

Sonderdruck aus
Automotive Electronics 2001

Dr. Helmut Schelling
Dr. Joachim Fetzer
Dr. Meinhard Erben

Software- Komponenten -

Ein neuer Trend in der
Automobilelektronik



Software-Komponenten – Ein neuer Trend in der Automobilelektronik

Weltweite Rückrufaktionen von Automobilen und spektakuläre Produkthaftungsfälle in den USA können rasch zu Forderungen in dreistelliger Millionenhöhe führen. Oft ist der Automobilhersteller nicht selbst für den Defekt verantwortlich. Die Kosten fallen dann beim Zulieferer an. Die hohe Summe würde jedoch nahezu jedes mittelständische Unternehmen wirtschaftlich vernichten. Viele Software-Häuser leben dennoch mit diesem Risiko. Dieser Beitrag der Vector Informatik GmbH zeigt mögliche Lösungsansätze des latent vorhandenen Problems auf.

1 Einleitung

Ein Blick auf das letzte Jahrzehnt zeigt, dass nahezu alle wichtigen Innovationen in der Kfz-Technik durch softwarebasierte elektronische Systeme getragen worden sind. Dies wird verdeutlicht, wenn man die Entwicklung und den Grad der Vernetzung in Kraftfahrzeugen betrachtet. Seit der Serieneinführung des CAN-Protokolls 1989, mit drei Busteilnehmern, ist die Anzahl heute auf bis über 40 Steuergeräte angestiegen. Damit einher ging auch eine zunehmende Verteilung der immer komplexeren Funktionalität. Wurden anfänglich die an sich autonom arbeitenden Komponenten wie Motormanagement und Getriebe verbunden, sind heute die einzelnen Funktionen auf viele Steuergeräte verteilt.

Diese Entwicklung wird sich in Zukunft weiter beschleunigt fortsetzen. Dafür sorgen steigende Ansprüche an den Komfort und die Zuverlässigkeit eines Kraftfahrzeugs sowie die Forderungen nach geringer Umweltbelastung und reduziertem Kraftstoffverbrauch. Eine wesentliche Voraussetzung für erfolgreiche Innovationen ist dabei die Fähigkeit, die Software kostengünstig sowie mit hoher Produktivität und hoher Qualität zu entwickeln.

2 Ausgangssituation

Die in den Steuergeräten implementierte Software kann man in die sogenannte Infrastruktur-Software – also Betriebssystem, Kommunikations-Software wie zum Beispiel CAN-Netzwerkanbindung [1] – und grundlegende Ein-/Ausgabedienste sowie die Applikations-Software, die eigentliche gewünschte Funktionalität, trennen. Für die Applikations-Software zeichnet sich als neuer Trend ab, dass diese, im Gegensatz zu heute, entweder vom Automobilhersteller selbst oder von Software-Zulieferern bereit gestellt wird, die nicht auch gleichzeitig die Steuergeräte-Hardware liefern. Ein Beispiel für eine solche Trennung ist die in **Bild 1** dargestellte Software-Architektur. In diesem Beispiel sind Software-Teile von sechs Firmen zur Gesamt-Software integriert:

- Hardware-Zulieferer: Hardware und Basis-I/O-Funktionen.
- Betriebssystem-Hersteller: OSEK-Betriebssystem.
- Zulieferer Software-Komponenten: Anbindung an die verschiedenen Netzwerke CAN, LIN.
- Software-Zulieferer 1: Wesentlicher Teil der Applikation und Integrator des Steuergerätesystems.

Die Autoren



RA Dr. Meinhard Erben ist bundesweit als selbstständiger Rechtsanwalt tätig. Er beschäftigt sich ausschließlich mit IT-Recht.



Dr.-Ing. Joachim Fetzer ist Geschäftsführer der Vector-Consulting GmbH. e-mail: joachim.fetzer@vector-consulting.de



Dr.-Ing. Helmut Schelling ist Geschäftsführer und verantwortlich für den Bereich Applikation und Vernetzung für Kraftfahrzeugsteuergeräte der Vector Informatik GmbH.

■ Kfz-Hersteller: Stellt eine selbst entwickelte, wettbewerbsrelevante Teilfunktion her.

■ Software-Zulieferer 2: Spezialist für Teilfunktion A.

Da die Elemente der Infrastruktur-Software keine Wettbewerbsdifferenzierung für die Automobilhersteller beinhalten, können sie als kostengünstige Gleichteile von allen Automobilherstellern gleichermaßen eingesetzt werden. Die Verwendung von Software-Komponenten im Rahmen eines architekturbasierten Software-Designs erlaubt dem Kfz-Hersteller oder Zulieferer eine einfachere Portabilität und Wiederverwendung der eigentlichen Applikation auf verschiedenen Steuergeräten und führt damit zu einer erheblichen Senkung der Entwicklungskosten.

