

Software Entwicklung in der Automobilindustrie – Strategie für die Zukunft

Dr. Elmar Paul Selbach · Leiter Automotive Practice · HCL Technologies GmbH

Wie überall, hat die Mikroelektronik auch in der Automobilwelt in den letzten 10 Jahren rasant an Bedeutung zugelegt: Bereiche wie Sicherheitselektronik (ABS, ASR), Diagnose, Multimedia, Telematik (Navigationssysteme) und vor allem Motorsteuerung haben die Fahrzeuge mit zu den Systemen mit dem höchsten Anteil an Mikrocontroller-Steuerungen gemacht.

Und diese Mikrocontroller werden natürlich durch Software gesteuert. Das heißt, diese Hardware kann noch so gut sein, letztendlich bestimmt die Qualität der Embedded Control Software, ob ein System funktioniert oder nicht.

Das bekannte Bonmot der Softwareentwickler lautet: „Es gibt keine fehlerfreie Software – Software, von der keine Fehler bekannt sind, ist einfach nicht im Einsatz“. Leider scheint dieser Satz zu stimmen, denn die massiven Probleme und Rückrufaktionen der Automobilhersteller gehen zum größten Teil auf das Konto der Steuerungselektronik. Dabei stehen die großen OEM und Zulieferer wie Bosch oder Visteon erst am Anfang: der Anteil an Software im Automobil – so groß er schon ist – wird weiter stark wachsen und anteilmäßig im Jahr 2010 mehr als 40% der Herstellungskosten eines Autos ausmachen, wenn man die heutigen Entwicklungsprozesse und Standards als Massstab nimmt. Mehr als 90% der Innovationen und 70% der Entwicklungskosten werden dann auf Elektronik und Steuergeräte-Software entfallen und der Software Umfang in Steuergeräten verdoppelt sich alle 2-3 Jahre (*zitiert nach H.G. Frischkorn, BMW AG, aus der Zeitschrift: Automotive Electronics*).

Dazu kommt, dass auch der Speicherbedarf exponentiell ansteigt, auch dieser verdoppelt sich also alle 2 Jahre, das heisst, das Moor'sche Gesetz gilt in der Steuerungselektronik nicht nur für die angebotene Leistung, auch der vorhandene Speicher wird sofort ausgenutzt – de facto eine Mangelwirtschaft im Memory Management.

Es ist also nur eine Frage der Zeit, bis Software zum hauptsächlichen Begrenzungsfaktor automobiler Innovation wird. Darüber hinaus besteht auch die Gefahr, dass bei der bereits erreichten Leistungsfähigkeit der Segen der Elektronik in das Gegenteil umkippt, was von kybernetischen Systemen am Rand ihrer Leistungsfähigkeit ein bekanntes Phänomen ist. Spätestens dann, wenn Software als Grundursache für Vorfälle identifiziert wird, die Menschen in Gefahr bringt, Geschäftsprozesse und Transaktionen stört und massive

volkswirtschaftliche Störungen hervorruft, wird unser ganzes Wirtschaftssystem unter der jetzt noch relativ kleinen Fehlerquelle zu leiden haben

Das ist nicht von der Hand zu weisen und auch nicht sonderlich spekulativ: In der Raumfahrt ist schon lange bekannt, dass kleinere Probleme sich zum Totalausfall hochschaukeln können und auch Automobilentwicklern ist das Problem zumindest in Testphasen schon begegnet. Der Flughafen in Denver in USA konnte Ende der Achtziger Jahre schon nicht in Betrieb gehen, weil die vielen einzelnen Softwaresysteme nicht ineinander griffen und auch das Maut-Debakel ist bekannt – unzureichend getestete Gesamtsysteme spielten nicht zusammen. Denn Fehler haben nun mal die scheußliche Eigenschaft, sich nicht additiv sondern multiplikativ fortzusetzen. Es sind eben kybernetische Systeme (*wem das Wort „kybernetisch“ nicht gefällt: das heutige Wort dazu ist öfter auch „systemisch“*)

In der Luft- und Raumfahrttechnik war das Problem schon lange manifest und daher kamen auch die ersten Bemühungen, der sogenannten „Software-Krise“ Herr zu werden. Das war bereits 1968, als in den USA dazu erste Forschungen anliefen, die zum Begriff „Software-Engineering“ führten.

