DEN DRACHEN BÄNDIGEN

So wie in alten Mythen Drachen die Menschen plagten und ihr Leben bedrohten, stellen heute technisch unsichere Systeme eine Gefahrenquelle dar. Der entscheidende Unterschied: Während die Drachen in das Reich der Fabeln gehören, sind technische Systeme als potentielle Gefahrenquellen Teil unseres Alltags.

TEXT: MicroConsult **FOTOS:** MicroConsult www.eue24.net/PDF/EE811003

Wenn wir auf Embedded-Systeme treffen, was jedem von uns jeden Tag unzählige Male passiert, gehen wir davon aus, dass diese Systeme sicher sind. Aber was bedeutet Sicherheit überhaupt? Unterschieden wird grundsätzlich zwischen der so genannten Betriebssicherheit (Safety) und der Zugangssicherheit (Security). Während Safety Menschen beim Umgang mit einem System schützen soll, ist Security der Schutz vor Menschen, also dem Missbrauch und unbefugtem Zugriff. Wenn man das Beispiel einer Eingangstür nimmt, wird der Unterschied schnell deutlich: Ein Klemmschutz, der Verletzungen verhindern soll, oder ein Sensor, der bei Gefahr eine Drehtür anhält, fallen in den Bereich Safety. Eine Überwachungskamera hingegen oder die Zugangsbeschränkung per Fingerprint sind eindeutig der Security zuzuordnen.

Zunächst spielt die Betriebssicherheit oder auch technische Sicherheit für Entwickler die wichtigere Rolle. Die folgende Definition macht deutlich, dass es dabei um mehr geht als nur ein funktionierendes Produkt: „Unter ‚technischer Sicherheit' wird begrifflich verstanden, dass ein technisches System, eine Anlage, ein Produkt über einen geplanten Zeitraum (gegebenenfalls die geplante Lebensdauer) hinweg die vorgesehenen Funktionen erfüllt und bei bestimmungsgemäßer Nutzung keine geschützten Rechtsgüter verletzt, d. h. weder Personen noch Sachen geschädigt werden, soweit dafür das System, die Anlage oder das Produkt ursächlich sein können. Die Zuverlässigkeit der Funktion über die vorgesehene Lebensdauer ist kein notwendiger Bestandteil der Sicherheit, sofern der Verlust der Funktion zu keinem unsicheren Zustand führt." (Quelle: Qualitätsmerkmal, „Technische Sicherheit" – Eine Denkschrift des Vereins Deutscher Ingenieure, Düsseldorf, 2007) . Jedoch wirken die Security-Eigenschaften aufgrund der immer stärkeren Vernetzung quasi durch die Hintertür auch auf die Safety-Merkmale, z. B. wenn Software von Hackern oder Würmern wie Stuxnet manipuliert werden kann.

„In vielen Firmen ist das Ziel der Software immer noch, dass sie einfach nur läuft – und das ist viel zu wenig", erklärt Peter Siwon, Business Development Manager beim Schulungs- und Consulting-Spezialisten MicroConsult. „Das ist so als würde ich sagen: Hauptsache, das Auto fährt, und es ist mir egal, ob die Bremsen funktionieren oder der Scheibenwischer korrekt arbeitet." Dieter Volland, als Dozent für Softwareengineering ebenfalls bei MicroConsult tätig, ergänzt: „Aufgrund von Zeitdruck sind Entwickler viel zu oft zum Pfuschen gezwungen." Um im Bild zu bleiben: Wir ziehen einen feuerspeienden Drachen auf und nehmen uns keine Zeit ihm gute Manieren beizubringen, zum Beispiel keine Menschen anzuniesen.

Bedrohung von innen

Während bei Security die Bedrohung meist von außen kommt, werden Probleme in der Betriebssicherheit folgerichtig meist system-intern verursacht. Zu den größten Bedrohungen der Betriebssicherheit gehören ungetesteter Code, unerwünschte Wechselwirkungen in der Architek-

tur und vor allem die mangelnde Qualifikation der Menschen, die für die Betriebssicherheit zuständig sind. Das bedeutet nicht, dass ein Entwickler keinen Code schreiben kann – darin besteht aber auch nicht die Kunst betriebssicherer Software. Es fehlt vielmehr das Wissen, wie man Sicherheitsanforderungen formuliert und diese dann auch tatsächlich in die Software implementiert.