Eine solche arbeitsteilige Entwicklung bringt sowohl aus technischer Sicht – Experten lösen jeweils Teilaufgaben – wie auch in Bezug auf die Entwicklungskosten – nicht wettbewerbsrelevante Teile werden nur einmal entwickelt – entscheidende Vorteile bei den Kosten und der Qualität. Bisher wurden diese Vorteile unter Hinweis auf unlösbare juristische Haftungsprobleme jedoch oft nicht genutzt.

Neben den technischen Herausforderungen bei der Integration der einzelnen Softwarebestandteile stellt sich deshalb die Frage, welcher der Beteiligten wofür die rechtliche Verantwortung übernimmt. Man denke dabei nur an die Kosten einer Rückrufaktion oder die in den USA verhängten „punitive damages“ bei einem Produkthaftungsfall.

Als Hersteller des Endprodukts ist zunächst einmal der Automobilhersteller gegenüber dem Fahrzeugkäufer in der Verantwortung. Dieser trägt die Schäden aus einer Rückrufaktion sowie den unmittelbaren Sach- und/oder Personenschaden. In vielen Fällen ist allerdings nicht der Automobilhersteller selbst für den Defekt verantwortlich, sondern der Hersteller des Einzelprodukts, also der Zulieferer. Dieser haftet dann dem Kfz-Hersteller aus dem mit diesem geschlossenen Vertrag – der Kfz-Hersteller stellt also Regressansprüche gegen den Zulieferer.

Dazu ein Beispiel: Der Kfz-Hersteller wird in den USA aufgrund eines Produkthaftungsfalls verklagt und zur Zahlung von US \$ 10.000.000 an tatsächlich entstandenem Schaden verurteilt (zum Beispiel Totalschaden am Auto: Sachschaden,

Summary

Software Components – A New Trend in the Automotive Industry

A look at the past decade shows that almost all important innovations in automotive technology have been carried by software based electronic systems. This is illustrated by the development and the degree of network use in cars. In 1989 the CAN-protocol was introduced into production cars, which then was used to connect three modules. Now the total amount of electronic control units has risen to over 40 pieces. This was accompanied by an increasing distribution of the even more complex functionality. In the first applications only modules have been connected, that can also work in an autonomous manner (e.g. engine management, and transmission control). Now functions have been spread across many modules.

This development will be continued with more speed in the future. This is due to increasing demands for convenience and reliability of a car as well as the requirements for minor pollution and reduced fuel consumption. An essential condition for successful innovations in this context is the capability to develop software at a reasonable price as well as with high productivity and high quality.

2 Ausgangssituation



Bild 1: Beispiel für die Software-Architektur eines Steuergeräts bei dem verschiedene Entwicklungspartner Teile der Software beistellen

Tod des Fahrzeugführers und schwerwiegende Verletzung der Gattin nebst Berufs- und Erwerbsunfähigkeit: Personenschaden und Vermögensschaden als Folge des Personenschadens, Unterhalts- und Halbweisenansprüche der drei Hinterbliebenen: Reiner Vermögensschaden).

Weiterhin werde der Kfz-Hersteller zur Zahlung von US \$ 50.000.000 als „punitive damages“ verurteilt, weil er ein so gefährliches Produkt in den Verkehr brachte (das ist die Rechtfertigung für diese Ansprüche nach US-amerikanischem Recht).

Der Kfz-Hersteller nimmt den für den Defekt verantwortlichen Zulieferer in Regress: Dieser haftet nicht auf die „punitive damages“ in voller Höhe, sondern nur entsprechend seiner Leistungsfähigkeit, weil es nicht Sinn und Zweck der „punitive damages“ ist, das Unternehmen wirtschaftlich zu vernichten, sondern „nur“, es zu bestrafen (daher der Name: Strafschadensersatzansprüche).

Bild 2 zeigt beispielhaft die Rechtsansprüche und die vertraglichen Beziehungen zwischen den Beteiligten. Neben den Regressansprüchen gibt es den Fall, dass der Fehler entdeckt wird, bevor es zu einem Unfall aufgrund des fehlerhaften Produkts kommt. Dann ist eine – möglicherweise weltweite – Rückrufaktion geboten. Man kann sich aufgrund der gefertigten Stückzahlen die Kosten dafür ausrechnen. Der dadurch entstehende (reine) Vermögensschaden erreicht nicht selten einen dreistelligen Millionenbetrag. Eine Summe, die nahezu jedes mittelständische Unternehmen wirtschaftlich vernichten oder dazu nötigen würde, sich – dann wohl zu einem läppischen Preis – vom Kfz-Hersteller aufkaufen zu lassen.