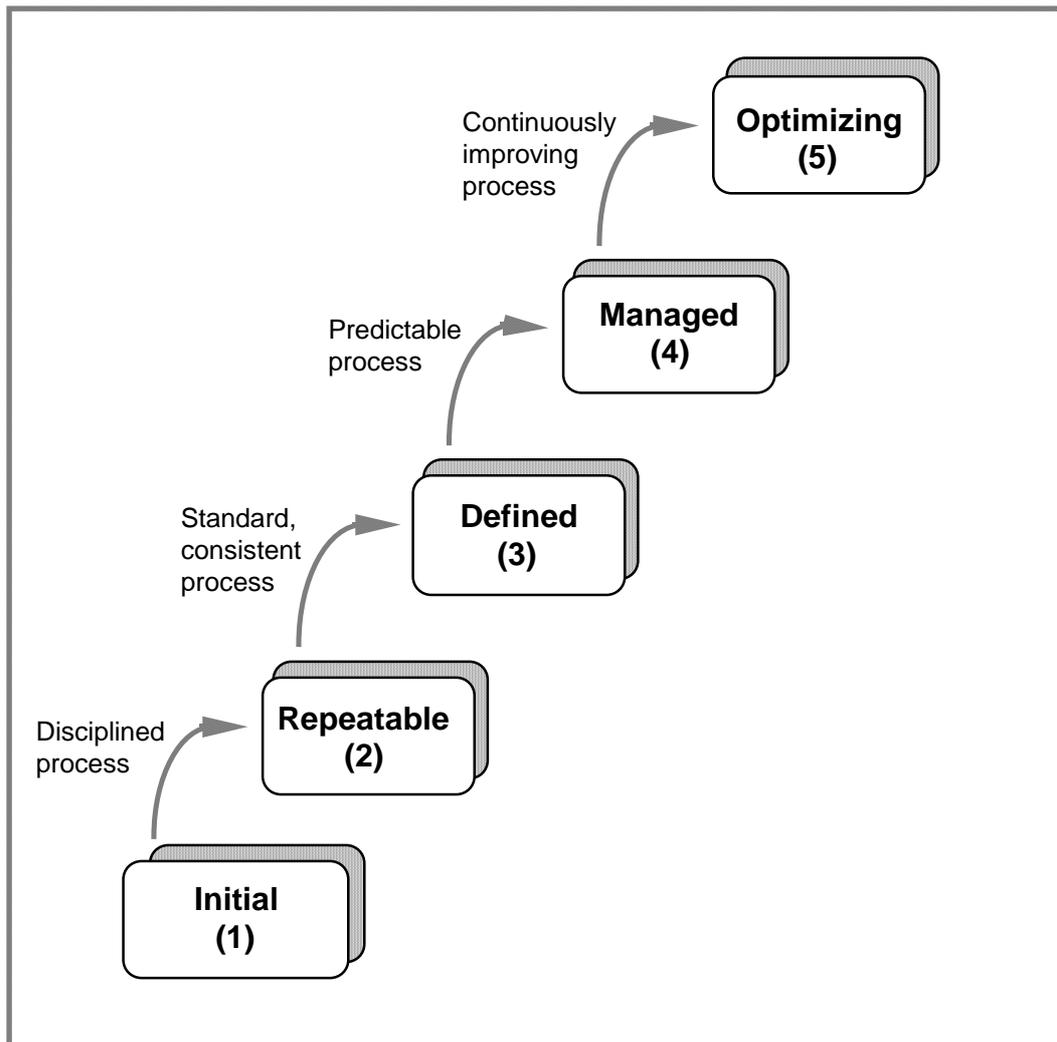
Software Engineering: mehr als Entwicklung

Dieses Fach ist seit vielen Jahren schon Ingenieurdisziplin und wird weltweit als Grundlage für die wissenschaftliche Vorgehensweise bei Software- und Systementwicklung angesehen.

Führend in diesem Bereich ist seit mehr als 10 Jahren die Carnegie-Mellon Universität in den USA, deren Software-Engineering Institut den sogenannten CMM Standard etablierte. CMM steht für „Capability-Maturity-Modell“ und ist ein Prozeßmodell, das den „Reifegrad“ einer Vorgehensweise beschreibt.

