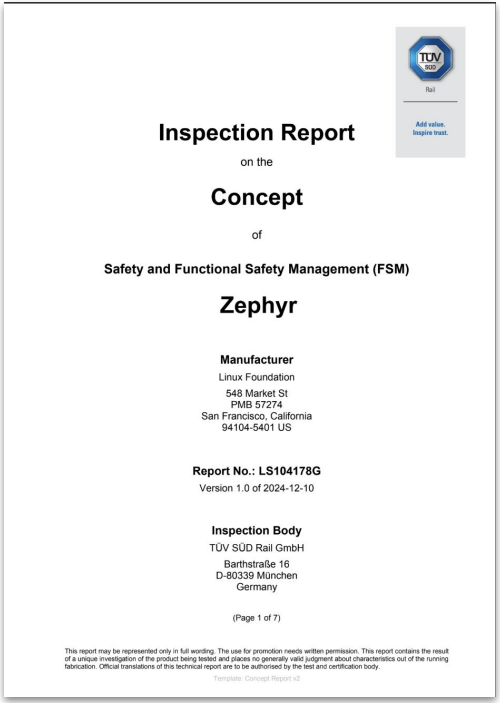
Status Today

- Coding Guidelines established based on MISRA rules and applied
- Static Analysis tooling to check for adherence to Guidelines selected and being deployed for future contributions
- Reference Requirements and Traceability started in the code base. Working on visible end-to-end example
- 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

Safety Assessment & Certification Plan

Zephyr®

Phase 1 - Concept Phase

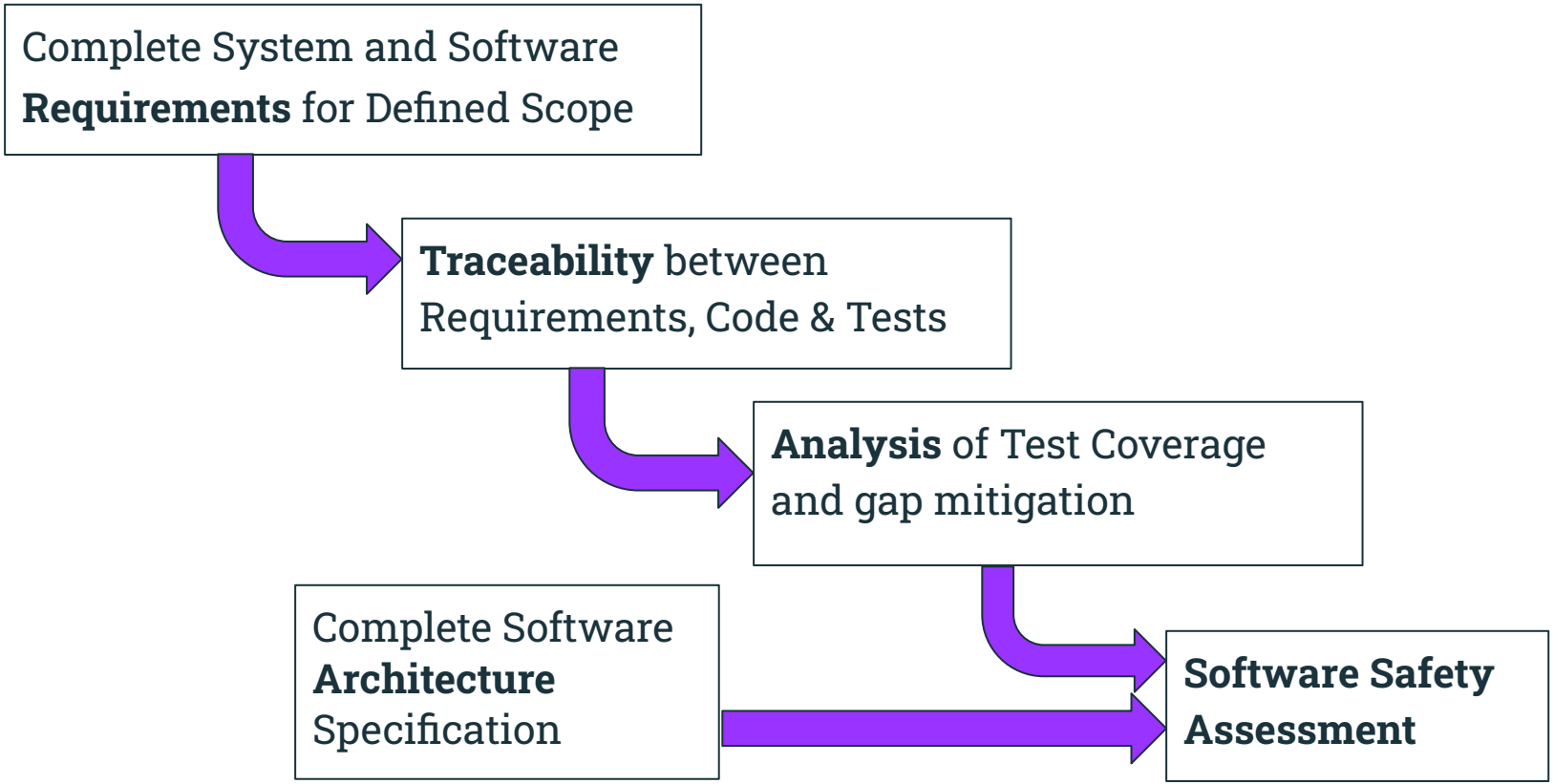
Approval of the overall plans and strategy, scope and high level requirements & architecture

Phase 2 - Detailed Phase

Filling the gaps:

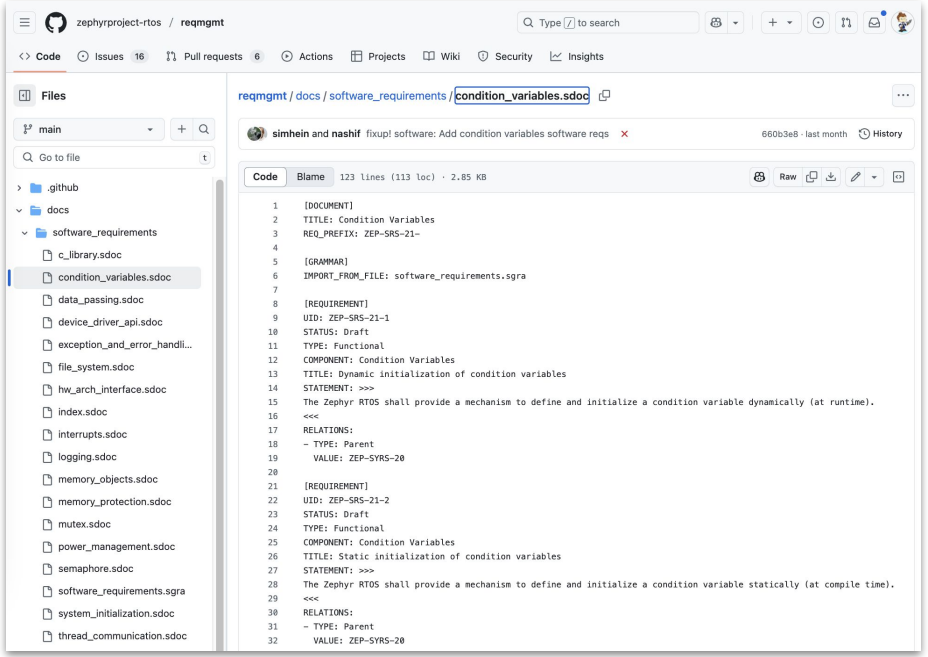
- Completeness of requirements
- Safety Analysis
- Traceability
- Verification & Test Coverage