„Projektteams müssen frühzeitig dafür sensibilisiert werden, dass das Schreiben von Code im Grunde das Allerwenigste ist", führt Peter Siwon aus. „Sicherheit kommt dadurch zustande, dass ich in der Lage bin, bestimmte Architekturmerkmale umzusetzen und Regeln einzuhalten, die diese Sicherheit unterstützen." Dieses Wissen wird in der Ausbildung oft nicht ausreichend vermittelt – und ist Quereinsteigern meistens gar nicht bekannt. Auch Frank Listing, Dozent für Softwareengineering bei MicroConsult, sieht die unzureichende Ausbildung als Problem: „Ingenieure wie Elektrotechniker, Maschinenbauer oder Physiker sind fachlich meistens gut, aber bei der Software-Entwicklung gibt es oft Schwachstellen." Qualifikationsmängel sind auch der Grund für eine weitere Bedrohung: zugekaufte Software. Wer nicht in der Lage ist, betriebssichere Software zu entwickeln, kann auch bei zugelieferten Programmen nicht erkennen, ob die Sicherheitsanforderungen tatsächlich erfüllt werden. Dabei ist das Problem keinesfalls nur auf Entwickler-Ebene zu sehen. „Das Management hat die Safety oft nicht im Blick und meint, dieses als lästig empfundene Thema auf Externe abwälzen zu können", beschreibt Dr. Günter Glöe, Geschäftsführer von CATS Software Tools, die alltägliche Praxis. „Oft fehlt es den Projektbeteiligten an Überblick und Sensibilität", ergänzt Dieter Volland.

Normen bieten Sicherheit

Das mangelnde Bewusstsein für Safety ist umso erstaunlicher, wenn man die möglichen Konsequenzen bedenkt. Ein unzufriedener Kunde mag zu einem wirtschaftlichen Schaden führen. Auch das kann zu einem ernsthaften Problem werden, ist aber relativ harmlos im Vergleich zu einem Fall, bei dem aufgrund mangelnder Betriebssicherheit Menschen zu Schaden kommen. Hier muss auch der Entwickler mit drastischen Konsequenzen rechnen, wenn er Anforderungen nicht erfüllt hat. Umgekehrt macht es eine umfassende Dokumentation von Entwicklung und Test leicht, entsprechende Vorwürfe zu entkräften. „Entwicklern ist oft nicht bewusst, dass die Safety-Standards den Stand unserer Technik widerspiegeln und unsere Arbeit – wenn wir diesen Standards nicht gerecht werden – handwerklich mangelhaft ist", erklärt Dr. Günter Glöe.

Das Beispiel des Seilbahnunglücks von Kaprun macht dies deutlich: Bei dem Unglück im November 2000 kamen 155 Menschen beim Brand einer Seilbahnkabine ums Leben. Alle Angeklagten wurden freigesprochen, da die Entwicklung und Fertigung nach dem Stand der Technik erfolgte und kein Verstoß gegen gültige Normen festzustellen war. Wichtig war hier natürlich der Nachweis, dass die geltenden Normen eingehalten wurden. Nichts anderes bedeutet letztlich Stand der Technik: Bestehende Normen müssen angewandt und nachweislich erfüllt sein. Und diese Normen geben allen Beteiligten Sicherheit: dem Kunden und dem Entwickler. Da Normen schwer interpretierbar und für Entwickler oft auch nicht zu verstehen sind, ist es wichtig, die entsprechenden Auditoren frühzeitig in den Entwicklungsprozess einzubeziehen. „Sicherheitsstandards sind Kochrezepte für Elektronik und Software, die dem Stand der Technik entsprechen", führt Günter Glöe aus. „So wie man Kochrezepte für die eigene Küche anpassen muss, muss man auch die Standards an den eigenen Bedarf anpassen." Und eine absolute Fehlerfreiheit ist nahezu illusorisch. „Zero Defects sind zwar wünschenswert, aber physikalisch nicht zu erreichen", erklärt Prof. Dr. Jürgen Mottok, Scientific Head of Laboratory for Safe and Secure Systems in Regensburg. „Die Betriebssicherheit eines technischen Systems versteht sich als die Reduktion des Risikos auf ein (wirtschaftlich) vertretbares Maß. Damit verbleiben tolerierbare Restfehler in unseren technischen Systemen. Geeignete Fehlererkennung und -reaktionen müssen umgesetzt werden."

