FOSS & Safety
The case of Zephyr

Team inovex
Karlsruhe · Köln · München · Hamburg
Berlin · Stuttgart · Pforzheim · Erlangen

Dr. Tobias Kästner

Tobias Kaestner

@tobiaskaestner

@tobiaskaestner

Solution Architect Medical IoT

#FOSS4MEDICAL

- PhD in Physics (long ago)
- SW/System Architect since 15 years
  - mainly Medical Devices
- Trainer & Technical Consultant
  - SW-Architecture, Zephyr, Yocto
- In Love w/ Zephyr since 2016
  - realised several prototype projects for life-science R&D
  - Maintainer of TiacSys-Bridle Project
  - Participant Zephyr Safety-WG

Agenda for today

- Functional Safety for SW Systems
- Zephyr, FOSS & Functional Safety
- Functional Safety & Beyond

inovex

A Functional Safety 101

Definition of Functional 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**

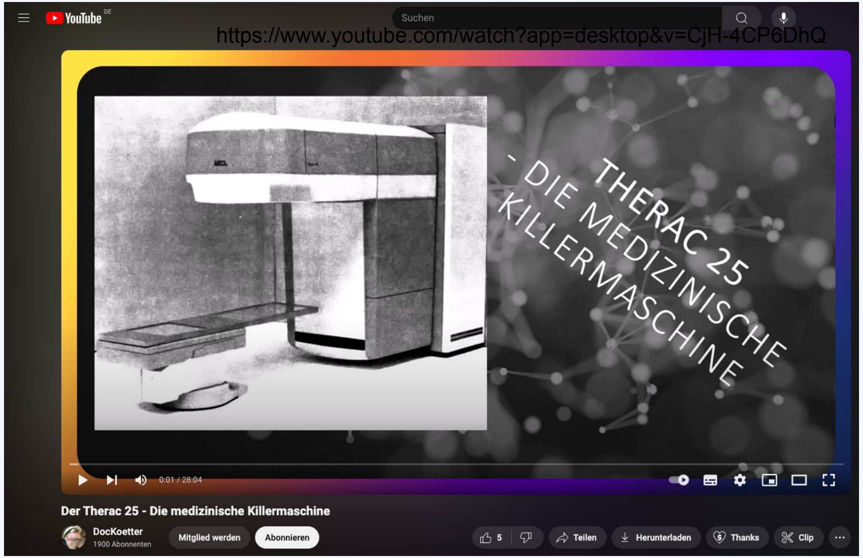
- **Functional Safety**
    - Part of safety that depends on a system or equipment operating correctly in response to its inputs
    - **Detecting potentially dangerous conditions**, resulting either in the activation of a protective or corrective device or mechanisms to **prevent hazardous events**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

Functional Safety

https://www.youtube.com/watch?v=CUjat1JA_rw

When Software lost its innocence

- Therac-25 was a radiation therapy machine in the 1980s sold by Atomic Energy of Canada Ltd.
- 100x radiation overdose from what operators had intended to apply
- three fatalities and many more injured as a consequence of treatment
- later severe SW design flaws were identified as the root cause for the malfunctioning of the machine


https://www.youtube.com/watch?app=desktop&v=CjH-4CP6DhQ

Der Therac 25 - Die medizinische Killermaschine

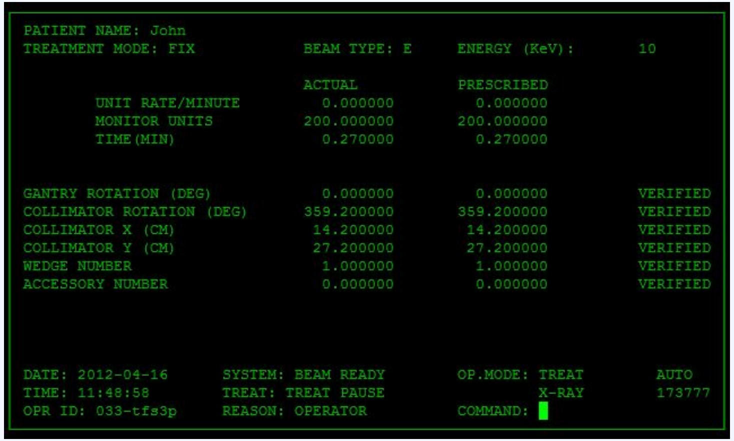
- Read the full story here:

https://en.wikipedia.org/wiki/Therac-25

Functional Safety for Software Systems

Therac-25 incidents became possible due to

1. inappropriate development process
   - single Developer doing all coding & testing
   - no risk analysis considering malfunctioning of SW
   - no final integration testing prior to deployment
2. inappropriate user interface
   - obscure error messages
   - operators could simply proceed
3. inappropriate SW design
   - SW-code reuse from previous machines that relied on HW-interlocks which Therac-25 had not
   - arithmetic overflows due to coding errors