Critical Path to Phase 2



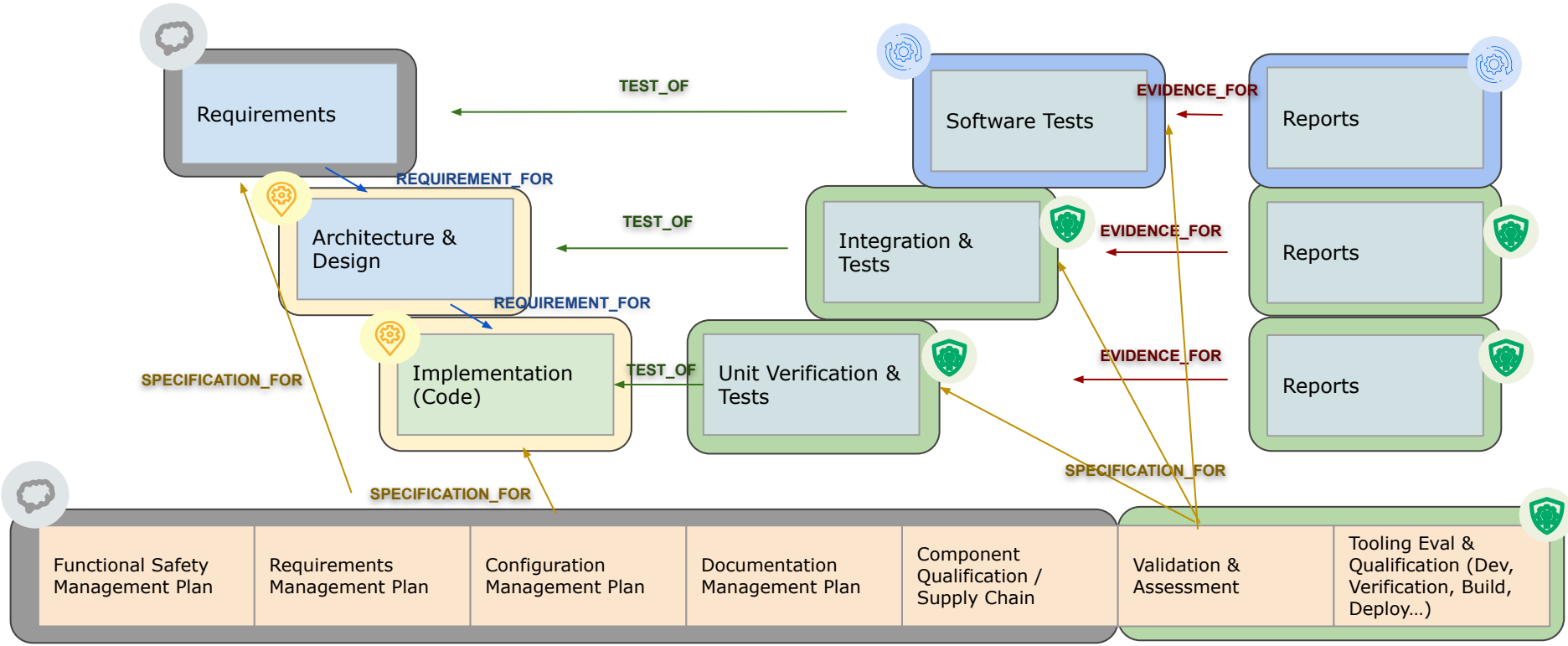
Critical Path: Requirement & Traceability

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

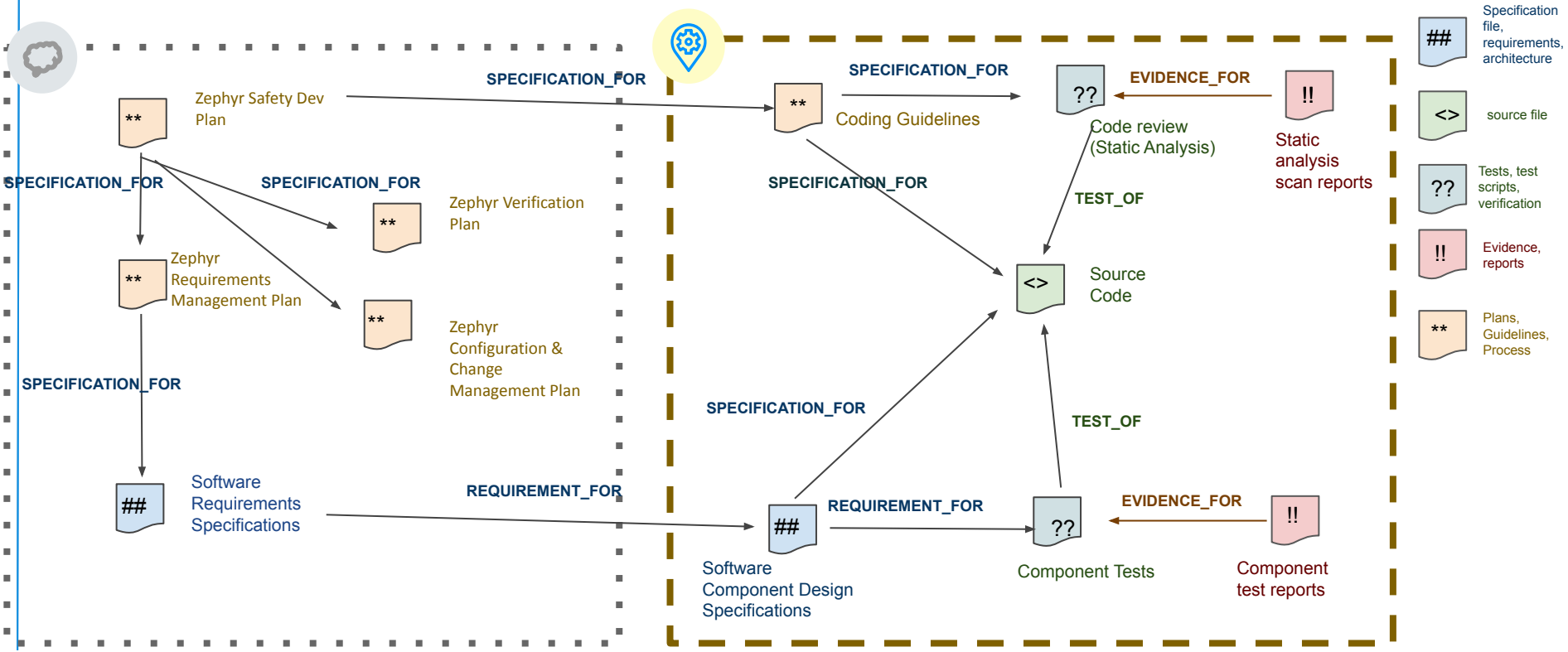


Why do we need requirements and traceability between requirements to the code and tests?