Viele Software-Häuser leben mit diesem Risiko. Im Folgenden werden mögliche Lösungsansätze für Kfz-Hersteller und Software-Zulieferer am Beispiel der Software-Komponenten diskutiert. Die Ausführungen können analog auf die Applikations-Software übertragen werden.

3 Software-Komponenten

Wer Software-Komponenten für Embedded-Anwendungen im Kraftfahrzeug herstellt, erscheint zwar auf den ersten Blick als ein gewöhnlicher Zulieferer. Er begibt sich aber in eine juristisch sehr schwierige Situation. Er liefert Software-Komponenten, wie OSEK-Betriebssysteme und Software zur Anbindung von CAN-Controllern üblicherweise als Quellprogramm an die Anwender, die damit ihre spezifischen Aufgaben erfüllen.

Die Software-Komponenten bestehen aus einer großen Zahl von Software-Routi-

2 Ausgangssituation



Bild 2: Beispielhafte Darstellung der wechselseitigen Rechtsansprüche und vertraglichen Beziehungen zwischen den Beteiligten

nen, die über vielfältige Konfigurationsparameter möglichst optimal für die Zielanwendung angepasst werden können. Dadurch ergibt sich eine so große Zahl an möglichen Varianten, dass diese nicht alle durch den Software-Komponenten-Hersteller vor der Auslieferung verifiziert und validiert werden können. Der Testaufwand für alle möglichen Varianten strengen den Kosten- und Zeitrahmen sprengen.

Erschwerend kommt hinzu, dass auch Fehler aufgrund falscher Parametrierung entstehen können. Außerdem werden oft die im Quelltext vorliegenden Software-Komponenten vom Anwender direkt modifiziert, damit dieser seine speziellen Anforderungen erfüllen kann.

Haftungsrechtlich hat dies zur Folge, dass genau geprüft werden muss, ob ein aufgetretener Defekt durch eine Änderung und/oder Ergänzung des Quellcodes oder die Parametrierung seitens des Anwenders verursacht wurde. Weiterhin ist zu prüfen, ob der Defekt vielleicht a priori in den ausgelieferten Software-Komponenten bereits vorhanden war.

Es liegt auf der Hand, dass diese beson-

dere Situation rasch zu schwer auflösbaren Zuordnungsproblemen führen kann. Dies gilt insbesondere vor dem Hintergrund, dass zum einen die Steuergeräte über verschiedene Bussysteme gekoppelt sind und zum anderen die Applikationssoftware auf mehrere Steuergeräte verteilt sein kann (sogenannte verteilte Steuergerätearchitektur) [2].

4 Versicherbarkeit der Risiken

Der typische Kfz-Zulieferer verkauft seine Produkte auf Stückzahlbasis. Software-Komponenten werden dagegen derzeit nicht auf der Basis einer Stückzahl-Lizenz verkauft, sondern zu einem einmaligen Festpreis. Dabei wird keine Stückzahlbegrenzung festgelegt, die Software-Komponenten können also vom Käufer beliebig oft in Steuergeräten verwendet werden. Eine Limitierung in der Praxis ergibt sich nur aus der Tatsache, dass die Software-Komponenten für einen bestimmten Prozessortyp entwickelt wurden und nur auf diesem lauffähig sind. Dadurch ist es dem Hersteller der Software-Komponenten

nicht möglich, a priori anzugeben, mit welcher Stückzahl sein Produkt im Markt eingesetzt werden wird. Außerdem kennt er die potenziellen Risiken der einzelnen Steuergerätfunktionen nicht, er weiß also nicht, ob die Software-Komponenten zum Beispiel in einem Steuergerät für eine elektrohydraulische Bremse oder in einem Fensterhebersteuergerät eingesetzt werden.

Da die Versicherer aber das Produkthaftungsrisiko nur adäquat kalkulieren können, wenn sie die Stückzahlen und die konkreten Risiken wissen, ist es sehr schwierig, einen Versicherer zu finden, der bereit wäre, das Produkthaftungsrisiko einschließlich reiner Vermögensschäden – also insbesondere derjenigen, die aufgrund einer Rückrufaktion entstehen – mit einer festen jährlichen Prämie zu versichern.