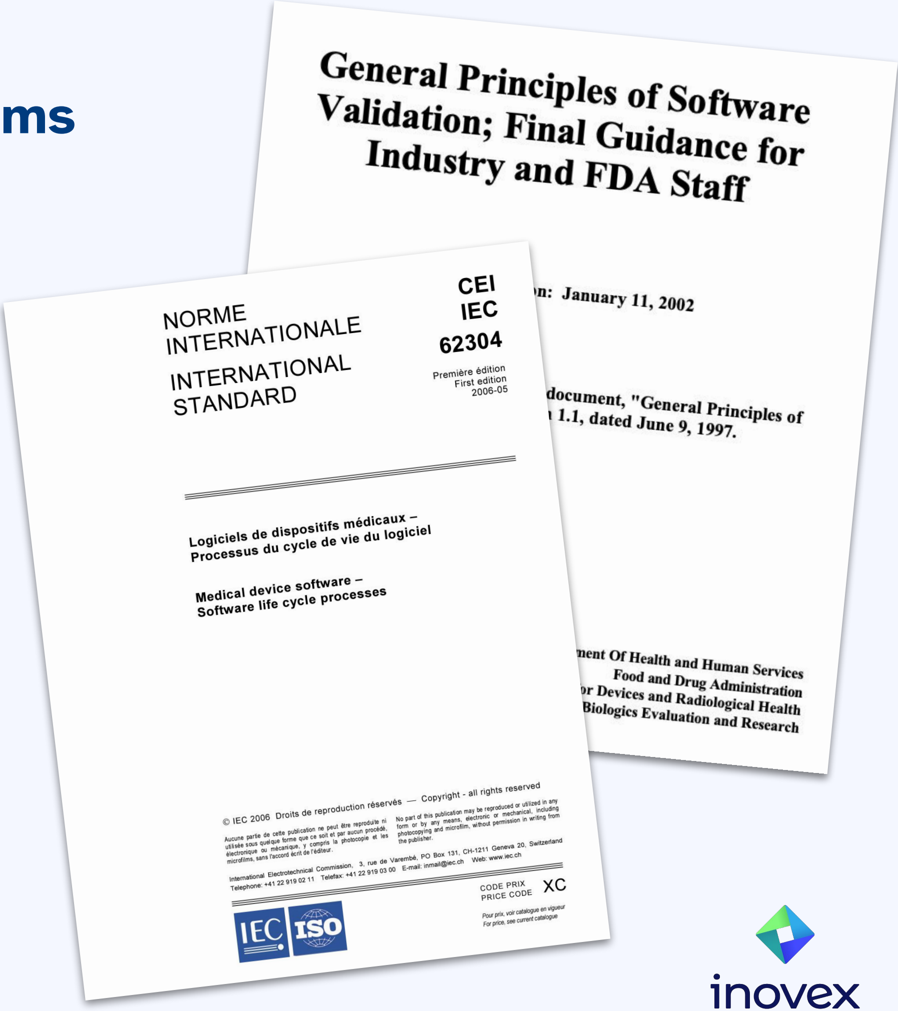


```
PATIENT NAME: John
TREATMENT MODE: FIX            BEAM TYPE: E     ENERGY (KeV):       10

                               ACTUAL           PRESCRIBED
        UNIT RATE/MINUTE         0.000000         0.000000
        MONITOR UNITS          200.000000       200.000000
        TIME(MIN)                0.270000         0.270000


GANTRY ROTATION (DEG)            0.000000         0.000000          VERIFIED
COLLIMATOR ROTATION (DEG)      359.200000       359.200000          VERIFIED
COLLIMATOR X (CM)               14.200000        14.200000          VERIFIED
COLLIMATOR Y (CM)               27.200000        27.200000          VERIFIED
WEDGE NUMBER                     1.000000         1.000000          VERIFIED
ACCESSORY NUMBER                 0.000000         0.000000          VERIFIED



DATE: 2012-04-16      SYSTEM: BEAM READY        OP.MODE: TREAT        AUTO
TIME: 11:48:58        TREAT: TREAT PAUSE                 X-RAY        173777
OPR ID: 033-tfs3p     REASON: OPERATOR          COMMAND: █
```

Functional Safety for Software Systems

Therac-25 lead to creation of **IEC 62304** and FDAs **“General Principles of Software for Medical Devices”**

to make sure manufacturers **act responsibly** during the creation of SW that could potentially harm or kill people



General Principles of Software Validation; Final Guidance for Industry and FDA Staff

n: January 11, 2002

document, "General Principles of
1.1, dated June 9, 1997.

nent Of Health and Human Services
Food and Drug Administration
or Devices and Radiological Health
Biologics Evaluation and Research

NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
62304

Première édition
First edition
2006-05

**Logiciels de dispositifs médicaux –
Processus du cycle de vie du logiciel**

**Medical device software –
Software life cycle processes**

© IEC 2006 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

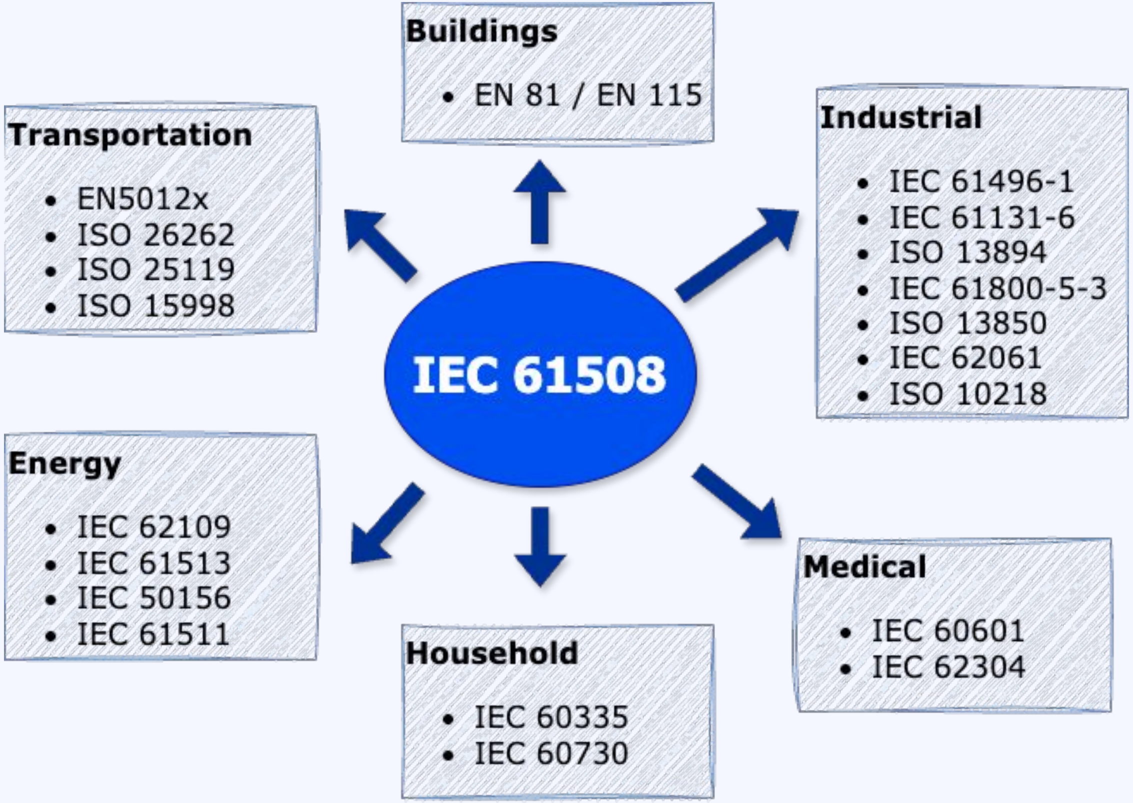
No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