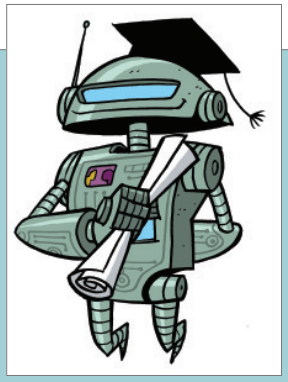
Sicherheit als Prozess

Aber wie kann man als Entwickler sichere und qualitativ gute Software entwickeln? Zunächst einmal muss man sich darüber im Klaren sein, dass Sicherheit und Qualität keine Zufallsprodukte sind – sie sind das Ergebnis gezielter Maßnahmen während des gesamten Entwicklungsprozesses. Man kann nicht am Ende des Prozesses Qualität hineintesten. Aufgrund der geforderten Sicherheitsstufe sollten in jeder Phase des Projekts (z. B. Anforderungsanalyse, Design, Implementierung, Unit-Test, Integrationstest, Systemtest, Inbetriebnahme) geeignete Maßnahmen ergriffen werden, die einerseits die Erfüllung der Qualitätsanforderungen sicherstellen und andererseits durch geschickte Kombination dazu dienen, den Gesamtaufwand zu minimieren. „In fast allen Projekten ist die fehlende Verzahnung der Teststufen ein Problem", erläutert Christian Nachreiner, Geschäftsführer Intence Automotive Electronics. „Dadurch können weiße Flecken in der Testabdeckung entstehen, die in einem Safety-Projekt nicht akzeptabel sind. Abhilfe schafft ein projektübergreifendes Testkonzept. Leider ist in vielen Unternehmen keine Rolle in der Entwick-

lung besetzt, welche die Tests auf allen Ebenen von der Implementierung bis zum Prüflabor betrachtet.“ Auch Dr. Günter Glöe sieht in mangelnder Planung des Testens eines der größten Probleme: „Ingenieurstätigkeiten unterscheiden sich vom Basteln auch dadurch, dass sie geplant ablaufen. Die Planung der Prüfarbeiten muss parallel mit der Entwicklung des Arbeitsproduktes laufen, gegen das geprüft wird.“

„Sicherheit und andere Qualitätmerkmale müssen bereits Bestandteil des Systementwurfs sein“, so Peter Siwon. „Das fängt bei der Anforderungsanalyse an. Es geht weiter beim Design, wo geprüft werden muss, ob die Architektur überhaupt die Sicherheitsanforderungen abbilden kann. Es gibt Designs, bei denen man an der Struktur erkennen kann, dass die Sicherheitsanforderungen nicht erfüllbar sind.“ Auch bei der Implementierung müssen bestimmte Regeln eingehalten werden, um die Sicherheit zu garantieren. Es ist sinnvoll, in jeder Prozessphase darüber nachzudenken, wie man die Sicherheitsanforderungen erfüllbar machen kann. In vielen Fällen wird zu sehr implementierungsorientiert gedacht. Phasenorientierte Prüfschritte verzögern zwar die Implementierung, reduzieren aber gleichzeitig den Aufwand für große Korrekturschleifen, die sich durch den gesamten Entwicklungsprozess ziehen. Man sollte den Drachen nicht erst versuchen zu bändigen, wenn er bereits groß und stark ist. „Im Bezug auf Sicherheit und Qualität hinkt die Software-Entwicklung noch hinter der Hardware-Entwicklung her“, so Peter Siwon. „Viele Entwickler glauben, man könne in der Software schnell mal etwas ändern, aber dafür sind viele Systeme bereits zu komplex.“

Aber die Integration in den Prozess alleine reicht auch noch nicht aus: Jeder der am Entwicklungsprozess Beteiligten muss sich seiner Rolle und seiner Aufgaben bewusst sein – und diese auch akzeptieren. Dazu gehört auch die Übernahme von Verantwortung. Das setzt eine klare Definition der Rollen und eine klare Verteilung der Verantwortung voraus. „Wollen, Kön-



SEMINARANGEBOT VON MICROCONSULT

Software-Projektmanagement: Erfolgreiches Führen von Projektteams durch alle Projektphasen
Diese Grundlagen für professionelles Projektmanagement sind eine wichtige Voraussetzung für die Erreichung gesteckter Qualitäts- und Sicherheitsziele.