5 Problemlösung

Im Folgenden soll nun erläutert werden, wie diese Probleme durch einen kombinierten technischen und juristischen Ansatz optimal gelöst werden können. Im Mittelpunkt steht dabei eine Entwicklung der Software-Komponenten, die den Anforderungen nach hoher Produktivität, hoher Qualität und, als Konsequenz daraus, nach möglichst geringen Kosten entspricht.

5.1 Juristische Komponente – Haftungsbegrenzung

Der Hersteller von Software-Komponenten ist typischerweise ein kleineres mittelständisches Unternehmen. Würde der Kfz-Hersteller den Hersteller der Software-Komponenten bei einem von diesem verschuldeten Fehler, der zum Beispiel zu einer Rückrufaktion führt, in vollem Umfang in Regress nehmen, müsste der Software-Hersteller Insolvenz anmelden. Dies würde den längerfristigen Interessen der Anwender nicht entsprechen, da dadurch wichtiges Know-how verloren ginge und kostspielige Verzögerungen bei einigen Entwicklungsprojekten entstünden.

Ein problemgerechter Lösungsansatz ist, dass der Kfz-Hersteller oder -Zulieferer und der Hersteller der Software-Komponenten eine angemessene Teilung des Haftungsrisikos vereinbaren, soweit dem Hersteller der Software-Komponenten nicht grobe Fahrlässigkeit oder Vorsatz zur Last fällt und soweit nicht die Haftpflichtversicherung des Software-Herstellers die Schäden deckt. Dabei wird der Hersteller der Software-Komponenten eine be-

5.2 Technische Komponente – Qualitätsmanagement

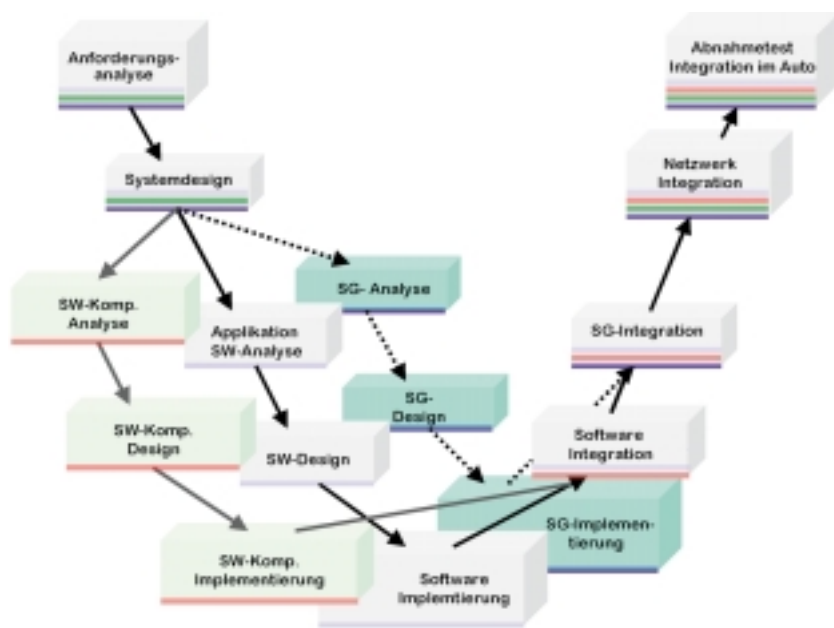


Bild 3: Beispielhafte Darstellung eines durchgängigen herstellerübergreifenden Entwicklungsprozesses auf der Basis des V-Modells. Die farbigen Querstriche referenzieren die verschiedenen Entwicklungspartner (siehe auch Bild 1 + 2)

stimmte vertraglich festgelegte Haftungssumme übernehmen, unabhängig davon, ob sein Haftpflichtversicherer darüber hinaus eintritt.

Für den Kfz-Hersteller oder -Zulieferer entstehen durch die Verwendung von Software-Komponenten keine besonderen neuen Probleme. Das Hinzutreten eines weiteren Lieferanten, des Software-Komponenten-Herstellers, führt nicht einmal dazu, dass der Kfz-Hersteller oder -Zulieferer das dem Versicherer als Risikohöherung anzeigen müsste oder dass er deshalb eine höhere Versicherungsprämie zahlen müsste. Für Kfz-Hersteller oder Kfz-Zulieferer ändert sich also durch das Hinzutreten des Software-Komponenten-Herstellers in versicherungstechnischer Hinsicht nichts.

Dieser Lösungsansatz hat für die Automobilindustrie sogar einen besonderen Charme: Die Automobilhersteller und/oder Kfz-Zulieferer erhalten die Software-Komponenten zu einem vergleichsweise günstigen, da nicht von hohen Versicherungsprämien belasteten Preis und erwerben daran ein Nutzungsrecht für eine unbegrenzte Anzahl von Automobilen.

