Sichere Software in Medizinprodukten

Durch ein effektives Risikomanagement können Prozesse in der Software-Entwicklung optimiert werden

Stephan Blab und
Wolfgang Weber

Da jede Software von Menschen gemacht wird und jeder Mensch Fehler machen kann, wird jede Software Fehler enthalten können. Wenn aber über jede Software zu sagen ist, dass sie Fehler enthalten könne, kann es „sichere Software" dann überhaupt geben? Schon nach den Denkgesetzen lautet die Antwort: „Nein."

Das Regelwerk ISO 14971, das die Anwendung des Risikomanagements auf Medizinprodukte betrifft, umschifft dieses Dilemma, indem es Sicherheit definiert als „Freiheit von unvertretbaren Risiken" und Risiko als „Kombination der Wahrscheinlichkeit des Auftretens eines Schadens und des Schweregrades dieses Schadens". Software-Sicherheit bedeutet demnach, dass alle Risiken bekannt und im vertretbaren Rahmen beherrscht sind, die bei der Verwendung der Software auftreten könnten. Die Software-Entwicklung kombiniert eine produktorientierte mit einer prozessorientierten Methode. Das Endprodukt soll die, an es gestellten sicherheitsbezogenen, Anforderungen erfüllen, indem es gemäß einem definierten Prozess entwickelt und getestet wird. Dieser Entwicklungsprozess wiederum muss die, an ihn gestellten sicherheitsbezogenen, Anforderungen erfüllen. Die heute oft präferierten agilen Methoden erfüllen nur schwer diese Anforderungen. Deswegen hat sich das V-Modell als traditionelle Methode bewährt. Es eignet sich zur Anwendung, weil es neben den Entwicklungsphasen auch die Testphasen berücksichtigt.

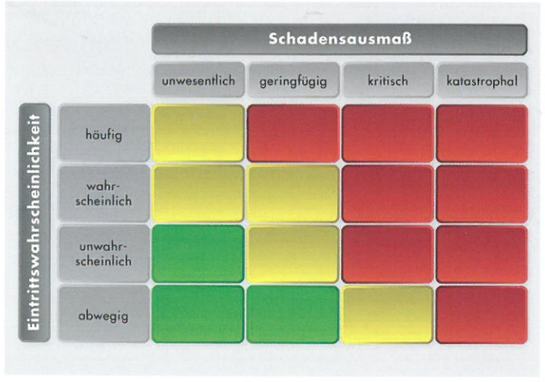
Eigenes Risikomanagement für jede Komponente

Entscheidend für die Effektivität eines produkt- und prozessorientierten Entwicklungsprozesses sind zwei miteinander verflochtene Ansätze:

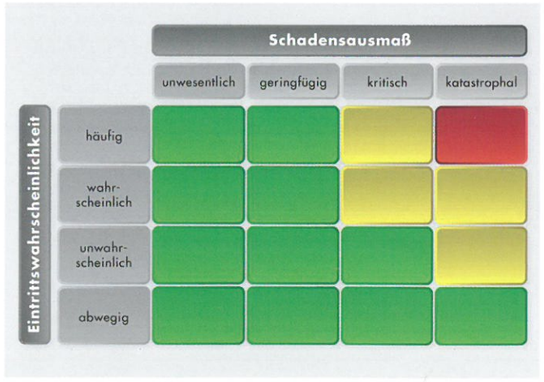
Erstens ist die Software in Komponenten zu untergliedern, die so gekapselt sind, dass sie unabhängig voneinander vollständig testbar sind. Sie werden so entwickelt, dass Komponententests und Funktionstest (OQ) zeitlich nahezu parallel laufen und dadurch bei Abschluss des Entwicklungsprozesses auch die Funktionsqualifizierung – zuletzt nur noch als Schnittstellentest – abgeschlossen werden kann.

Zweitens ist für jede dieser Komponenten ein eigenes Risikomanagement gemäß ISO 14971 durchzuführen.

Risikobewertungsmatrix für Software zur Steuerung einer Insulinpumpe



Risikobewertungsmatrix für Software zur Chargenverfolgung bei der Produktion von Mullbinden



„Gemäß" wohlgemerkt, denn was in der Praxis häufig Anwendung findet, ist eine Bewertungsmethodik, die gerade nicht die Maßgaben der Norm verwirklicht. Vielmehr wird – im Namen gemutmaßter Effizienz – eine Vereinheitlichung vorgenommen, die beispielsweise eine Software zur Chargenverfolgung bei der Produktion von Mullbinden und eine Software zur Steuerung einer Insulinpumpe gleich behandeln würde. Aus Eintrittswahrscheinlichkeit und Schadensausmaß werden für alle Produkte dieselben Risikostufen abgeleitet, dieselben Größenordnungen zugrunde gelegt. Ob aber eine Wahrscheinlichkeit von $1:10^3$ „häufig" und $1:10^6$ „unwahrscheinlich", wie in ISO 14971 nur als Beispiel angeführt, repräsentativ für die jeweiligen Risikoszenarien ist, hängt stets vom Produkt und von der Definition des Anwendungsfalls im Lebenszyklus ab. Ist jeder Programmstart (der nicht funktionieren könnte) ein Anwendungsfall einer Software, die eine Million Anwender zehnmal täglich starten sollen – und das über zwei Jahre hinweg? Oder ist einer von 10.000 Anwendern ein Anwendungsfall im Jahr?

Risikobewertung individuell zuschneiden – ein Heftpflaster ist kein Herzschrittmacher

Es ist beliebt, eine symmetrische Matrix mit den immer gleichen vier Stufen 1, 3, 7 und 10 aufzuspannen, aus denen Risikoprioritätszahlen (von 1x1 = Mindestwert bis 10x10 = Höchstwert) errechnet werden, daraus wiederum drei Stufen „niedrig", „mittel", „hoch". Aber besteht bei als „hoch" eingestuftem Risiko wirklich Gefahr für Leib und Leben eines Patienten oder vielleicht nur die Aussicht, dass er mit dem Produkt unzufrieden sein könnte? Ein mechanistisches Vorgehen bei der Einstufung führt somit zu einem ineffektiven Risikomanagement: Hohe Risiken können als zu niedrig bewertet werden, umgekehrt können Testdichte und -tiefe viel aufwendiger geraten als nötig.

Fazit

Software-Entwicklungsprozesse lassen sich normenkonform optimieren. Geeignetes Kapseln und Testen der Komponenten samt ihrer Schnittstellen und individuelles Zuschneiden der Risikobewertungsmatrizen durch einen fachlich angemessen besetzten Sachverständigenkreis ermöglichen dies. Das kostet zu Beginn mehr personelle Ressourcen, aber amortisiert sich rasch, weil das Testen genau am individuellen Bedarf des Produktes ausgerichtet wird. ■

Mehr Informationen zum zertifizierten V-Modell® XT Projekt- und QS-Verantwortlichen V-Modell finden Sie unter: *www.isqi.org/de/zertifikate.html*

Stephan Blab ist Leiter des Kompetenzcenters Prozessqualifizierung der EXCO GmbH.

Wolfgang Weber ist Leiter der Außenstelle Jena der EXCO GmbH.