

## Functional Safety wird zur Software-Disziplin



KI-gestützte Robotik  
**Funktionale Sicherheit  
für Humanoide**

Interview  
**Mehr Resilienz  
im Maschinenbau**

Mikrosegmentierung  
**IT/OT-Grenzen  
neu denken**

# Functional Safety wird zur Software-Disziplin

von Michael Plankensteiner



Bilder: Neuron Automation

Während sich Steuerungs- und IT-Welt rasant weiterentwickeln, verharrt Safety vielerorts noch in Architekturen aus einer Zeit, in der Hardware teuer und Software vor allem Mittel zum Zweck war. Neuron Automation denkt Functional Safety grundlegend neu und konzipiert sie als standardisierte, skalierbare und softwaredefinierte Plattform.

Neue Maschinenkonzepte, wachsende Robotik-Anwendungen, modulare Systemarchitekturen sowie verschärfte regulatorische Anforderungen treiben den Bedarf an sicheren Funktionen zunehmend voran. Doch genau dieses Wachstum legt die strukturellen Schwächen klassischer Safety-Ansätze offen: Functional Safety wächst schneller als der restliche Automatisierungsmarkt und wird gleichzeitig zum strukturellen Flaschenhals. Denn wenn Safety weiterhin projektspezifisch entwickelt wird, bremst sie Innovation aus. Daher hat sich Neuron zum Ziel gesetzt, Safety zu einer standardisierten, wiederverwendbaren Plattform zu machen.



## Event-Tipp

Besuchen Sie Neuron auf der Hannover Messe:  
Halle 27, Stand 659

## Vom Hardware-Silo zur softwaredefinierten Safety-Architektur

Über Jahrzehnte wurde funktionale Sicherheit primär über physische Trennung realisiert: separate Sicherheitscontroller,

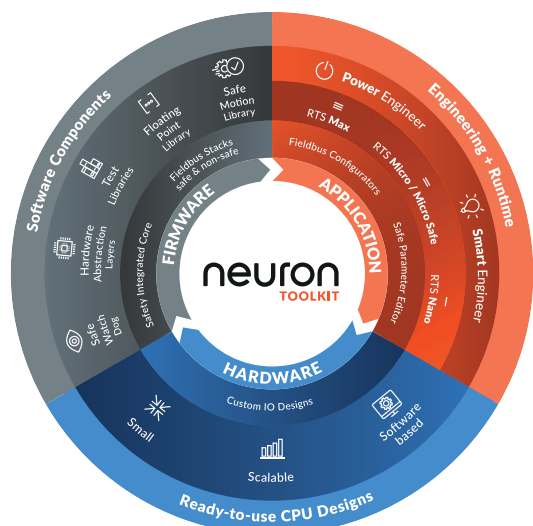
redundante Hardwarepfade oder dedizierte Safety-CPU's. In einer Zeit begrenzter Rechenleistung und klar definierter Systemgrenzen war dieses Modell technisch sinnvoll. Bei Neuron ermöglichen moderne Multicore-SoCs die parallele Ausführung sicherheitskritischer und nicht-sicherheitskritischer Funktionen auf einer gemeinsamen Plattform. Die notwendige Trennung erfolgt nicht mehr physisch, sondern logisch auf Runtime-Ebene. Duale Safety-Kanäle lassen sich softwaredivers auf einem Prozessor realisieren, mit zyklischer gegenseitiger Überwachung und normkonformer Auslegung bis SIL3/PL e. Parallel dazu entwickeln sich Steuerungen zu konvergenten Plattformen. Automatisierung, Motion, Edge-Funktionalität, IT-Anwendungen und Safety wachsen technologisch zusammen. Separate Safety-Controller geraten damit zunehmend in den Hintergrund. So wird Safety zum integralen Bestandteil moderner Systemarchitekturen.

## Blueprint für integrierte Functional Safety

Neuron verfolgt dafür einen konsequent plattformbasierten Ansatz. Statt Sicherheitsfunktionen für jedes Projekt neu zu entwickeln, stellt das Unternehmen ein durchgängiges Safety-Toolkit bereit: mit Multicore-Referenzarchitekturen, vorzertifizierter Firmware, einer hardwareunabhängigen Safety-Runtime, Engineering-Tools, Kommunikationskomponenten sowie Zertifizierungs-Assets. Damit liefert das Unternehmen einen Blueprint für integrierte Functional Safety: Sicherheitsfunktionen lassen sich aus standardisierten Bausteinen modular zusammensetzen, anstatt für jedes Projekt erneut bei null zu beginnen. Gerade für Gerätehersteller gewinnt dieser Plattformansatz zunehmend strategische Bedeutung. Functional Safety ist für viele Unternehmen kein Differenzierungsmerkmal mehr, sondern eine notwendige Voraussetzung für den Marktzugang. Gleichzeitig binden Entwicklung, Zertifizierung und Pflege erhebliche Ressourcen. Hinzu kommt, dass viele Gerätehersteller nicht über tiefes Know-how im Bereich Functional Safety verfügen. Der Aufbau entsprechender Kompetenzen ist teuer und bindet langfristig Entwicklungsressourcen. Der Plattformgedanke verschiebt hier den Fokus: Hersteller können sich stärker auf ihre eigentlichen Innovationsfelder konzentrieren, während Safety als robustes, skalierbares Fundament bereitsteht.