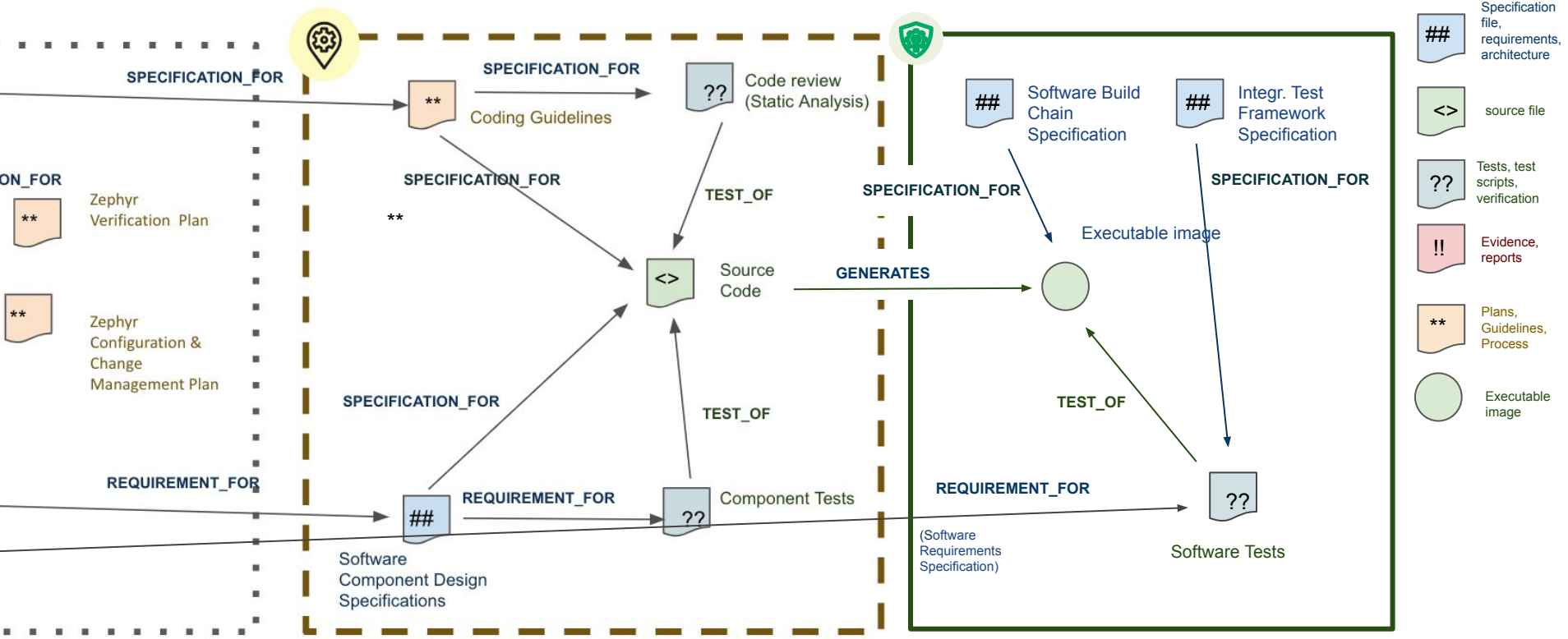
SPDX Style Dependencies in a FuSa Project



Knowledge Graph of Design to Source

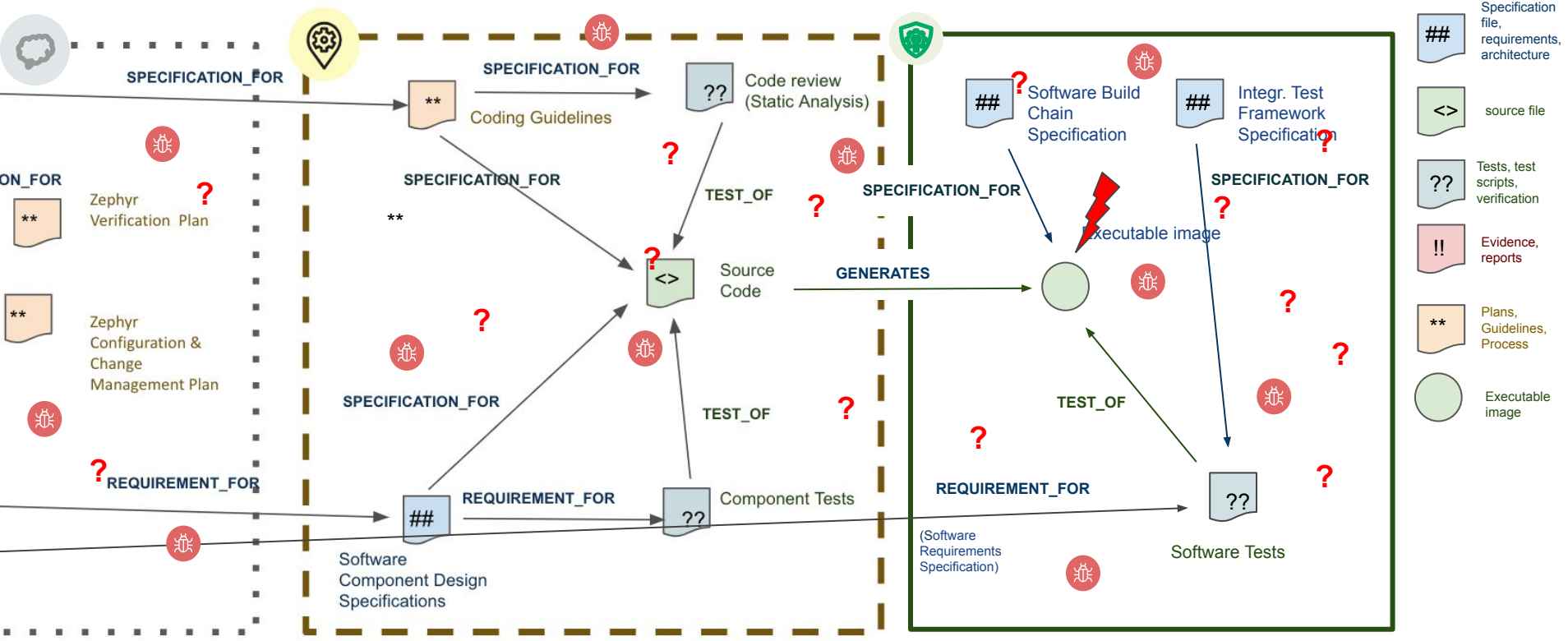


Knowledge Graph of Source to Build



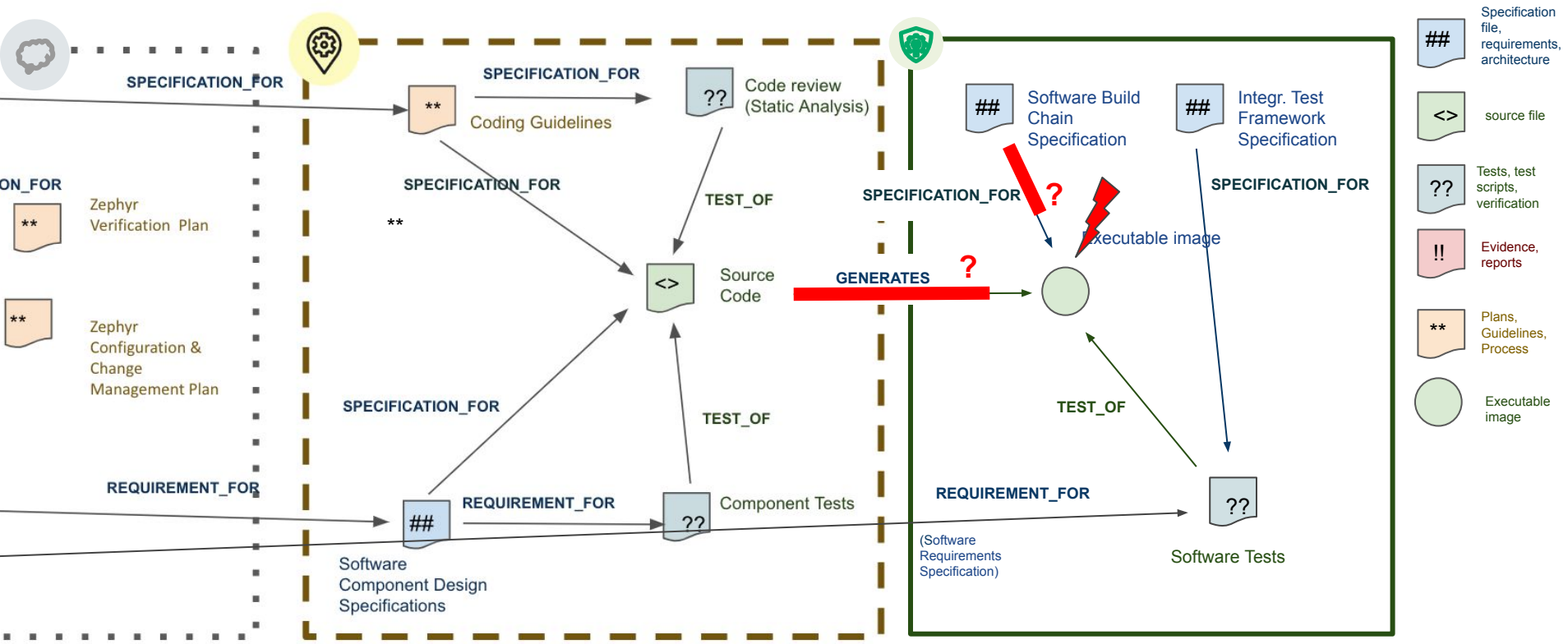
Dependency Identification between Components