CODE PRIX
PRICE CODE XC

*Pour prix, voir catalogue en vigueur
For price, see current catalogue*

The many standards of Functional Safety



IEC 61508

Buildings

- EN 81 / EN 115

Transportation

- EN5012x
- ISO 26262
- ISO 25119
- ISO 15998

Industrial

- IEC 61496-1
- IEC 61131-6
- ISO 13894
- IEC 61800-5-3
- ISO 13850
- IEC 62061
- ISO 10218

Energy

- IEC 62109
- IEC 61513
- IEC 50156
- IEC 61511

Household

- IEC 60335
- IEC 60730

Medical

- IEC 60601
- IEC 62304

taken from https://www.tuvsud.com/ko-kr/-/media/global/pdf-files/infographics/tuvsud-functional-safety-in-a-nutshell-infographic.pdf

What IEC 61508 wants us to do

Think ahead

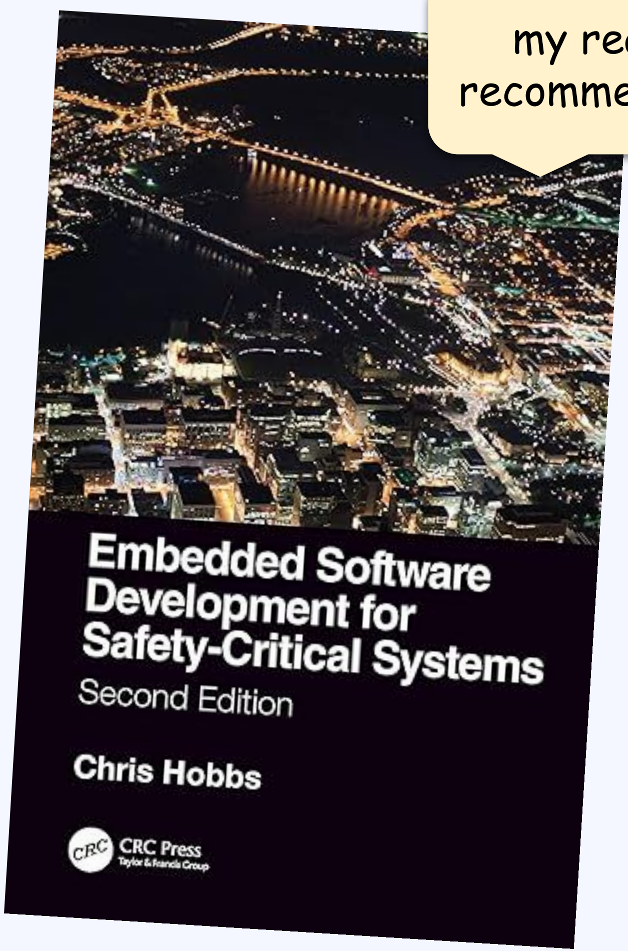
- Hazard & Risk analysis
- Failure analysis

Apply design methodology

- Architect for Safety
- Error Detection & Handling
- Expect the Unexpected
- Redundancy
- Out of scope software elements

Compile Evidence

- SW Verification & Validation
- Safety Case



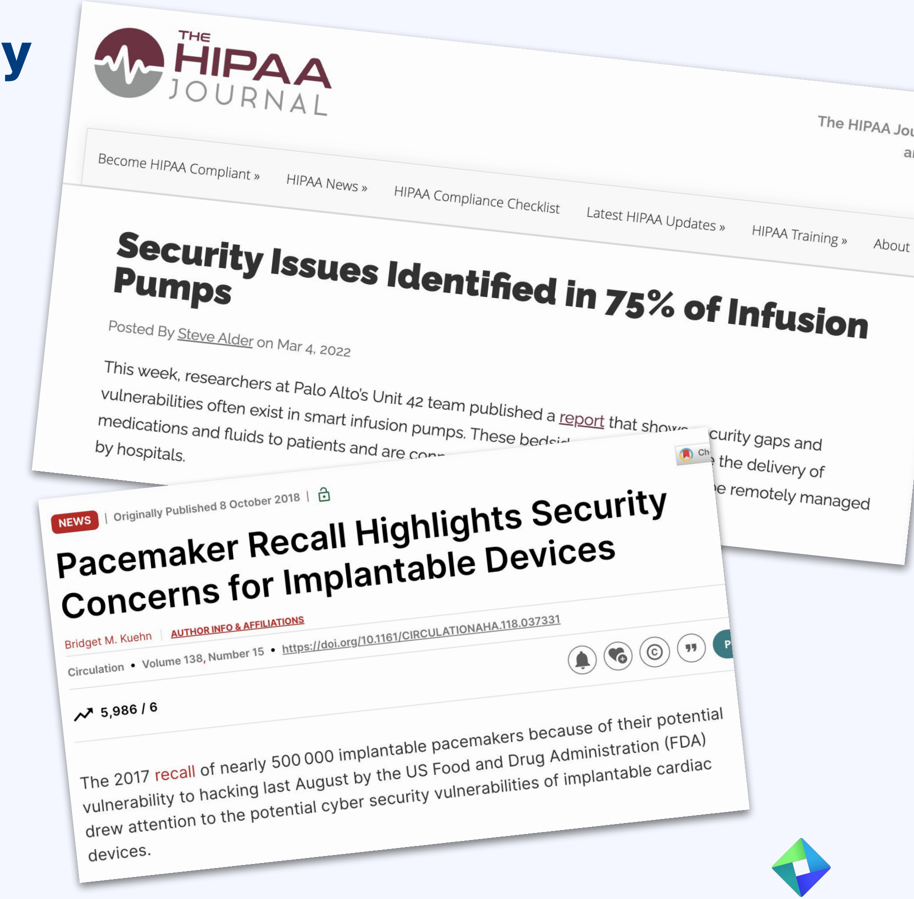
my reading recommendation

Sure, but what about SW Security

Security: Protect machines from (maliciously acting) humans

Safety: Protect humans from machines going wild

- Insecure systems most likely un-safe, too
  - e.g. attackers could nullify safety measures to harm people
- Yet, securing systems may introduce safety risks
  - e.g. FOTA updates to mitigate CVEs