Software-Qualität für Embedded Systeme: Umsetzung wichtiger Normen und Standards durch die Anwendung bewährter Prozessmodelle und Methoden
Hier werden Antworten auf folgende Fragen gegeben: Welche Normen und Standards sind zu beachten? Wie lassen sie sich klar definieren? Wie kommen sie praktisch ins Produkt? Wie lassen sie sich nachweisen?

Sicherheit (Safety) von Elektronik und deren Software
Dieses Training schafft die Voraussetzungen, dass alle Projektbeteiligten in der Lage sind, die Anforderungen an die Sicherheit von Elektronik und deren Software gemäß dem Stand der Technik zu erfüllen.

Software-Test: Strukturiertes und effizientes Testen von Embedded-Systemen
Sie lernen in Theorie und Praxis alle Phasen des Software-Testprozesses kennen, von der Test-Anforderungsanalyse bis zum Abnahmetest. Sie erfahren dabei das Wichtigste über bewährte Methoden zur Planung, Spezifizierung, Durchführung, Dokumentation und Auswertung von Tests.

In dem Kontext von Qualität und Sicherheit spielen außerdem rechtliche Gesichtspunkte eine wichtige Rolle, die in folgenden Seminaren behandelt werden:

Embedded-Software: Wem gehört sie und welche Haftungsrisiken bestehen?
Die Teilnehmer verstehen anhand anschaulicher Beispiele aus der Praxis die typischen rechtlichen Haftungsrisiken. Sie können die grundsätzliche juristische Denkweise des Gesetzgebers und der Gerichte nachvollziehen und daraus eigenständig Schlüsse ziehen.

Open Source Software: Worauf ist rechtlich zu achten?
Die Teilnehmer lernen die rechtlichen Strukturen kennen, die im Zusammenhang mit dem Einsatz von Open Source Software im eigenen Unternehmen oder gegenüber Kunden wichtig sind.

Maßgeschneiderte Dienstleistungen für Sicherheit und Qualität
Zu allen Themen bieten wir auch maßgeschneiderte Trainings, Workshops, Projektberatungen und Projektcoachings an, die sich an Ihrer Projektsituation und Aufgabenstellung orientieren.

Weitere Informationen finden Sie unter www.microconsult.de

nen und Dürfen sind notwendige Kriterien für die Akzeptanz von Aufgaben und Verantwortung", erläutert Peter Siwon. „Und diese Kriterien lassen sich am besten durch eine professionelle Ausbildung und regelmäßige Information aller Beteiligten (Entwickler und Management) erfüllen." Spezialisten wie MicroConsult bieten für alle Phasen und Rollen des Entwicklungsprozesses geeignete Maßnahmen (Seminare, Workshops, Beratung, Projektcoaching, Kongresse) an, mit denen Unternehmen die Risiken reduzieren und so Zeit und Geld sparen können. Die Verantwortlichen gewinnen dabei an Selbstbewusstsein und fachlicher Sicherheit – und haben die Gelegenheit, sich mit Spezialisten aus anderen Unternehmen auszutauschen. Dieser Blick über den Tellerrand bringt wertvolle Erkenntnisse und Anregungen für die eigene Projektarbeit .

Das richtige Bewusstsein

Die Themen Qualität und Sicherheit ziehen sich wie ein roter Faden durch den gesamten Entwicklungsprozess. In jedem Prozessschritt gibt es sinnvolle Maßnahmen, um Qualität abzusichern und Anforderungen an die Sicherheit zu überprüfen, zu validieren oder zu testen. Der erste Schritt zu sicherer und qualitativ hochwertiger Software ist die Schaffung des Bewusstseins ihrer Notwendigkeit für den nachhaltigen Projekterfolg.

Nur wer sich der Bedeutung bewusst ist, wird sich mit der Problematik im Entwicklungsprozess auseinandersetzen und geeignete Aktivitäten starten. Eine der wichtigsten und ersten Maßnahmen besteht dabei darin, alle Projektbeteiligten ihrer Rolle und Verantwortung gemäß sinnvoll zu qualifizieren und zu informieren. Das garantiert zwar noch keine sichere Software, aber immerhin bestehen sehr gute Chancen, die Drachen so professionell zu zähmen, dass sie uns künftig viel Freude bereiten. □

Weitere Tipps der in diesem Artikel benannten Experten sowie Fachartikel und Literaturtipps finden Sie unter http://www.microconsult.de/service/

> MORE@CLICK EE811003