SPDX SAFETY

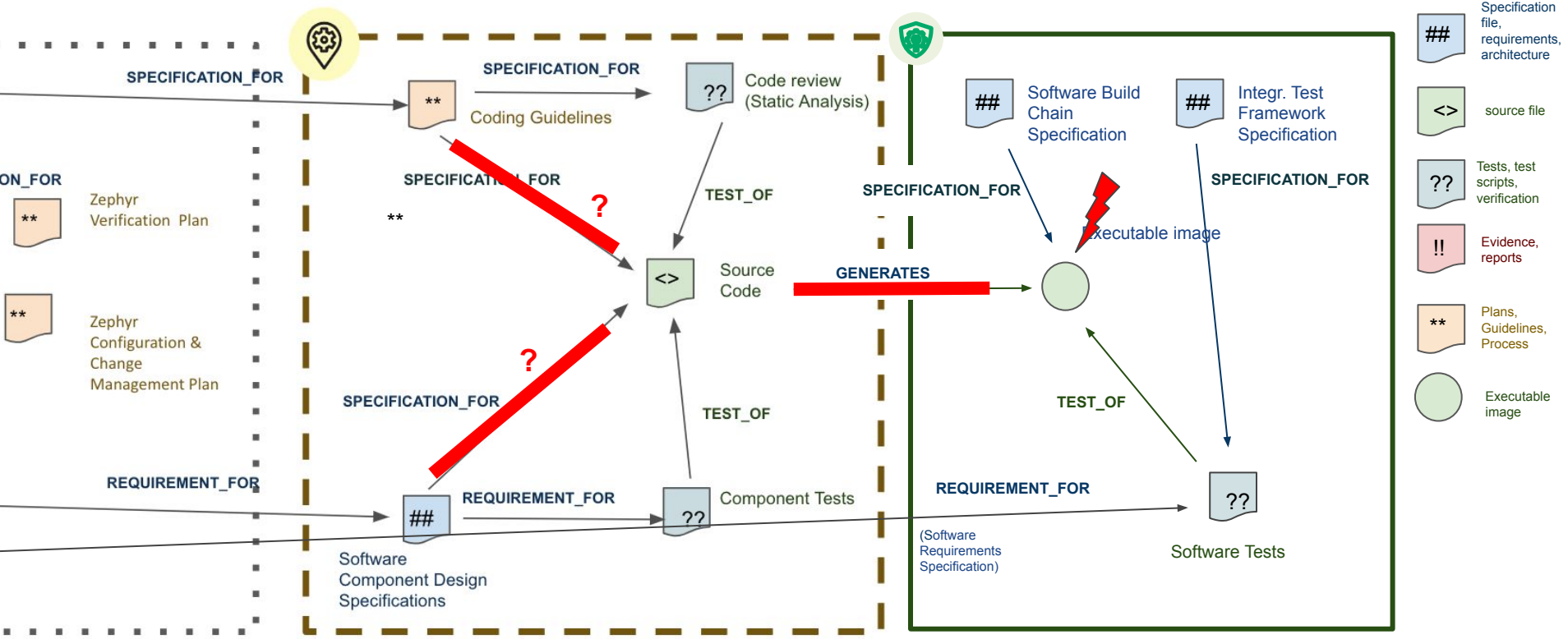


SPDX

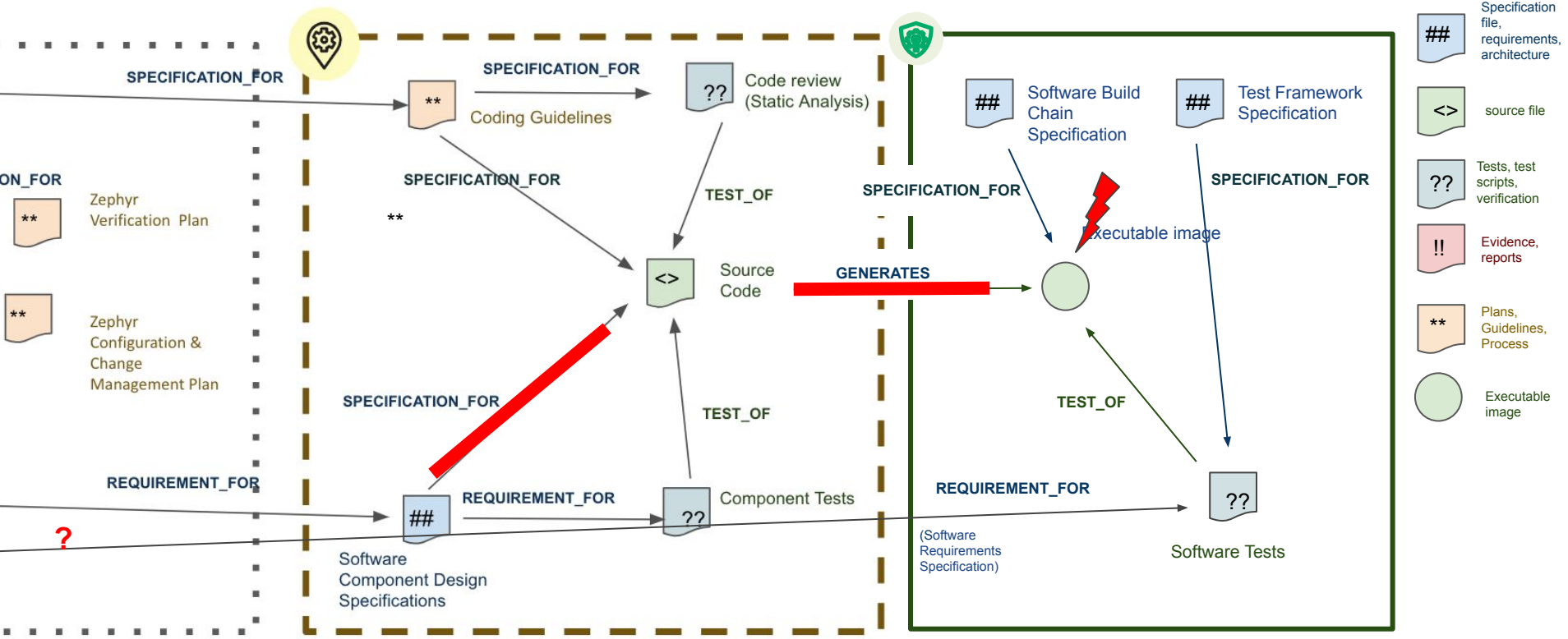
Dependency Identification at Component Level