THE HIPAA JOURNAL

The HIPAA Jou

Become HIPAA Compliant » HIPAA News » HIPAA Compliance Checklist Latest HIPAA Updates » HIPAA Training » About

Security Issues Identified in 75% of Infusion Pumps

Posted By Steve Alder on Mar 4, 2022

This week, researchers at Palo Alto's Unit 42 team published a report that show curity gaps and vulnerabilities often exist in smart infusion pumps. These bedsi the delivery of medications and fluids to patients and are con e remotely managed by hospitals.

NEWS | Originally Published 8 October 2018 |

Pacemaker Recall Highlights Security Concerns for Implantable Devices

Bridget M. Kuehn | AUTHOR INFO & AFFILIATIONS

Circulation • Volume 138, Number 15 • https://doi.org/10.1161/CIRCULATIONAHA.118.037331

5,986 / 6

The 2017 recall of nearly 500 000 implantable pacemakers because of their potential vulnerability to hacking last August by the US Food and Drug Administration (FDA) drew attention to the potential cyber security vulnerabilities of implantable cardiac devices.

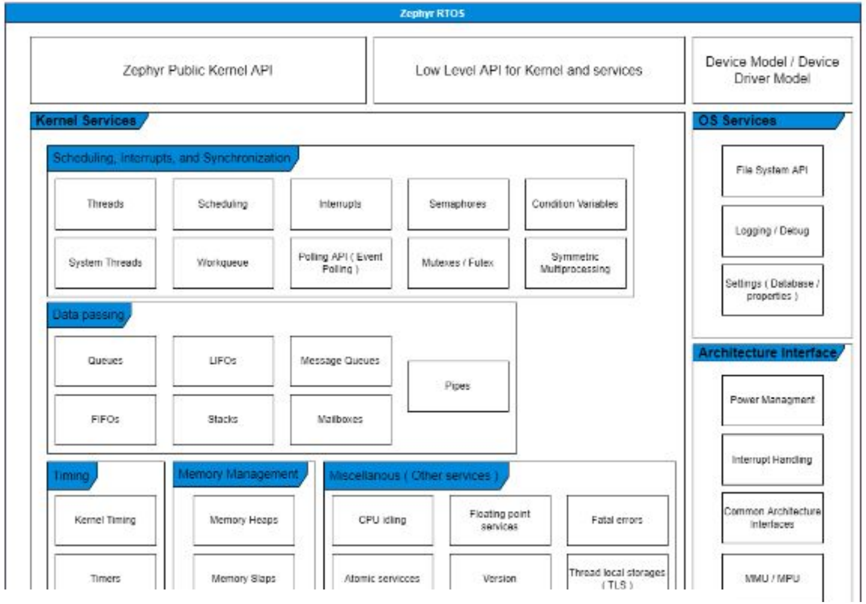
https://www.hipaajournal.com/security-issues-identified-in-75-of-infusion-pumps/
https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.118.037331

Functional Safety & Zephyr

“Preparing Zephyr for Safety Element out of Context Certification”, Nicole Pappler, alektometis - OSS Vienna 2024, used with permission of the author

Safety - Initial certification focus

- Start with a limited scope of kernel functions and interfaces
- Initial target is **IEC 61508 SIL 3 / SC 3**
  - Option for 26262 ASIL D certification has been included in contract with certification authority should there be sufficient member interest
- **Zephyr** to be treated as **Safety Element out of Context** (SEooC)



Starting scope

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

Zephyr - systematic 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

“Preparing Zephyr for Safety Element out of Context Certification”, Nicole Pappler, alektometis - OSS Vienna 2024, used with permission of the author

Zephyr - systematic 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) … “

- Providing a **safety scope definition**
- Creating **requirements** & establishing **traceability** to code & tests
- Creation of **system- & software specification**
- Definition of the **safety claims**
- Using the existing tests, establishing traceability & enhancing coverage
- Creation of the **safety manual**

“Preparing Zephyr for Safety Element out of Context Certification”, Nicole Pappler, alektometis - OSS Vienna 2024, used with permission of the author

Safety Work Product Creation

Safety Committee

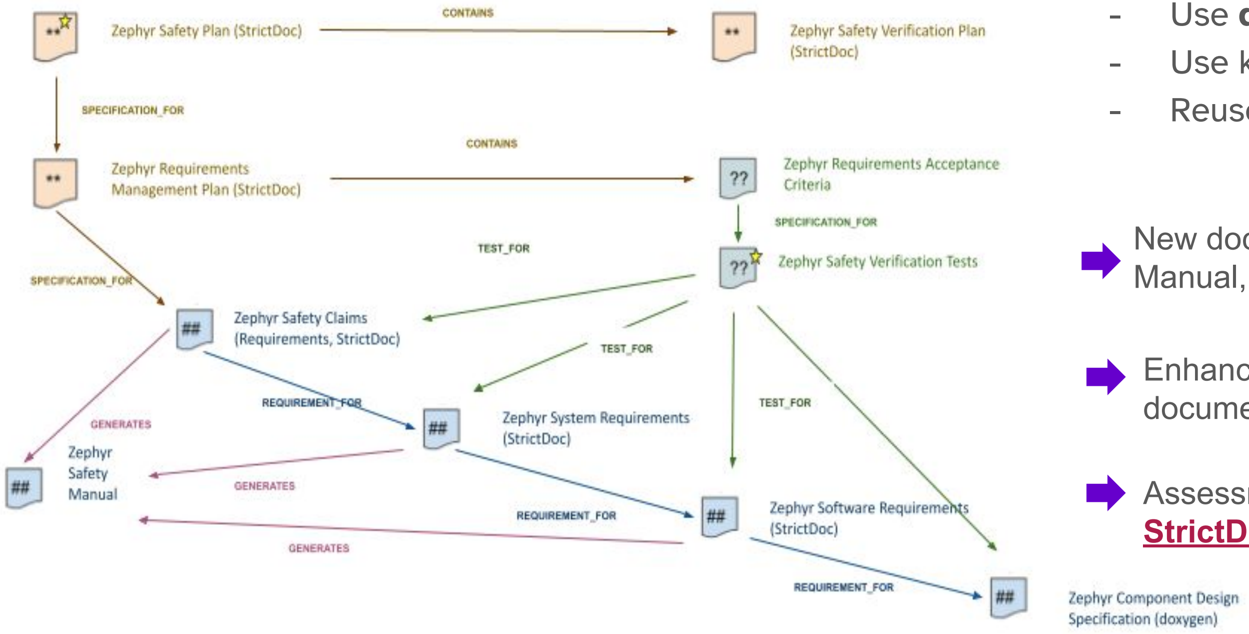
- Safety Certification strategy decisions
  - Scope of certification
  - Certification standards
  - Certification timeline
- Assessment and audit specific tasks
- Owner of certification artefacts
- Participation limited to the project’s platinum members, the safety architect and the functional safety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

