Rundum sicher

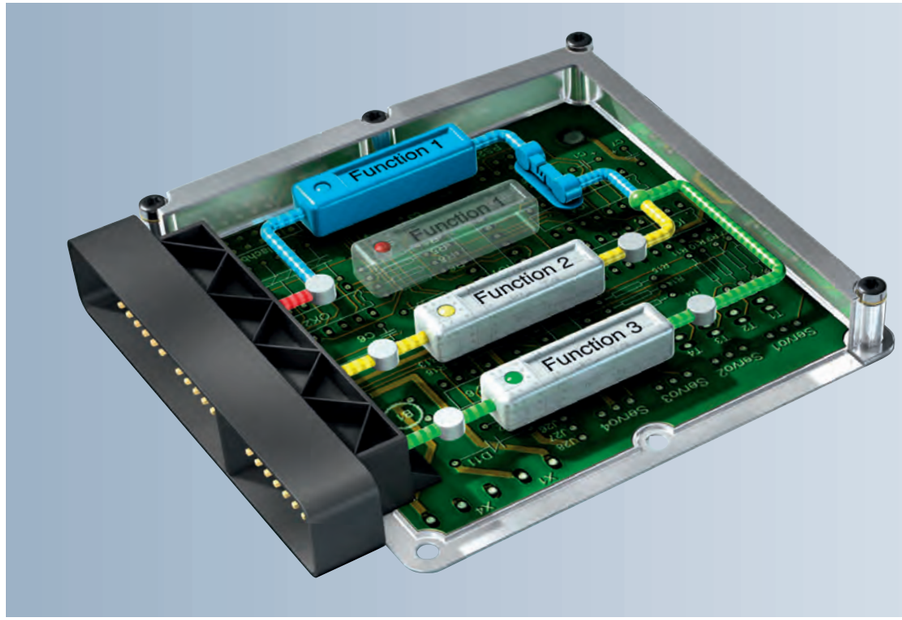

Bild 1:
Mit Freischnitt-Werkzeugen wie ETAS EHOOKS können gezielt Fehler eingeschleust werden, um die Robustheit des Systems zu testen.

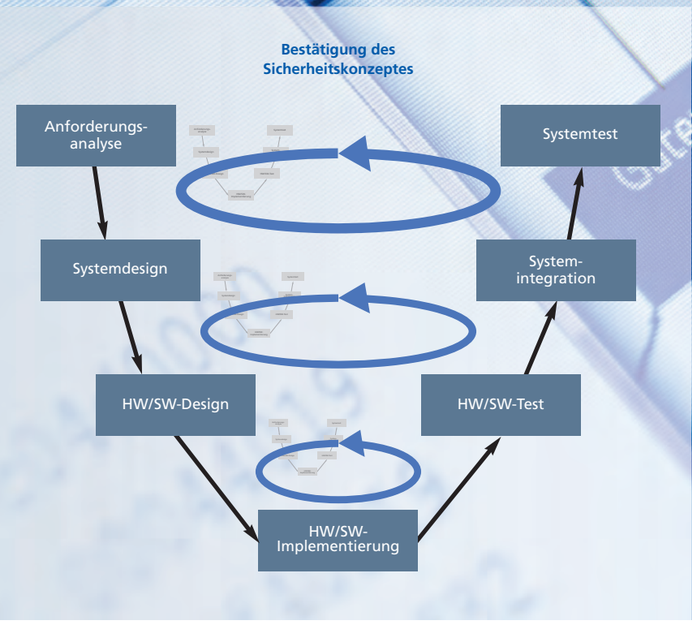

Bild 2 (Grafik):
Das Sicherheitskonzept muss im gesamten Entwicklungszyklus betrachtet werden – nur testen reicht nicht aus.

Konsequenter Entwicklungsprozess für funktional sichere Steuergeräte

Wo verteilte Entwicklerteams mehrerer Unternehmen am selben Steuergerät arbeiten, wird funktionale Sicherheit auch zu einer Frage der Organisation. Im Prozess gilt es, methodisches Know-how, bewährte Entwicklungswerkzeuge und fachliche Expertise konsequent zusammenzuführen.

AUTOR

Dr. Simon Burton ist Director Global Embedded Software Services bei der **ETAS GmbH**.