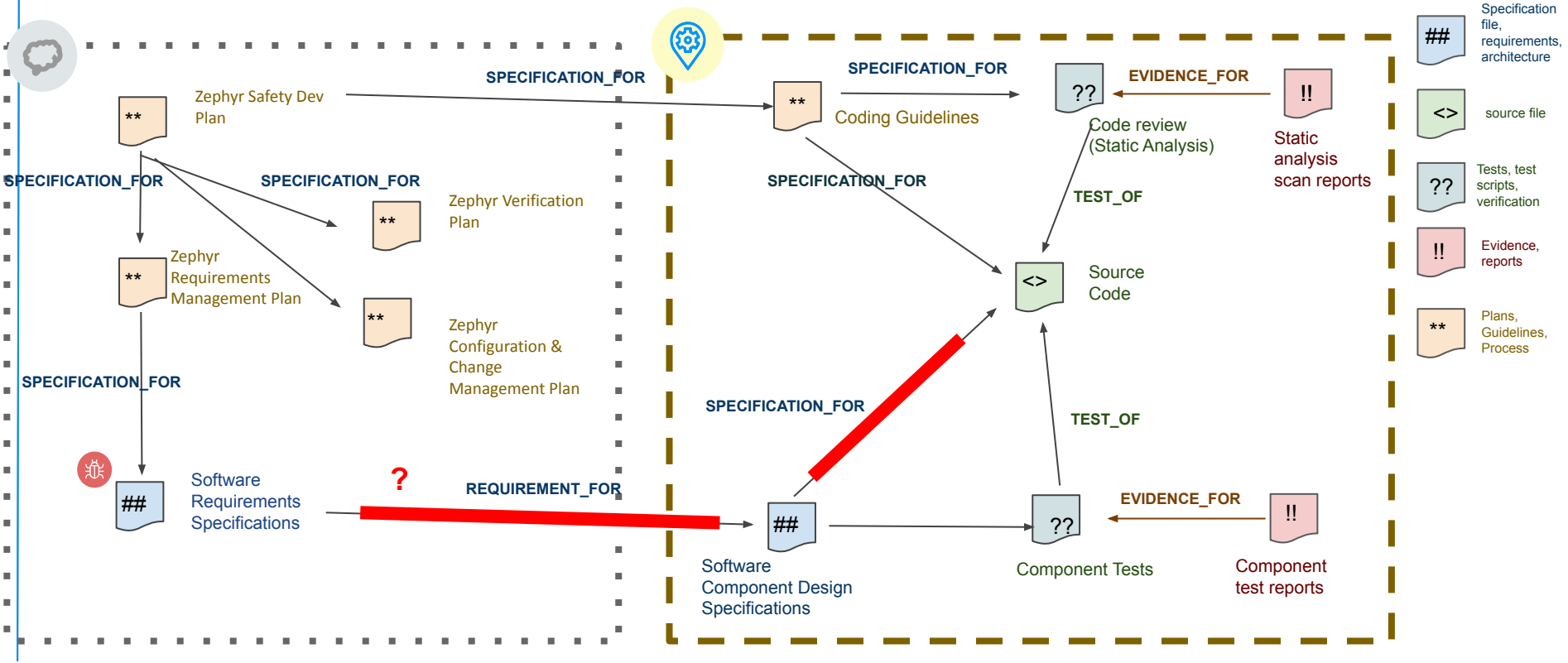
Dependency Identification at Component Level



Dependency Identification at Component Level

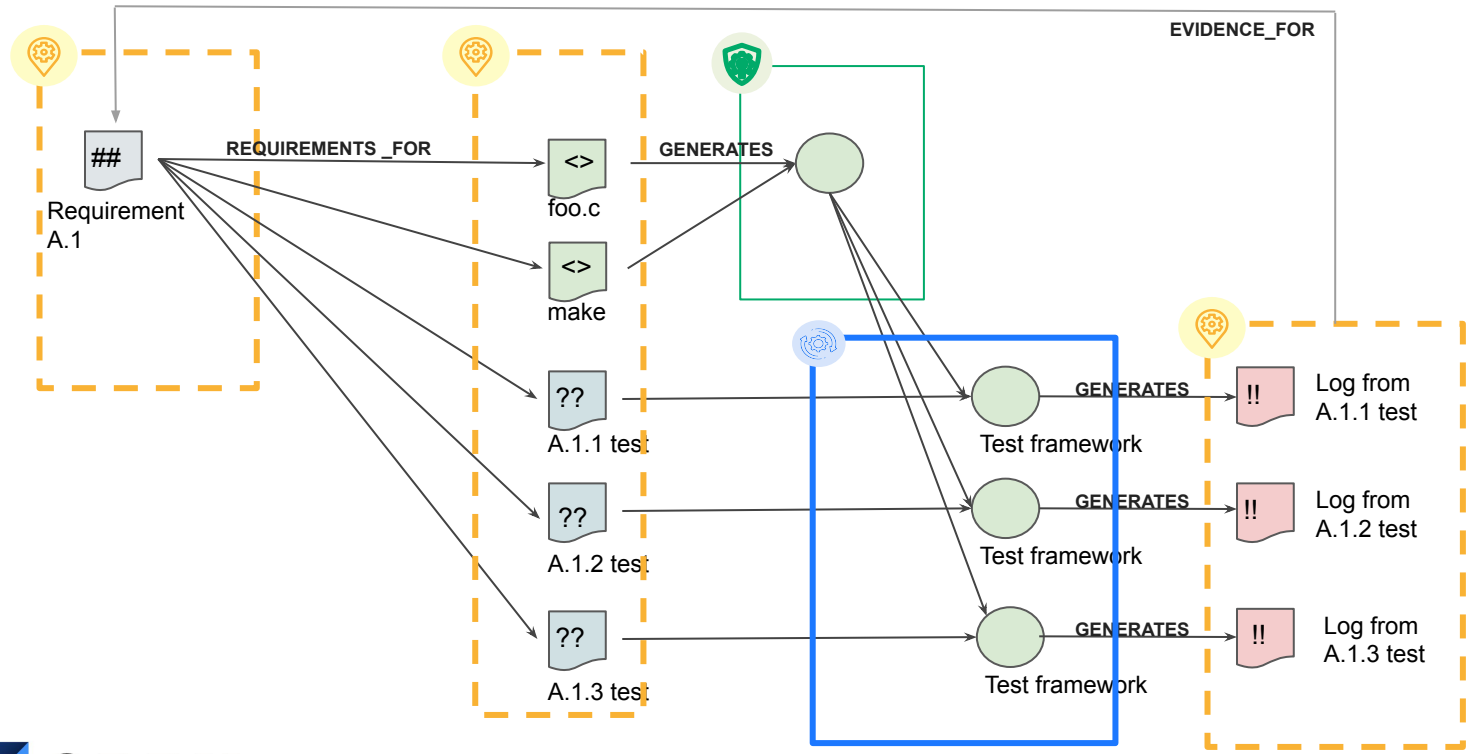


Dependency Identification at Component Level



When needed: Traceability Inside Component

Requirement to Code to Tests to Evidence + detailed SBOMs ⇒ Automation of Analysis



