solcept
electronics • software

Software Safety Analyse

Ein pragmatischer Ansatz zur Eliminierung systematischer Fehler

Daniel Megnet, Solcept AG

Weitere Informationen:
https://www.solcept.ch/de/blog/software-safety-analyse/

Motivation I

- Erhöhung der SW-Sicherheit durch Eliminierung systematischer Fehler
- Gefordert für Projekten im Bereich der funktionalen Sicherheit
  - z. B. Automotive ISO26262-6, Section 7.4.13:

> **7.4.13** Safety analysis shall be carried out at the software architectural level in accordance with ISO 26262-9:2011, Clause 8, in order to:
>
> — identify or confirm the safety-related parts of the software; and
>
> — support the specification and verify the efficiency of the safety mechanisms.

Motivation II

- z. B. IEC61508-3, 7.4.3 Requirements for software architecture design

NOTE 6 For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the software architecture design should be considered:

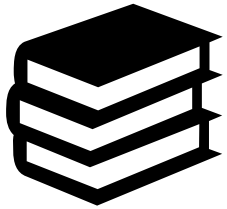
– completeness with respect to software safety requirements specification;
– correctness with respect to software safety requirements specification;
– freedom from intrinsic design faults;

- z. B. ISO13849-1, 4.6.2 Safety-related embedded software (SRESW)

For SRESW for components with PL_r a to d, the following basic measures shall be applied:

— software safety lifecycle with verification and validation activities, see Figure 6;
— documentation of specification and design;
— modular and structured design and coding;
— control of systematic failures (see G.2);

Normen / Literatur



- Nur spärliche Hinweise zur Methodik in den Normen
- Wenig Hilfreiches in der Literatur
- Beispiel: Methodischer Hinweis in ISO26262-9, section 8:

Qualitative analysis methods include:

— qualitative FMEA at system, design or process level;

— qualitative FTA;

— HAZOP;

— qualitative ETA.

NOTE 1 The qualitative analysis methods listed above can be applied to software where no more appropriate software-specific analysis methods exist.

Ausgangslage

- Eine Software Sicherheitsanalyse muss gemacht werden
- Methodische Hinweise spärlich:
  - Qualitativ (FMEA, FTA)
  - Auf Architektur-Level
- Wie weiter?

Typisches Umfeld

… aber wir machen doch schon so viel …

- Coding Guidelines (prevention of low-level, language-specific errors)
- Design Guidelines (e.g. hierarchical structure, restricted coupling between SW components)

- Code Reviews (acc. checklist)
- Statische Code Analyse (MISRA Rules, …)
- Testing, Verifikation, Validation

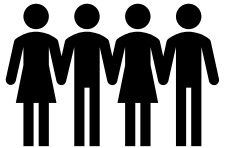
Untere SW-Schichten sind bereits genügend abgedeckt.

Prüfung der SW Architektur auf systematische Fehler fehlt.
Manche SW Architekturen begünstigen Fehler.

Methodik (Grob)

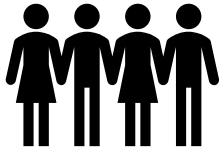
- FMEA
  - Bottom-Up
  - Qualitativ (Severity, Occurrence, Detection)
  - Expertengremium (3 – 5 SW-Entwickler/ -Architekten)
- Beschränkung auf SW-*Architektur*
  - Analyse aller sicherheitsrelevanten SW-Packages
  - Definition der zu untersuchenden Architekturelemente und ihrer möglichen Fehler

Team-Bildung



- Expertengremium
  - nicht interdisziplinär, d.h. nur SW-Spezialisten
  - Möglichst unterschiedlicher Projekt- bzw. Erfahrungshintergrund
  - Kenntnis des zu beurteilenden Systems und der SW Architektur
- Moderator
  - Prozessführung …

Moderator



- Organisation der Experten-Meetings
- Voraussetzungen schaffen
  - notwendige SW-Dokumentation in Auftrag geben
  - für ein offenes Klima sorgen
- Dokumentation der Ergebnisse
  - Protokolle der Experten-Meetings
  - Für Vollständigkeit sorgen
  - Entscheide vorantreiben
  - Verfassen des Analyseberichts

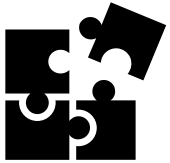
Sicherheitsziel

- Das Sicherheitsziel muss allen bekannt sein
  - Beispiel Sensor: Keine Ausgabe falscher Messwerte
  - Beispiel Roboter: Bewegung innert max. 100 ms stoppen, falls der Roboterarm weniger als 20 cm Abstand zu einem Objekt hat.
- Ergebnisse aus früheren Analysen können nicht übernommen werden, wenn das Sicherheitsziel nicht übereinstimmt.