Mehr und mehr Funktionen im Fahrzeug werden durch Software in elektronischen Steuergeräten realisiert. Auch die steuergeräteübergreifende Verknüpfung von Softwarefunktionen nimmt stetig zu. In Modellen der Oberklasse sind es teils mehr als hundert Steuergeräte, die über Datenbusse vernetzt untereinander und mit den im Fahrzeug verteilten Sensoren kommunizieren. Zudem nimmt die Vernetzung nach außen rasch zu. Automobile werden zu Netzwerkknoten einer komplexen vernetzten Verkehrswelt.

Diese Vernetzung verspricht sicheren, effizienteren Verkehr. Doch sie birgt auch Risiken. Diese liegen unter anderem im immer komplexeren Entwicklungsprozess der vernetzten Steuergeräte, in den oft global verteilte Teams involviert sind. Entwickler und Applikateure von Steuergeräteherstellern, deren Zulieferer sowie Entwickler der Automobilhersteller wirken bei der Entwicklung der Steuergerätesoftware und Bedatung mit. So sind bei Motorsteuergeräten verschiedene Teams damit befasst, Einspritzung, Luftzufuhr, Zündzeitpunkte und weitere Parameter zu regeln. Zudem bestellen OEMs im Sinne der Liefersicherheit bei mehreren Zulieferern identische Steuergeräte, die für Endkunden nicht unterscheidbar sein dürfen. Unabhängig von ihrem Ursprung müssen diese Steuergeräte und ihre Software zuverlässig arbeiten. Sicherheitskritische Fehler sind tabu.

Sicherheit im kollaborativen Entwicklungsprozess

Wie lässt sich funktionale Sicherheit unter diesen Bedingungen gewährleisten? Noch dazu, wenn Softwarefunktionen auf mehrere Steuergeräte verteilt sind und nur Teile der Software ausgetauscht werden sollen? Ein Schlüssel zum Erfolg sind innovative Methoden und Werkzeuge. Etwa Virtualisierung, um die Software schon im Frühstadium in realistischen Strecken-, Umgebungs- und Fahrermodellen validieren zu können. Ein Werkzeug wie ISOLAR-EVE von ETAS erlaubt es, kombiniert mit PC-Simulation in solche Tests einzusteigen, lange bevor ein Steuergeräteprototyp vorliegt. So fallen Fehler oder Fehlannahmen auf, ehe sie Schaden anrichten können. Auch können Entwickler im virtuellen Umfeld gefahrlos Grenzbereiche ausloten, was gerade das Auslegen sicherheitskritischer Assistenzsysteme deutlich erleichtert.
Virtualisierung ist nur ein Baustein einer umfassenden Architektur. Sie muss den Entwicklungsprozess vorab strukturieren (Bild 2) und dabei klare Hierarchien in den Software-Architekturen und den mitwirkenden Teams definieren. Zudem ist enges Prozess-Monitoring unabdingbar. Regelmäßig sollten Assessments und Audits prüfen, ob die Teams die abgestimmte Sicherheitsphilosophie beherzigen und alle Akteure Vereinbartes gleich verstehen. Die Grundlagen einer solchen Entwicklungsstruktur finden sich in Normen wie der ISO 26262.

Etablierte Regeln für Software- und Entwicklungsprozesse

Die Vorbereitung der Software-Entwicklung startet mit der Definition des Entwicklungsumfangs. Im Top-Down-Ansatz wird ausgehend von Gesamtsystem und Kontext erst der Funktionsumfang der Schnittstellen und die Interaktion mit anderen Systemen eingegrenzt. Es folgt eine strukturierte Gefährdungs- und Risikoanalyse, um die Wahrscheinlichkeit und Kontrollierbarkeit etwaiger

Fehler zu gewichten und das drohende Schadensausmaß zu bewerten. Aus dieser eingehenden Analyse leiten sich verbindliche Sicherheitsziele ab. Sie sind fortan Richtschnur der Entwicklung – und müssen dafür zunächst auf konkrete Arbeitspakete für die Arbeitsgruppen heruntergebrochen werden.

Die Programmierung muss sich auf Regelwerke und bewährte Methoden stützen: Orientierung an Best Practice, Einhalten von Styleguides sowie die strukturierte Dokumentation und dezidierte Reviews und Code-Analysen. Zudem muss ein Testplan her: Wann und wie wird die Software in welchen Kontexten getestet? Unabdingbar sind Fault-Injection-Tests, in denen die Reaktion der Software auf gezielt eingebrachte Fehler abgeprüft wird. Dafür speisen Entwickler per Bypass-Tool, etwa EHOOKS von ETAS, fehlerhafte Daten ein. Anders als in früheren Tests mit ISOLAR-EVE geschieht das auf realer Hardware (Bild 1). Mit EHOOKS können Prüfingenieure Steuergeräte überlisten, indem sie deren interne Signale durch eingeschleuste fehlerhafte Signale ersetzen oder gezielt falsch bedaten. Als Schleuse dienen die Kalibrier- und Diagnoseschnittstellen des Steuergeräts. Die Fehlbedatung wird mit der Mess-, Applikations- und Diagnosesoftware ETAS INCA generiert.