## Softwarebasierte Safety als Schlüsseltechnologie

Ein zentraler Baustein dieses Ansatzes von Neuron ist die softwarebasierte und hardwareunabhängige Safety-Lösung HIS (Hardware Independent Safety). Sie erlaubt es, sicherheitsgerichtete Funktionen auf unterschiedlichster Standard-Industriehardware auszuführen – von Embedded-ARM-Systemen bis zu leistungsfähigen Controllerplattformen. Damit wird Functional Safety von spezialisierter Hardware entkoppelt und zu einer integrierbaren Softwarefunktion.



Der Neuron Blueprint für integrierte Functional Safety.

Hersteller gewinnen dadurch mehr Freiheitsgrade bei Architektur, Performance und Plattformstrategie.

Die Runtime ist auf Auslegungen bis SIL3/PL e konzipiert und unterstützt gängige Safety-over-Ethernet-Protokolle sowie Black-Channel-Architekturen. Einen wichtigen Meilenstein stellt die finale Konzeptfreigabe der hardwareunabhängigen Safety-Lösung von Neuron durch TÜV Süd dar.

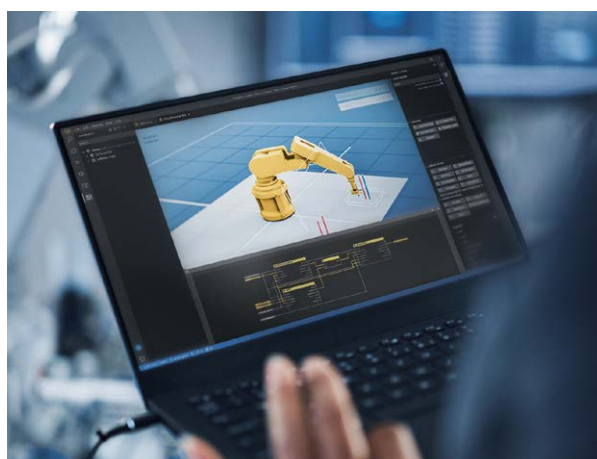
### Virtualisierung der Safety-Runtime

Wenn Safety zunehmend softwaredefiniert umgesetzt wird, ist Virtualisierung der nächste logische Schritt. Moderne Maschinen- und Robotikarchitekturen verlangen nach höherer Rechenleistung, Edge-Integration und modularen Software-Updates. Neuron entwickelt die Safety-Runtime deshalb so, dass sie auch in virtualisierten Umgebungen eingesetzt werden kann, beispielsweise auf containerisierten Steuerungsplattformen oder in RTOS-basierten Systemen. Gerade in modular aufgebauten Maschinenarchitekturen, der Robotik oder autonomen Systemen wird diese Flexibilität zu einem wichtigen strategischen Faktor.

### OT trifft IT

Neben der Systemarchitektur wurde auch das Engineering grundlegend neu gedacht. Während klassische Safety-Projekte häufig in isolierten Werkzeuglandschaften mit proprietären Datenformaten und begrenzten Automatisierungsmöglichkeiten entstehen, überträgt Neuron moderne IT-Methoden auf das Safety-Engineering: Anstelle binärer Engineering-Daten basieren Projekte auf textbasierten Strukturen. Dadurch werden etablierte Softwareentwicklungsprozesse möglich wie beispielsweise Versionsverwaltung mit Git, automatisierte Builds oder Continuous-Integration-Workflows.

Mit dem ‚Neuron Power Engineer‘ stellt das Unternehmen eine Entwicklungsumgebung bereit, in der Safe- und Non-Safe-Applikationen gemeinsam entwickelt werden können. Sowohl IEC-61131-3-Sprachen als auch C/C++ werden innerhalb einer Plattform unterstützt, während Safe- und Non-Safe-Code



Softwarebasierte Safety als Schlüsseltechnologie.

auf Runtime-Ebene strikt voneinander getrennt bleiben.

Der ‚Neuron Smart Engineer‘ erweitert diesen Ansatz um ein webbasiertes Engineering-Frontend sowie moderne Erweiterungsmöglichkeiten über Visual-Studio-Code-Extensions. So verlässt Safety-Engineering die proprietären Inseln und Safety wird in die gleichen Entwicklungsprozesse integriert wie moderne Software.

### KI als Assistenz im Safety-Engineering

Auch Künstliche Intelligenz hält zunehmend Einzug ins Engineering. Neuron integriert KI-Funktionen direkt in die Engineering-Plattform, um Ingenieure bei der Erstellung von Code-, Test- und Dokumentationsartefakten zu unterstützen. Auf Basis strukturierter Anforderungen kann die KI beispielsweise Vorschläge für Applikationscode generieren, Testfälle ableiten und Konsistenzprüfungen durchführen. Gerade im Bereich Test und Verifikation eröffnet sich dadurch ein erhebliches Effizienzpotenzial.