Analysebereich minimieren

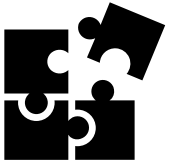
- Listen von Architekturelementen und dazugehörigen Fehlerarten erstellen
- Welche Architekturelemente werden im Team betrachtet?
- Liste der vorhandenen Präventions- und Detektionsmassnahmen
- Was wird über Guidelines und Reviews abgedeckt?

Architekturelemente I
Allgemein/ Funktional



- Stack: size too small/ underrun/ overflow
- Fixed-point Arithmetik: overflow/ quantization effects
- Interfaces: missing time-out/ handling of collisions
- Control loops: stability, output clipping

Architekturelemente II Kontrollfluss



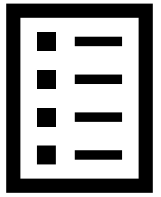
- State Machines
  - Wrong or missing transition
- Execution Framework/ Scheduling
  - tasks not called in correct sequence, dead-locks
  - violation of real-time constraints
- Interrupts:
  - IRQ flag not cleared

Architekturelemente III Datenfluss

- circular buffers, shared memory, message queue etc.:
  - Execution context
  - non-atomic read/write, critical section missing
  - Data synchronization
- Filter

https://www.solcept.ch/de/blog/datenflussdiagramme/

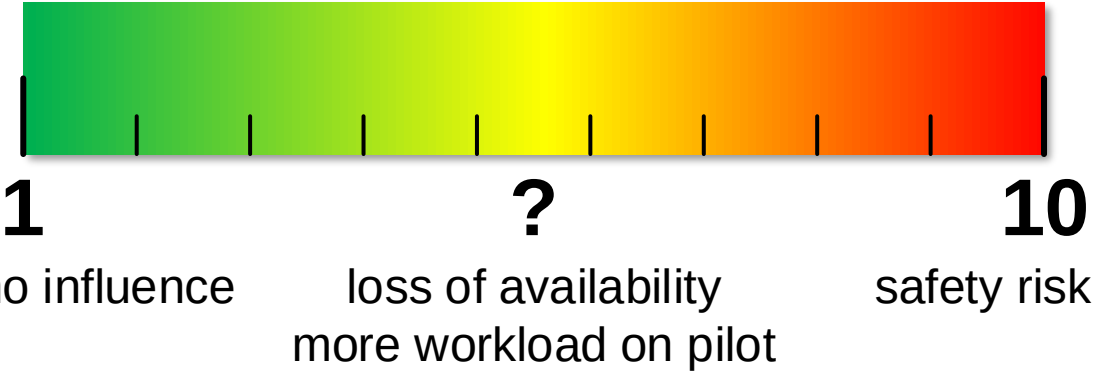
Liste der Präventions- und Detektionsmassnahmen



- Vorhandene Massnahmen des SW Entwicklungsprozesses auflisten
  - Anwendung von Guidelines, Code Reviews, Unit Tests, Robustness Tests, Testabdeckung
- Vereinfachte Behandlung während der Analyse
  - tiefe Occurrence und Detection Ratings
  - keine Massnahmen definieren, die ohnehin durchgeführt werden.
  - Team Review der Coding Rules, Design Guidelines, Review Checklisten

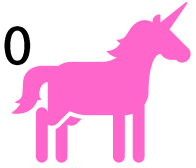
Bewertungskataloge I Severity

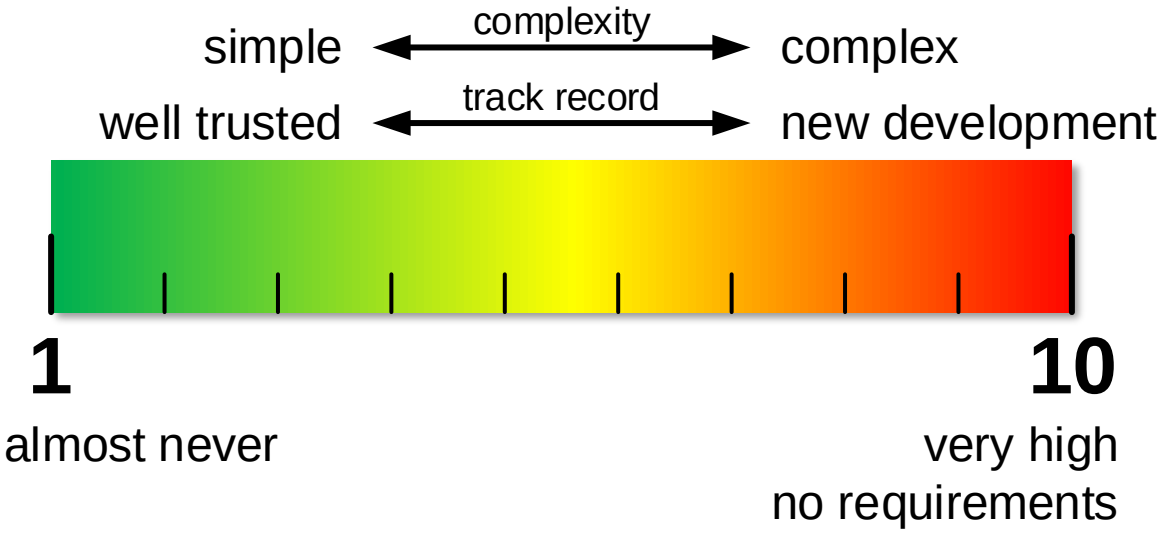
- Will man nur die grobe Sicherheit beurteilen, so kann man vereinfachen
- Zwischenstufen sind häufig sinnvoll