Funktionen mit AUTOSAR sicher abgrenzen

In den so provozierten Ausnahmesituationen zeigt sich, ob sich Sicherheitsziele und Konzeption tragen. Robuste Software muss Fehler erkennen und den Betrieb des Steuergeräts aufrechterhalten oder in einen sicheren Zustand bringen. Doch es sind auch Vorkehrungen zu treffen, damit sich Fehler nicht ausbreiten. Dabei hilft der AUTOSAR-Standard mit der Memory Protection: Entwickler können Software per Hardware-Support am Zugriff auf die Speicher anderer Softwarefunktionen hindern. Fehler bleiben so lokal begrenzt und können keine sicherheitsrelevanten Software-Applikationen beeinträchtigen.

Um die AUTOSAR-Mechanismen Memory Protection oder Timing Protection zu integrieren, müssen aber alle Projektbeteiligten ihre Softwarekomponenten als Quell- oder Objekt-Code offenlegen. Da diese Transparenz oft nicht gegeben oder gewünscht ist, hat ETAS den Hypervisor RTA-HVR entwickelt. Dieser kann ein Steuergerät in mehrere strikt voneinander getrennte, virtuelle Steuergeräte partitionieren und sicherheitsrelevante Funktionen so komplett voneinander abschotten. Zwischen den Partitionen ist Kommunikation wie bei verschiedenen Steuergeräten über definierte Schnittstellen nach vorab festzulegenden Regeln möglich.

Technische und organisatorische Partitionierung

Die Partitionierung hat einen weiteren großen Vorteil: Teams der verschiedenen Unternehmen können unabhängig voneinander Software entwickeln, da diese ja später abgeschirmt auf ihrer Partition läuft. Gegenseitiges Einsehen der Codes im Frühstadium ist nicht nötig. Allerdings setzt das parallele Entwickeln voraus, dass der Steuergerätehersteller den Prozess koordiniert und steuert. Im Eigeninteresse muss er alle Akteure auf die Sicherheitsziele sowie einen verbindlichen Fahrplan einschwören. Dieser legt fest, wann Entwicklungspartner welche Nachweise für die funktionale Sicherheit erbringen müssen. Die Integration liegt dann beim Steuergerätehersteller. Er führt die bei seinen Zulieferern getesteten Softwarekomponenten samt Dokumentationen zusammen und erbringt letztlich den Gesamtnachweis der funktionalen Sicherheit.

Umsichtige Projektsteuerung

Zu den Steuerungsaufgaben gehört es, Vorannahmen ständig zu validieren, zu plausibilisieren – und wo nötig nachzujustieren. Auch müssen alle Partner stets den neuesten Stand kennen. Der Aufwand zahlt sich aus: Wo eine gemeinsame Sicherheitsphilosophie Richtschnur der Entwicklung ist, sinkt der Testaufwand und es bleiben kostspielige, nervenaufreibende Korrekturen in der Endphase der Entwicklung aus. Bei laxer Projektsteuerung drohen Probleme: Denn Audits und Assessments fördern oft unterschiedliche Interpretationen von Sicherheitszielen, Vorannahmen und Normen zu Tage, aus denen ohne gezieltes Gegensteuern teure Fehlerketten entspringen können. Aus diesem Grund sind Überprüfungen in der Sicherheitsnorm ISO 26262 fest verankert.

Fazit

Funktionale Sicherheit von Steuergerätesoftware lässt sich auch im komplexen kollaborativen Entwicklungsprozess gewährleisten. Das setzt aber mehr voraus als nur Expertise in der Steuergeräte-Entwicklung. Vielmehr gilt es, eine normgerechte Entwicklung und moderne Methoden wie die Validierung im virtuellen Umfeld in einen strukturierten, sorgfältig gesteuerten Entwicklungsprozess einzubetten. Mit Werkzeugen, Services und Consulting bietet ETAS für jede Projektphase die richtigen Lösungen.