KI ersetzt keinen Safety-Ingenieur, kann aber dessen Arbeit insbesondere bei Tests, Dokumentation und Konsistenzanalysen deutlich produktiver machen.

### Functional Safety wird zur Commodity

Functional Safety bleibt normativ anspruchsvoll, doch ihre Umsetzung verändert sich grundlegend. Standardisierte Architekturen, hardwareunabhängige Runtimes und integrierte Engineering-Plattformen verschieben den Fokus zunehmend von projektspezifischen Einzelentwicklungen hin zu industrialisierten Plattformmodellen. So wird Functional Safety zur Commodity – nicht im Sinn geringerer Anforderungen, sondern als Technologie, die sich universell integrieren, wirtschaftlich beherrschen und strategisch nutzen lässt.



**Michael Plankensteiner**

ist CEO von Neuron Automation.

# Safety industrialisieren

Was den Ansatz von Neuron Automation von klassischen Safety-Anbietern im Detail unterscheidet, erläutert Robert Mühlfellner, CTO bei Neuron Automation.

## **Sie wollen einen neuen Branchenstandard definieren. Was bedeutet das konkret?**

Functional Safety wird heute in vielen Projekten immer noch wie ein Einzelkunstwerk entwickelt – mit eigenen Hardwaredesigns, individuellen Toolchains und neuen Zertifizierungszyklen. Das macht Projekte teuer, langsam und schwer skalierbar.

Unser Ziel ist es, Safety mit standardisierten Architekturen, wiederverwendbaren Referenzdesigns sowie einer einheitlichen Runtime- und Engineering-Basis zu industrialisieren. Sicherheitsfunktionen sollen nicht jedes Mal neu entwickelt werden müssen, sondern auf zertifizierten Fundamenten aufbauen. Genau darin sehen wir den nächsten industriellen Standard.

## **Was unterscheidet Ihren Ansatz von klassischen Safety-Anbietern?**

Viele Anbieter definieren Safety nach wie vor über spezialisierte Hardware wie zum Beispiel dedizierte Safety-CPU's oder separate Safety-Controller. Unser Ansatz ist ein anderer, nämlich softwarezentriert und hardwareunabhängig.

Mit unserer HIS-Runtime lassen sich sicherheitsgerichtete Funktionen bis SIL3/PL e auf Standard-Industriehardware realisieren. Multicore-Architekturen und softwarediverse Konzepte übernehmen dabei Aufgaben, die früher über physische Hardwaretrennung gelöst wurden. Functional Safety wird dadurch skalierbarer und stärker architekturgetrieben.

## **Was steckt hinter Ihrem Konzept des One-Stop-Shop?**

Heute müssen Hersteller Functional Safety aus vielen einzelnen Bausteinen zusammensetzen: Hardware, Firmware, Tool-



**Robert Mühlfellner**

ist CTO bei Neuron Automation.

» Virtualisierung wird ein zentraler Baustein moderner Safety-Architekturen. «

chain, Kommunikationsprotokolle und Zertifizierung. Genau diese Fragmentierung macht Safety-Projekte so komplex und aufwendig. Unser Ansatz ist eine integrierte Gesamtlösung. Wir liefern Referenzarchitekturen, Firmware, Runtime, Engineering-Tools und Zertifizierungs-Assets aus einer Hand. Für Hersteller bedeutet das deutlich weniger Integrationsaufwand, kürzere Entwicklungszeiten und vor allem besser planbare Safety-Projekte.

## **Welche Bedeutung spielt die Virtualisierung der Safety-Runtime künftig?**

Eine sehr große. Wenn Safety dauerhaft an dedizierte Hardware gebunden bleibt, wird sie schnell zum Innovationshemmnis. Deshalb entwickeln wir unsere Runtime bewusst so, dass sie auch in virtualisierten und containerisierten Umgebungen eingesetzt werden kann. Das eröffnet neue Möglichkeiten wie etwa Hardwarekonsolidierung, flexible Plattformstrategien oder virtuelle Entwicklungsumgebungen. Virtualisierung ist damit kein Zukunftsszenario mehr, sondern ein zentraler Baustein moderner Safety-Architekturen.

## **Wie verändert KI das Safety-Engineering?**

KI verstehen wir in erster Linie als Assistenzsystem und nicht als autonomen Entscheider. Sie unterstützt beispielsweise bei der strukturierten Ableitung von Applikationscode aus Anforderungen, bei der Generierung von Testfällen oder bei Konsistenzprüfungen. Entscheidend ist, dass alle Ergebnisse nachvollziehbar und zertifizierbar bleiben. Richtig integriert kann KI den Engineering-Aufwand deutlich reduzieren und Projekte beschleunigen, ohne die normative Strenge funktionaler Sicherheit zu gefährden.