Safety Working Group

- Enabling safety qualifications/certifications in the project
- Working on the creation of the required documentation and evidences
- Setting up requirements management tooling
- creating/deriving and documenting requirements
- Open to everyone to participate

“Preparing Zephyr for Safety Element out of Context Certification”, Nicole Pappler, alektometis - OSS Vienna 2024, used with permission of the author

Work Product Structure


**
Zephyr Safety Plan (StrictDoc)
CONTAINS
**
Zephyr Safety Verification Plan (StrictDoc)
SPECIFICATION_FOR
**
Zephyr Requirements Management Plan (StrictDoc)
CONTAINS
??
Zephyr Requirements Acceptance Criteria
SPECIFICATION_FOR
??
Zephyr Safety Verification Tests
TEST_FOR
SPECIFICATION_FOR
##
Zephyr Safety Claims (Requirements, StrictDoc)
TEST_FOR
TEST_FOR
REQUIREMENT_FOR
GENERATES
##
Zephyr System Requirements (StrictDoc)
##
Zephyr Safety Manual
GENERATES
REQUIREMENT_FOR
##
Zephyr Software Requirements (StrictDoc)
GENERATES
REQUIREMENT_FOR
##
Zephyr Component Design Specification (doxygen)

Principles for 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**

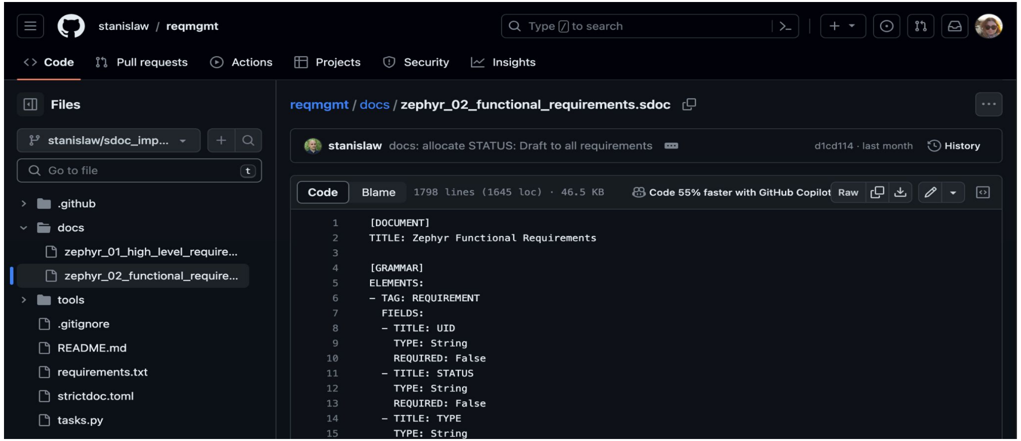
Current requirements work

- 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

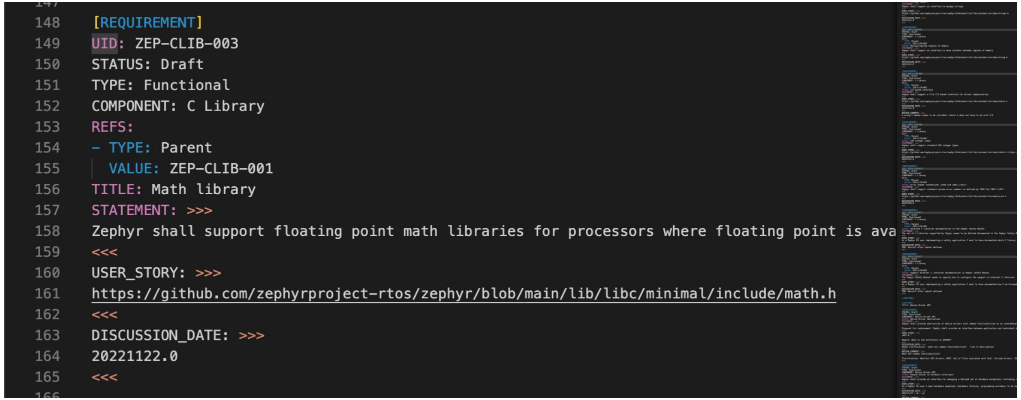


reqmgmt / docs / zephyr_02_functional_requirements.sdoc

stanislaw docs: allocate STATUS: Draft to all requirements d1cd114 · last month

1798 lines (1645 loc) · 46.5 KB

```
[DOCUMENT]
TITLE: Zephyr Functional Requirements

[GRAMMAR]
ELEMENTS:
- TAG: REQUIREMENT
  FIELDS:
  - TITLE: UID
    TYPE: String
    REQUIRED: False
  - TITLE: STATUS
    TYPE: String
    REQUIRED: False
  - TITLE: TYPE
    TYPE: String
```



```
[REQUIREMENT]
UID: ZEP-CLIB-003
STATUS: Draft
TYPE: Functional
COMPONENT: C Library
REFS:
- TYPE: Parent
  VALUE: ZEP-CLIB-001
TITLE: Math library
STATEMENT: >>>
Zephyr shall support floating point math libraries for processors where floating point is ava
<<<
USER_STORY: >>>
https://github.com/zephyrproject-rtos/zephyr/blob/main/lib/libc/minimal/include/math.h
<<<
DISCUSSION_DATE: >>>
20221122.0
<<<
```