Bewertungskataloge II Occurrence

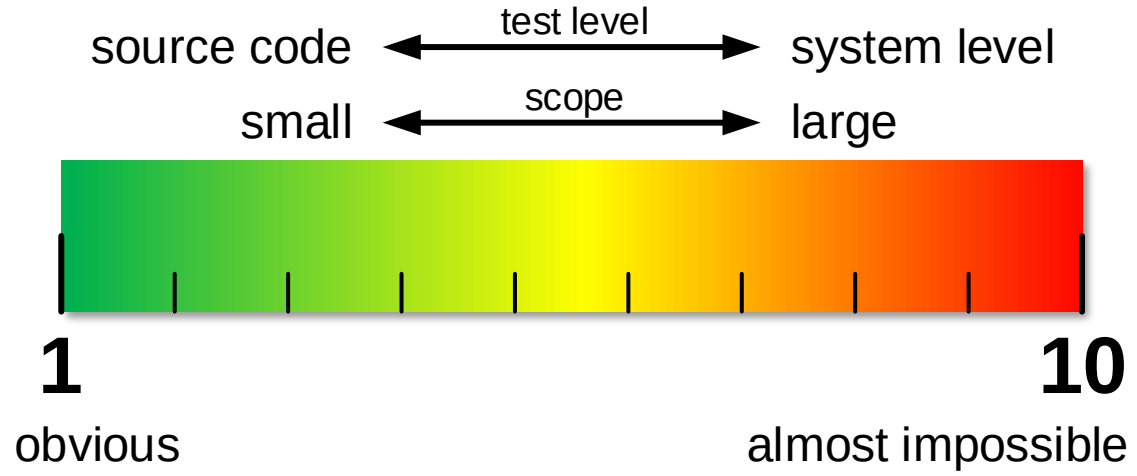
- Fehler bereits durch präventive Massnahmen eliminiert → low
- Fehlende oder unvollständige Anforderungen bezgl. Sicherheit→10





Bewertungskataloge III Detection

- Verifikationsmassnahmen: Test, Review- oder Analyse-Massnahmen
- Wenn man keine Ahnung hat, wie ein Fehler detektiert werden könnte… →10

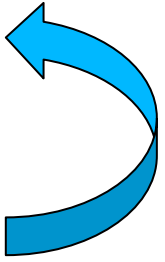


Risikomatrix

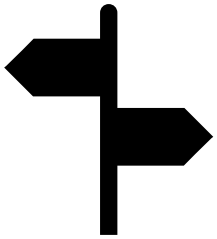
- In der Regel genügt die Berechnung der Risikoprioritätszahl
  - Produkt der Bewertungen von Severity, Occurrence und Detection, RPN = S * O * D
  - Sicherstellen, dass alle Skalen den gleichen Bereich (z. B. 1…10) haben.
- Festlegung eines RPN-Grenzwertes
  - Oberhalb des Grenzwerts müssen Massnahmen getroffen werden
- Evt. Bereich für Ermessensspielraum
- Kompliziertere Bewertungen sind möglich / sinnvoll
  - S x O, S x D, O x D

Ablauf der Experten-Meetings

- Sicherheitskritische Software Teile und die darin enthaltenen **wesentlichen** Architekturelemente im Experten-Team diskutieren
- Architekturelemente bewerten
- Massnahmen definieren und umsetzen
- Neubewertung



Entscheidungsfindung



- Die Experten im Team sind sich uneinig …
- Berücksichtigen Sie die Interessenlage
  - Wer im Expertenteam ist bei welcher Organisation angestellt
  - Auftraggeber, Kunde, Entwicklungspartner, Berater,…
- Dokumentieren Sie, wie die Entscheide zustande kommen

Analysebericht

- Der Analysebericht ist an die Auftraggeber und an die Zertifizierungs-behörden gerichtet.
- Nur der Bericht wird gelesen. Der Rest der Analyse ist nur relevant, falls es später zu juristischen Untersuchungen kommt.

Inhalt des Berichts:
https://www.solcept.ch/de/blog/software-safety-analyse/

solcept
electronics • software

Zeit für Fragen?

Solcept Blogs and Whitepapers:

https://www.solcept.ch/de/blog/