Specification file, requirements, architecture

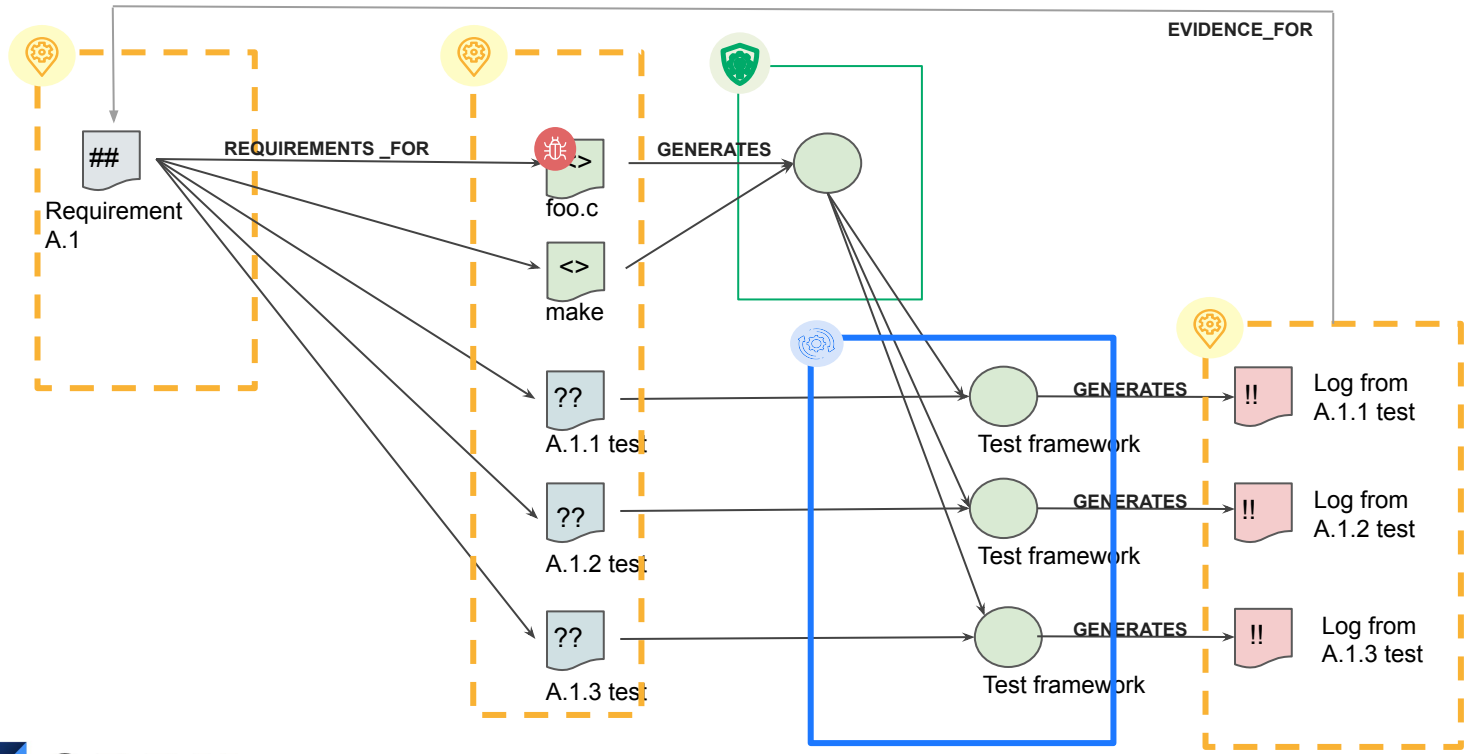
source file

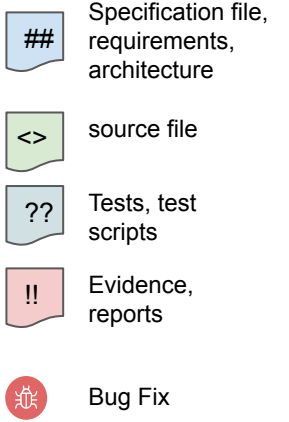
Tests, test scripts

Evidence, reports

When needed: Traceability Inside Component

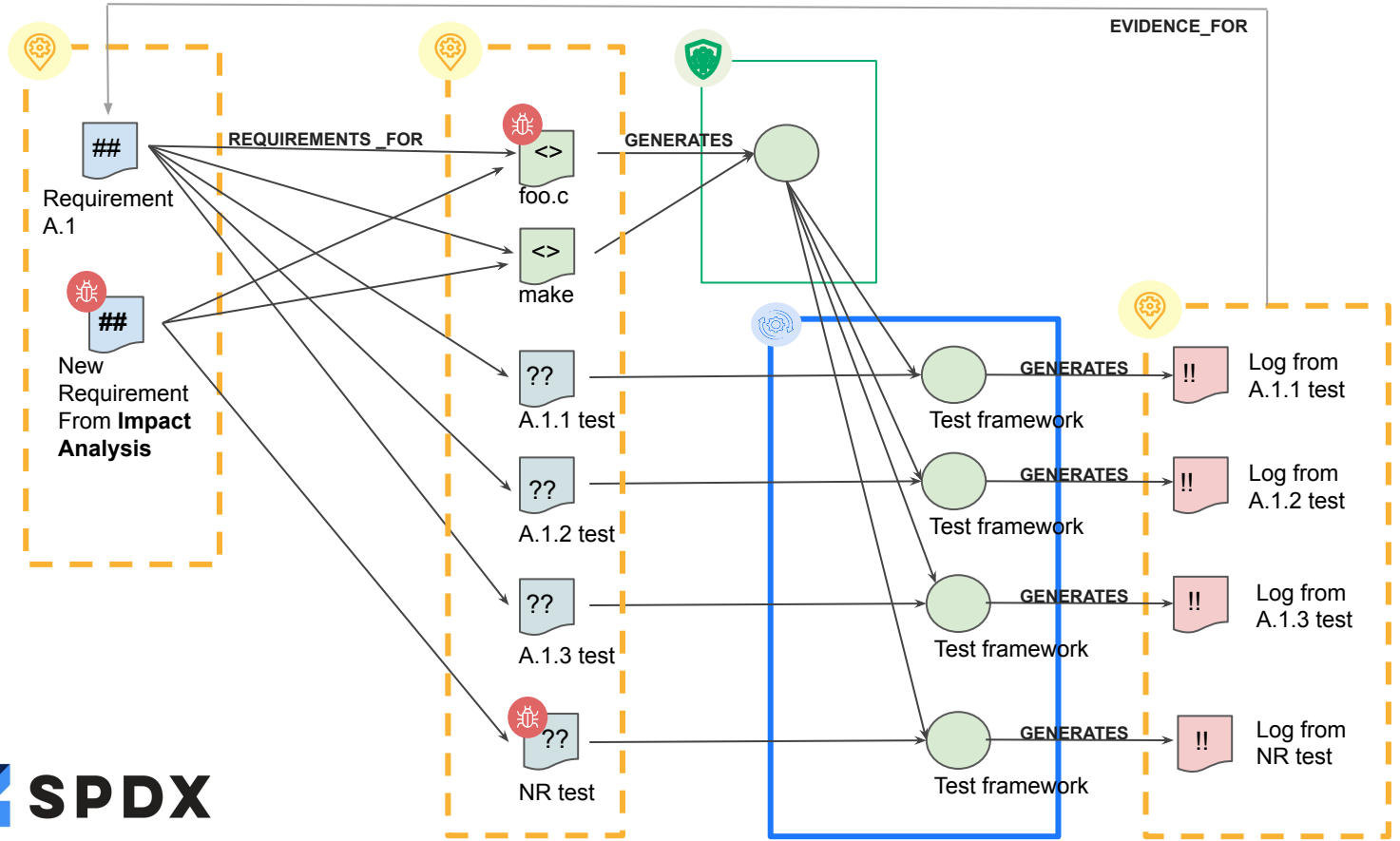
Requirement to Code to Tests to Evidence ⇒ Continuous Compliance