Compliance with Coding Standards

Project already has defined Coding Guidelines in the docs, based on MISRA

Identification of Coding Guideline violations and adaption of the code

- Initially done by Bugseng on a separate branch
- Recently merged to the main branch

Coming soon: Static Analysis in the CI to check for adherence, powered by Eclair from BUGSENG

Can’t wait? Join the Safety Working Group

BTW, security has got a Working Group, too

Write us …

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

… talk to us, ask us, …

https://discord.gg/mgZkSmq2

… meet us

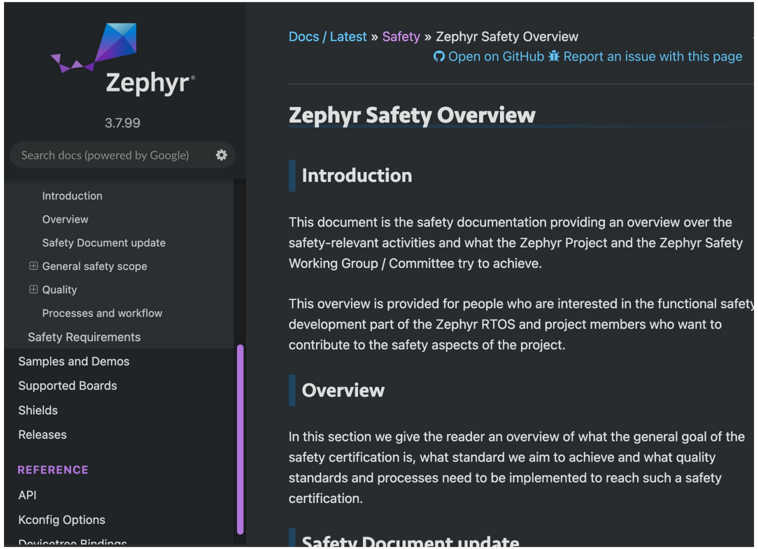
WG Video conference (almost) every **Tuesday 4pm CET**

https://docs.google.com/document/d/1HROTIAcp5TpzBdpAXlc2D7zmCvyFsg4NC4WTB5WK3oU/edit#heading=h.s8n3zq5dqe9f

“Preparing Zephyr for Safety Element out of Context Certification”, Nicole Pappler, alektometis - OSS Vienna 2024, used with permission of the author

Read the docs :-)

Safety Overview



Zephyr Safety Overview

Introduction

This document is the safety documentation providing an overview over the safety-relevant activities and what the Zephyr Project and the Zephyr Safety Working Group / Committee try to achieve.

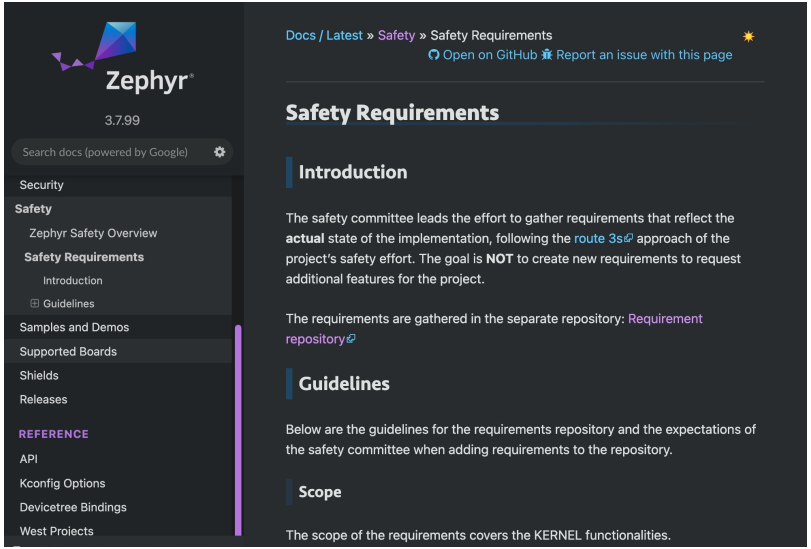
This overview is provided for people who are interested in the functional safety development part of the Zephyr RTOS and project members who want to contribute to the safety aspects of the project.

Overview

In this section we give the reader an overview of what the general goal of the safety certification is, what standard we aim to achieve and what quality standards and processes need to be implemented to reach such a safety certification.

Safety Document update

Requirements Guideline



Safety Requirements

Introduction

The safety committee leads the effort to gather requirements that reflect the **actual** state of the implementation, following the route 3s approach of the project’s safety effort. The goal is **NOT** to create new requirements to request additional features for the project.

The requirements are gathered in the separate repository: Requirement repository

Guidelines

Below are the guidelines for the requirements repository and the expectations of the safety committee when adding requirements to the repository.

Scope

The scope of the requirements covers the KERNEL functionalities.