5.2 Technische Komponente – Qualitätsmanagement

Neben dem verantwortungsvollen juristischen Umgang mit dem Haftungsrisiko steht natürlich die Vermeidung von Feh-

lern durch geeignete Qualitätsmanagementmaßnahmen im Vordergrund, die in einem ganzheitlichen Ansatz den Hersteller und die Anwender der Software-Komponenten umfassen. Hier sind aufgrund der arbeitsteiligen Entwicklung weitergehende Maßnahmen erforderlich.

Grundlegende Voraussetzung für eine hohe Qualität ist ein etabliertes prozessorientiertes Qualitätsmanagementsystem beim Hersteller der Software-Komponenten mit dem Ziel der beständigen Verbesserung und Optimierung der Qualität des Entwicklungsprozesses. Grundlage für die Prozessverbesserung – also für die kontinuierliche Erhöhung des Reifegrads der Organisation – sind regelmäßige interne und externe Prozess-Assessments, die sowohl der Reifegradbestimmung des Prozesses als auch zum Aufzeigen von möglichen Prozessverbesserungen dienen.

Als Modelle können hierfür das Capability Maturity Model (CMM) [3] oder das Assessment-Modell der ISO-Norm 15504 (SPICE) [4] verwendet werden. Beide Methoden erlauben durch die Definition von Referenzmodellen eine widerspruchsfreie, zuverlässige und reproduzierbare Prozessbewertung. Besondere Bedeutung haben in diesem Zusammenhang regelmäßige Prozess-Assessments beim Hersteller, die von den Anwendern der Software-Komponenten selbst durchgeführt werden. Dadurch wird sichergestellt, dass

der Hersteller der Software-Komponenten jederzeit nachprüfbar Qualität liefert und der Anwender in den Entwicklungsprozess der Software-Komponenten integriert ist. Die Anwender müssen mit dem Software-Komponenten-Hersteller sehr eng zusammenarbeiten und einen durchgängigen herstellerübergreifenden Entwicklungsprozess etablieren, **Bild 3**.

Die Mitarbeit des Software-Komponenten-Herstellers ist in den Phasen von der Software-Integration bis zum Abnahmetest unabdingbar. Dies betrifft zum Beispiel die vom Anwender gewählte Parametrierung und eventuell vorgenommene Programmänderungen. Diese sind zur Erreichung einer hohen Produktqualität vom Hersteller der Software-Komponenten zu verifizieren und zu validieren. Bei der Erstellung der Testspezifikation für ein Steuergerät ist die Unterstützung durch den Software-Komponenten-Hersteller ebenfalls sehr wichtig. Nur er kann aufgrund seines Wissens über den inneren Aufbau der Software-Komponenten wirkliche „Stresstests“ für zum Beispiel die CAN-Anbindung des Steuergerätes formulieren.

Der Automobilhersteller ist für die Integration und die Validierung des Gesamtfahrzeugs verantwortlich. Eine enge Mitarbeit des Software-Komponenten-Herstellers bei diesen Arbeiten ermöglicht durch dessen spezifisches Wissen eine Beschleunigung der Arbeiten und eine höhere Testtiefe des Systems.

6 Schlussfolgerungen

Der Einsatz von nicht wettbewerbsdifferenzierenden Software-Komponenten ermöglicht bei entsprechender Verteilung des Haftungsrisikos der Automobilindustrie eine kostengünstige, schnelle und erfolgreiche Realisierung von Innovationen. Durch eine enge Zusammenarbeit aller Partner, durchgängige herstellerübergreifende Entwicklungsprozesse und regelmäßige Prozess-Assessments durch die Anwender beim Software-Komponenten Hersteller kann das Fehler- und damit Haftungsrisiko für den Anwender der Software-Komponenten gegenüber einer Eigenentwicklung erheblich reduziert werden.

Vorteilhaft für die Qualität der Software-Komponenten ist auch eine hohe Testtiefe. Diese ergibt sich durch die Konzentration des Software-Komponenten-Herstellers auf wenige Software-Produkte und des damit verbundenen Aufbaus von spezifischem Know-how sowie des breiten Einsatzes der Software-Komponenten bei vielen Anwendern. ■

Literaturhinweise

- [1] <http://www.vector-informatik.de/deutsch/index.html>
- [2] Riegraf, T.: Verteilte Systeme im Automobil, Elektronik Automotive, WEKA Fachzeitschriften-Verlag, Mai 2001
- [3] <http://www.sei.cmu.edu/managing/managing.html>
- [4] <http://www.sqi.gu.edu.au/spice/>