Traceability Inside Component

Identification of New Requirement to Code to Tests to Evidence



Specification file, requirements, architecture

source file

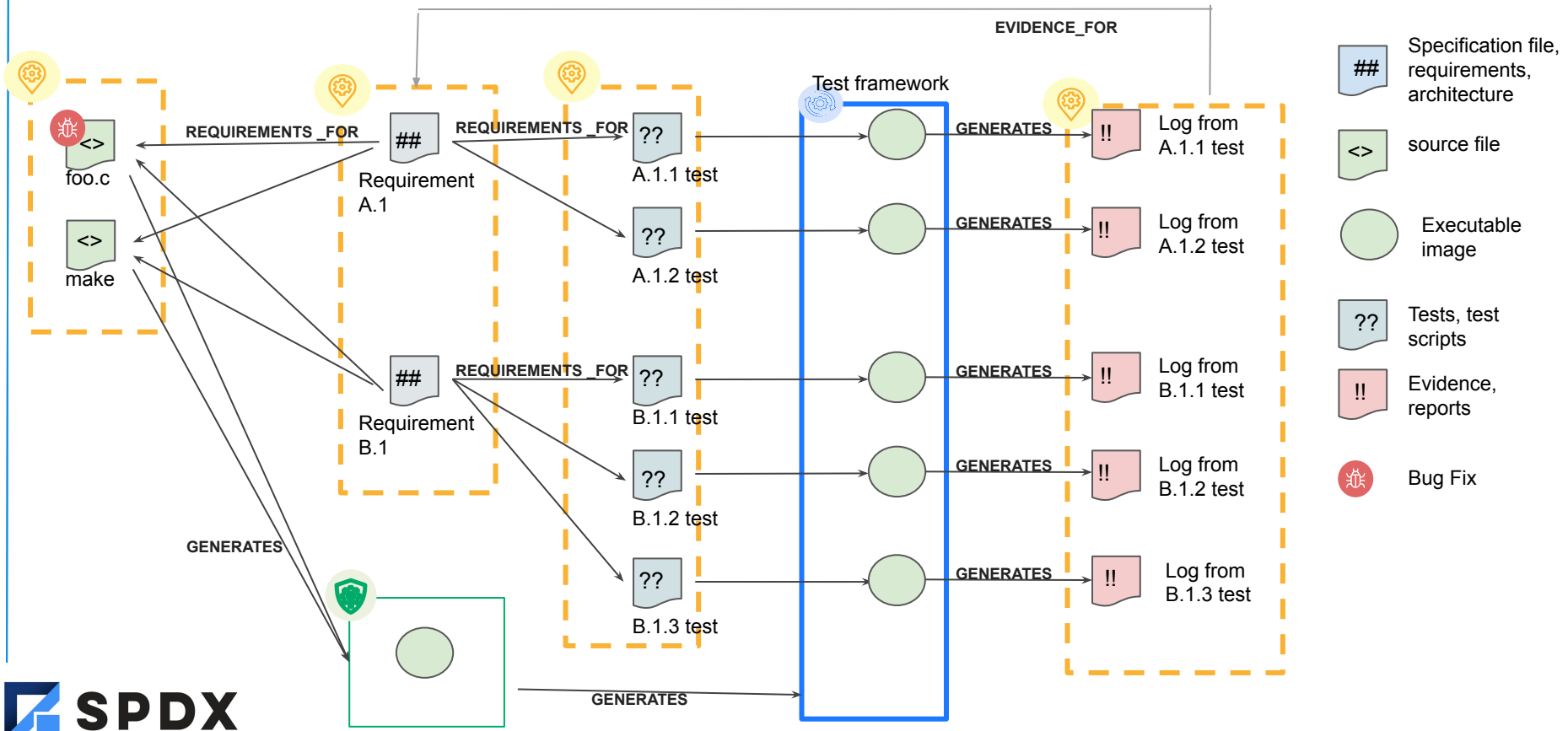
Tests, test scripts

Evidence, reports

Bug Fix

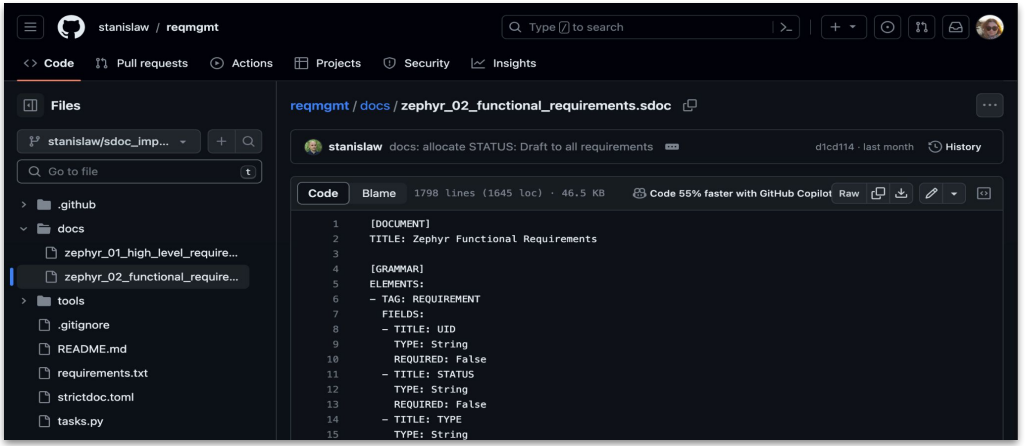
Inside Component: Traceability of Source to Requirements

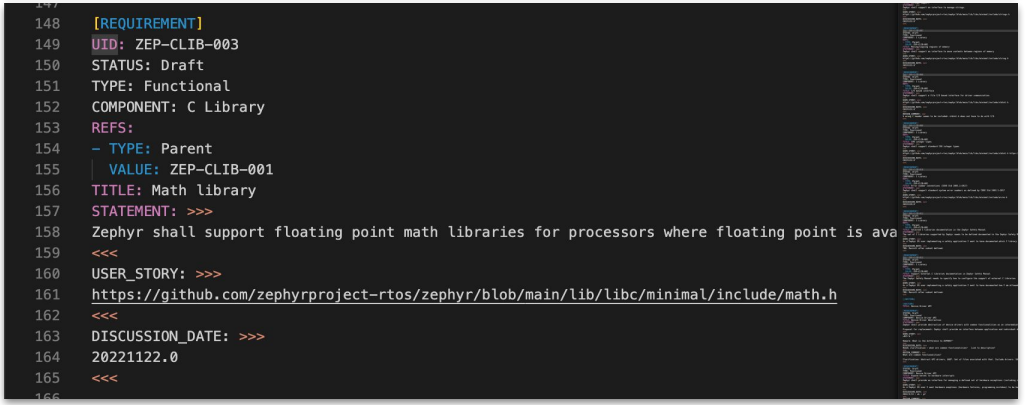
Efficient Regression Testing ⇒ Evidence of Compliance



Current Requirements Work In Progress

- Used tooling: StrictDoc (https://github.com/strictdoc-project/strictdoc)
- Hierarchical structure of requirements that works for the project
- Capturing the requirements in StrictDoc which is working towards import/export of SPDX





!

Also plans, like the Zephyr Safety Plan look like that, each planning item is tracked as a requirement

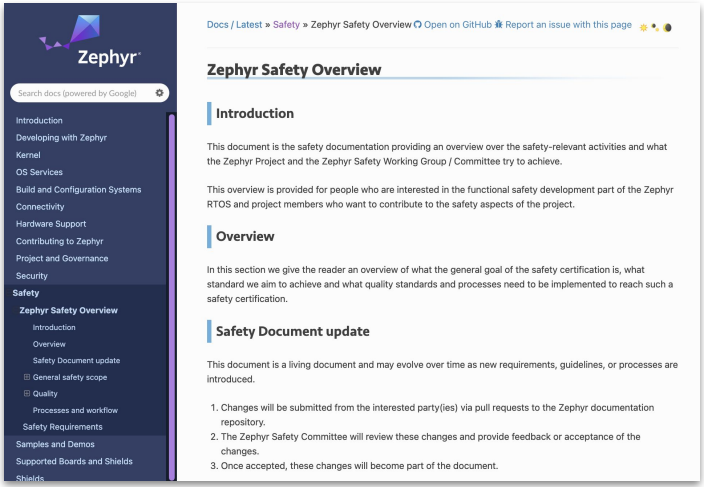
Assessment checklist -> each checkpoint is a requirement, tracing to the Zephyr's evidences


ARE WE
THERE YET?

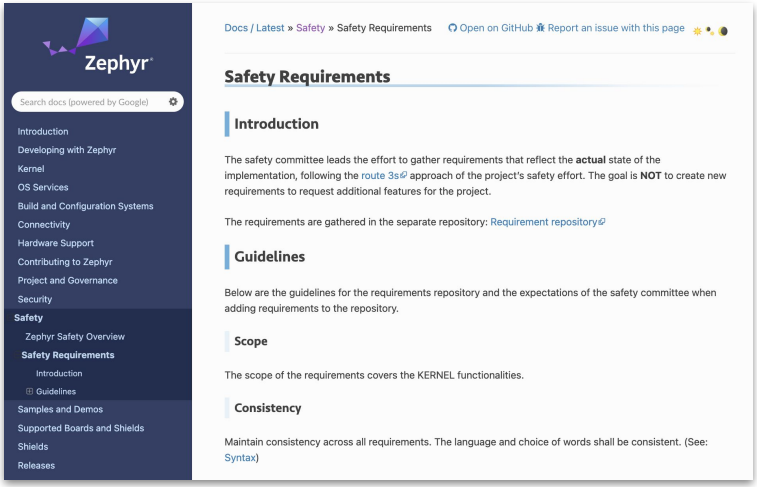
Want to help make it happen faster?