Go to 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 :-)

Safety Working Group Project:

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

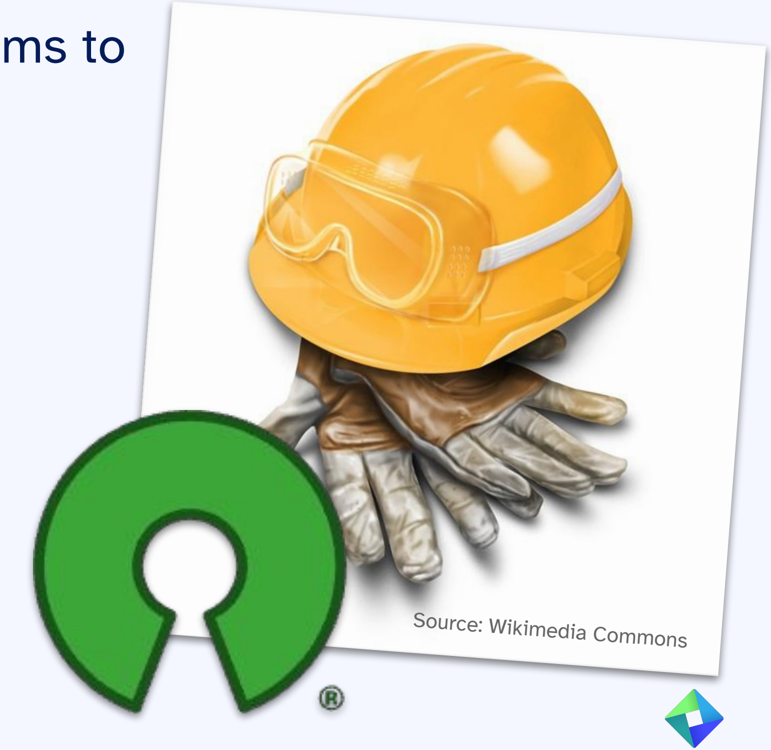
https://github.com/orgs/zephyrproject-rtos/projects/23/views/1

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

“To boldly go where no man has gone before”

Functional Safety & FOSS - The good ...

- More and more examples where FOSS aims to enter the safety-critical domain
  - XEN Hypervisor
  - ELISA (Embedded Linux in Safety Applications)
  - RTEMS
  - Eclipse ThreadX
  - **Zephyr**



Source: Wikimedia Commons

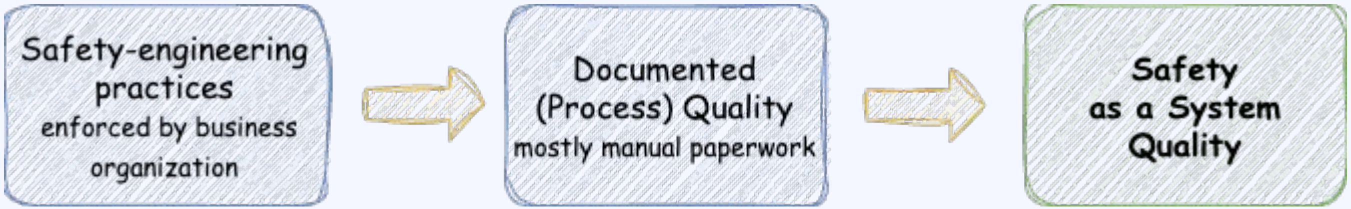
Functional Safety & FOSS - … the bad & the ugly

In practice several severe Challenges exist towards adoption of FOSS for safety-critical SW

- Non-free standards hamper participation
  - (almost) all ISO/IEC/EC standards
  - MISRA Coding Guidelines
- At their core safety standards are development process standards
  - tailored to fit business/enterprise processes
- Not all stakeholders in a FOSS project do actually care
  - unlike security which is (should be) on everyone's agenda

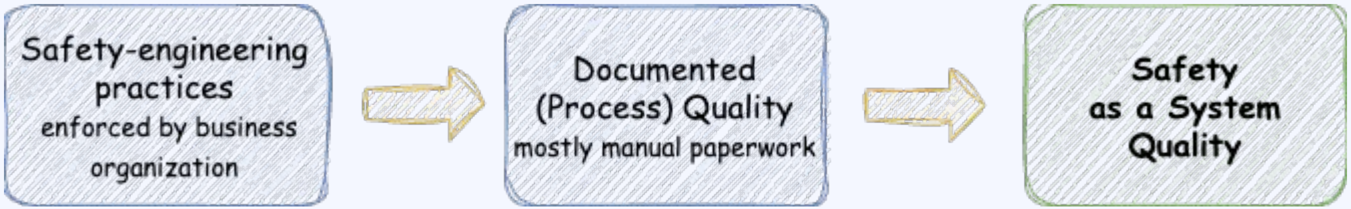
Check out
https://www.evs.ee/en/
most standards for a
reasonable pricee

Safety Standards as Process Standards


Safety-engineering practices
enforced by business organization
Documented (Process) Quality
mostly manual paperwork
Safety as a System Quality

- works best for **requirements driven** engineering
- however, FOSS better described as **contribution driven** engineering
    - mismatch forces FOSS projects to “backfill” many artifacts
    - extremely challenging to keep up w/ upstream development for these derived artifacts

Safety Standards as Process Standards



Safety-engineering practices enforced by business organization

Documented (Process) Quality mostly manual paperwork

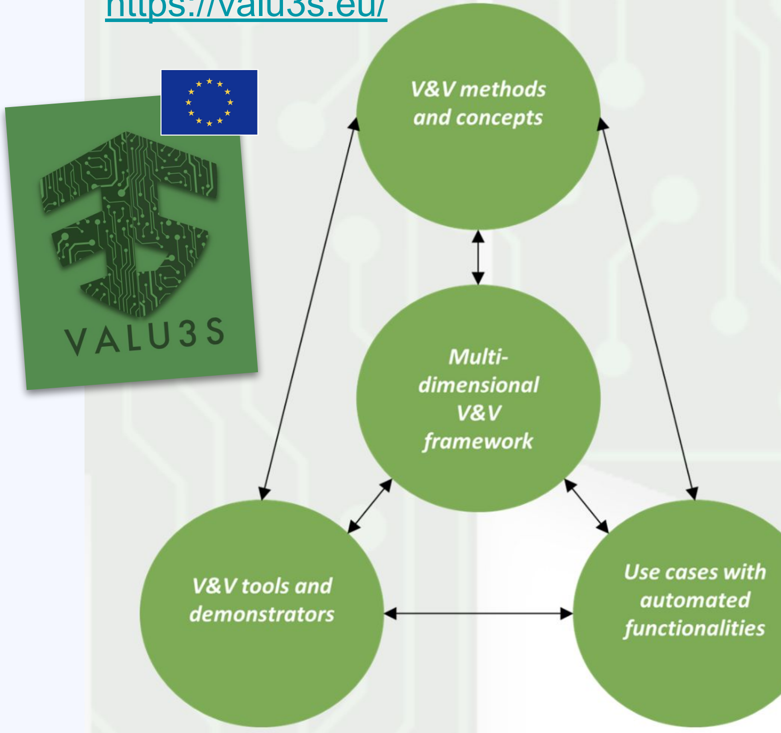
Safety as a System Quality

- assumes enforcement by business owners (liability)
- however, FOSS projects have a governance structure (at best)
    - have control over contribution guidelines to reject unsuitable work but no way to mandate “required” work to happen

Rethinking the current approach to functional safety

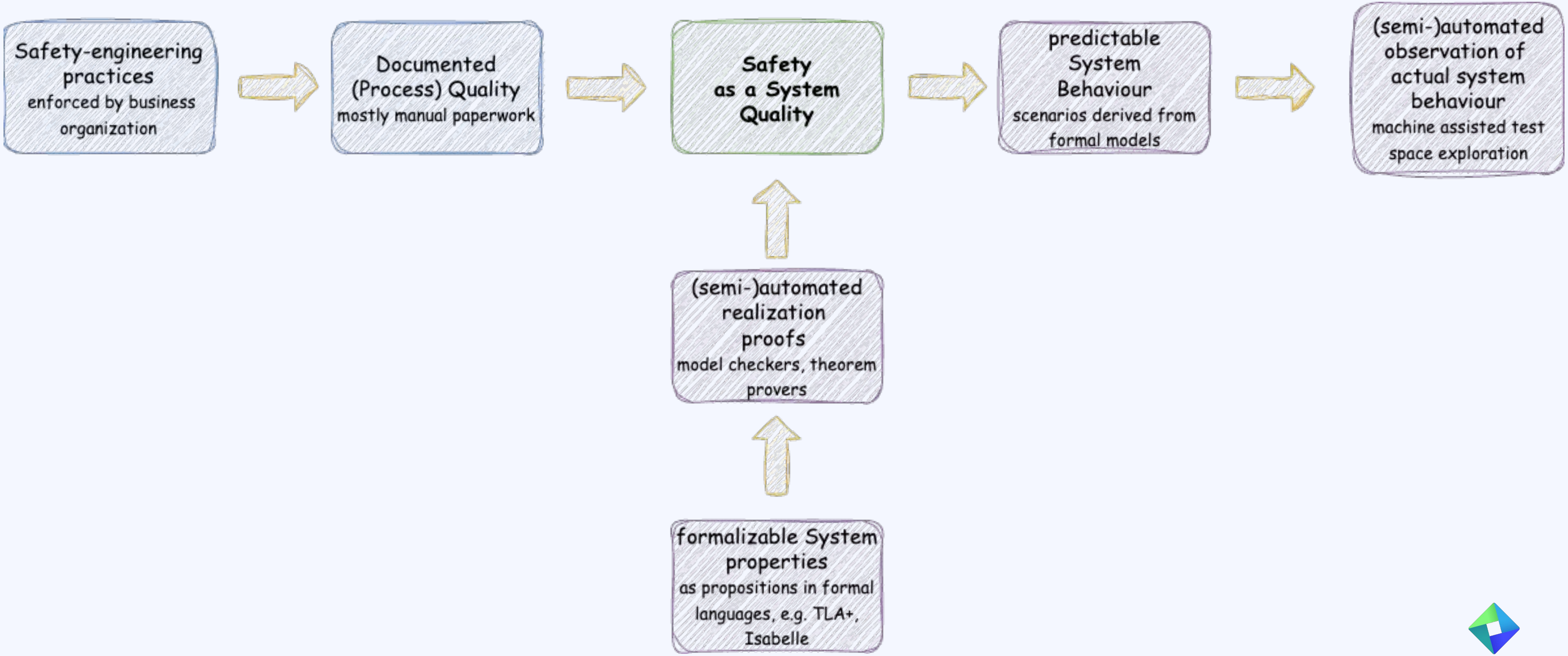
- Document-driven engineering hitting the complexity-wall anyways
- Rather than chasing the current paradigm boldly help to shape a new paradigm
- Holistic approach to Safety & Security
- Interesting work already available
  - ELISA
  - STPA (N. Leveson)
  - Formal Methods & Automation

https://valu3s.eu/


VALU3S
V&V methods and concepts
Multi-dimensional V&V framework
V&V tools and demonstrators
Use cases with automated functionalities

VALU3S focus on improving the V&V process.

So what to do - Ideas? Anyone?



Safety-engineering practices
enforced by business organization

Documented (Process) Quality
mostly manual paperwork

Safety as a System Quality

predictable System Behaviour
scenarios derived from formal models

(semi-)automated observation of actual system behaviour
machine assisted test space exploration

(semi-)automated realization proofs
model checkers, theorem provers

formalizable System properties
as propositions in formal languages, e.g. TLA+, Isabelle

So what now?

Joint-venture between academia, industries and open source projects/foundations needed

- Many open questions
  - Someone has to figure out what to do and how to do it
- It is going to cost something
  - Someone has to pay the bill
- We share in the sowing, we share in the harvest
  - Someone needs to make sure things work out for the good of all.

Source: Pixabay

Summary

- Software is everywhere, for our own sake we better care for safety & security
- The landscape of safety standards is wide and big
- Zephyr aims to become certified against IEC 61508 as SEooC (Route 3s) at SIL 3
  - Established Safety Committee and Safety Working Group to carry out necessary work
  - Done when it’s done, the more the faster
- Need to rethink our approach to functional safety
  - more and more FOSS projects will suffer similar problems

Thank You

Zephyr Hands-On Trainings
starting 2025: Jan 22/23, Apr 02/03, Jul 02/03

Find out more
https://www.inovex.de/de/training/zephyr-basic-training/

Dr. Tobias Kästner
Solution Architect Medical IoT

tobias.kaestner@inovex.de

+49 152 3314 8940

Allee am Röthelheimpark 11,
91052 Erlangen

Tobias Kaestner

@tobiaskaestner

@tobiaskaestner