Step 1: Read the docs to start …

Safety Overview



Requirements Guideline



Want to help make it happen faster?

Zephyr®

Step 2: look at our repos …

Requirements:

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

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

Safety Working Group Project:

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

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

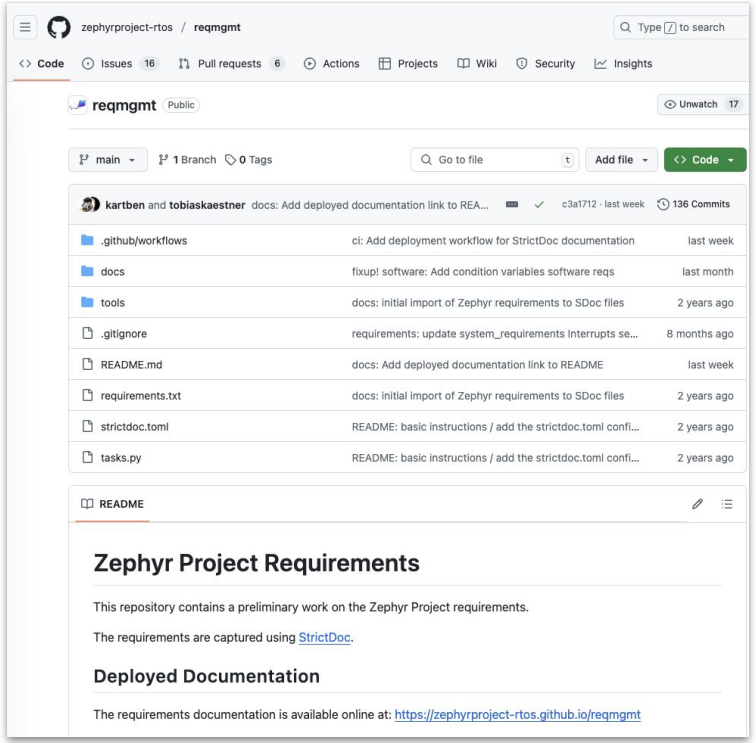
Want to help make it happen faster?

Zephyr®

Step 3: Contribute in our repos:

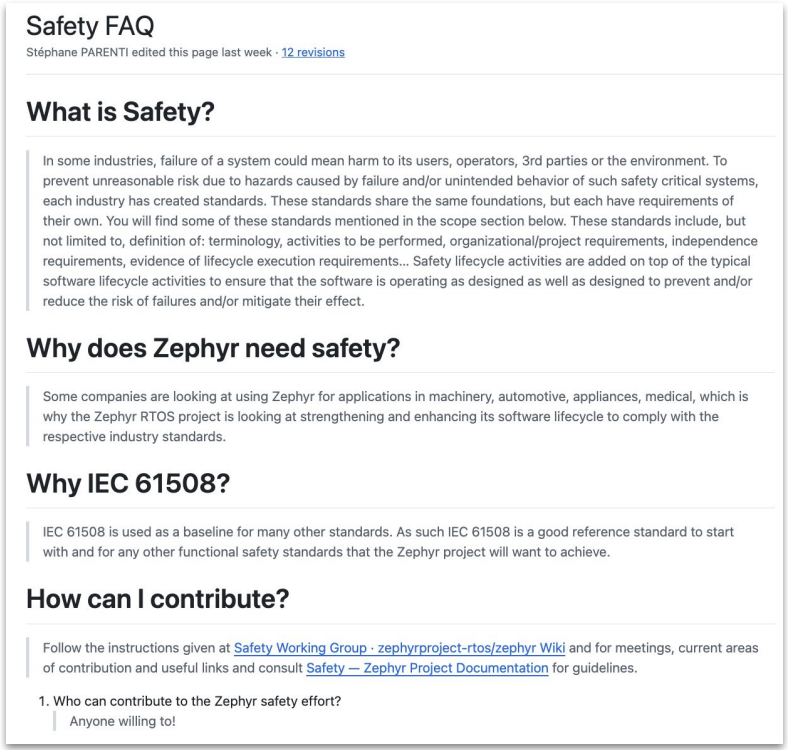
Requirements Repository

- 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

Overall Updates @ Zephyr Safety FAQ



Safety FAQ

Stéphane PARENTI edited this page last week · 12 revisions

What is Safety?

> In some industries, failure of a system could mean harm to its users, operators, 3rd parties or the environment. To prevent unreasonable risk due to hazards caused by failure and/or unintended behavior of such safety critical systems, each industry has created standards. These standards share the same foundations, but each have requirements of their own. You will find some of these standards mentioned in the scope section below. These standards include, but not limited to, definition of: terminology, activities to be performed, organizational/project requirements, independence requirements, evidence of lifecycle execution requirements... Safety lifecycle activities are added on top of the typical software lifecycle activities to ensure that the software is operating as designed as well as designed to prevent and/or reduce the risk of failures and/or mitigate their effect.

Why does Zephyr need safety?

> Some companies are looking at using Zephyr for applications in machinery, automotive, appliances, medical, which is why the Zephyr RTOS project is looking at strengthening and enhancing its software lifecycle to comply with the respective industry standards.

Why IEC 61508?

> IEC 61508 is used as a baseline for many other standards. As such IEC 61508 is a good reference standard to start with and for any other functional safety standards that the Zephyr project will want to achieve.

How can I contribute?

> Follow the instructions given at Safety Working Group · zephyrproject-rtos/zephyr Wiki and for meetings, current areas of contribution and useful links and consult Safety — Zephyr Project Documentation for guidelines.

1. Who can contribute to the Zephyr safety effort?
   > Anyone willing to!

https://github.com/zephyrproject-rtos/zephyr/wiki/Safety-FAQ

Join us!

Zephyr®

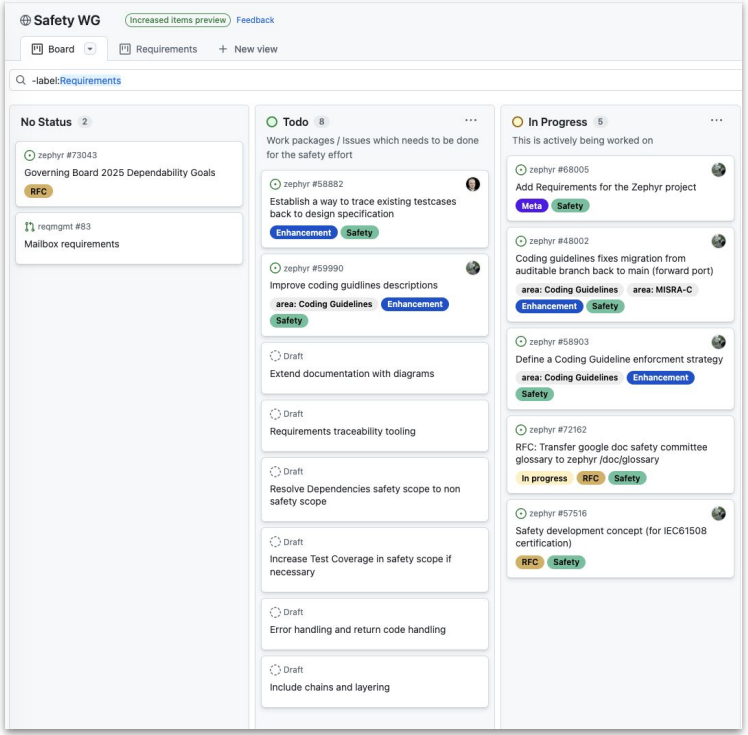
Participate in the Working Group:

[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

Thank you for your interest!

THE LINUX FOUNDATION OPEN SOURCE SUMMIT JAPAN

AI_dev Open Source GenAI & ML Summit

AUTOMOTIVE LINUX SUMMIT