Dieses Modell ist mittlerweile zu einem Standard herangewachsen, der einen eigenen Zertifizierungsprozeß benötigt. In Deutschland ist man hier noch sehr zurückhaltend, was unverständlich ist, da gerade in der Automobiltechnik ein dringender Bedarf besteht, die Prozesse in den Griff zu bekommen und das angesammelte Wissen zu integrieren, das sonst in Fachabteilungen eingesperrt bleibt.

⇒ Die derzeitige Situation in der automobilen Softwareentwicklung wirft Fragen nach dem Risikomanagement und dem Qualitätsmanagement der der Projekte auf.



Definition der Stufen des CMM Modells (nach SEI Institut der Carnegie Mellon Universität)

Bei den Standards und Prozessen der Software Entwicklung kann man ansetzen, will man nicht in eine Kosten- und Qualitätsfalle laufen. Damit schafft man sich wenigstens Luft, um in der eigentlichen Programmierung neue Methoden und Tools definieren zu können.

Qualität setzt Wissen um die Prozesse voraus

Die deutschen Hersteller von Software für Automobile, im wesentlichen die OEM selbst, haben als Antwort auf die derzeitige Krise die Herstellerinitiative Software (HIS) gegründet. Damit werden alle Lieferungen der Hersteller von Software auf ihre Kompatibilität mit ISO 15504 geprüft. Diese Norm ist auch als SPICE-Modell bekannt. SPICE ist eine zu CMMI kompatible Vorgehensweise der Qualitätssicherung in der Softwareentwicklung.

Dieser Weg ist sicher richtig, wenn man ihn denn konsequent unter dem Ziel sieht, eine einheitliche Wissens- und Qualitätsbasis zu schaffen, aber danach sieht es derzeit nicht aus. Die einzelnen Firmen und Branchen liegen im Verständnis der

HCL Technologies GmbH - Automotive Center - Am Söldnermoos 17 - D85399 München-Hallbergmoos - Tel.: 089-607-68671 - Fax: 089-607-68679 - Mail: elmar.selbach@hclt.de

zugrundeliegenden Prozesse und Werkzeuge und sogar im Verständnis der Anforderungen noch weit von einander entfernt. Nur wenige Schlüssellieferanten haben ein Gesamtkonzept für Entwicklungs- und Qualitätsstandards sowie übergreifende Collaboration Konzepte (die auch das Risikomanagement mit einschliessen).

Dazu müsste noch ein einheitliches Verständnis und Werkzeuge für Tests, wie Performance-, Last- und Funktionstests, der Gesamtsysteme kommen – da tun sich große Lücken auf.

Des weiteren – und das könnte sich zum Kardinalproblem von HIS auswachsen – ist der Test der Software projektspezifisch und nicht übergreifend, wie bei einem Standard wie CMMI. Dieser würde wenigstens eine einheitliche - und nicht eine projektspezifische - Basis vorgeben, an der sich alle Zulieferer orientieren könnten.

Etliche große Systemlieferanten sind mittlerweile (nicht nur aus Kostengründen) nach Indien ausgewichen. So läßt Siemens-VDO seine Qualitäts- und Testinitiativen in Bangalore überwachen und dort holt sich die Firma das benötigte Know-How, um die angestrebte Zertifizierung nach CMM-3 zu erreichen. (*Nach: Stefan Hohrein: Automotive Electronics September 2004*). Mit mehr als 70% nach CMM 5 (der höchsten Stufe) auditierter Firmen stellt Indien nämlich auch einen riesigen Brainpool zur Verfügung.

Nur CMM gibt bisher die Möglichkeit, einen lückenlos nachvollziehbaren und begründeten Prozess bereit zu stellen.

Wenn mehr als 50% aller Fahrzeugausfälle (*ADAC Pannenstatistik*) mittlerweile auf Elektronik-Fehler zurück gehen, was soll dann erst werden, wenn sich der Softwareanteil im Auto nochmals vervierfacht?

Aus Sicht eines Theoretikers sieht die Situation nicht so aus, als könnte mit den getroffenen Maßnahmen rechtzeitig gegen gesteuert werden – die Gefahr besteht, dass HIS ins Leere läuft, weil hier offensichtlich zu stark auf die Auditierung der bereits gelieferten Software fokussiert wird – an den Wurzeln packt man da nicht an. Es erscheint auch fraglich, ob hier eine einheitliche Wissens-Datenbasis und vor allem, ein Knowledge-Management Tool geschaffen werden kann, mit dem die Software und die Prozesse lückenlos dokumentiert werden.

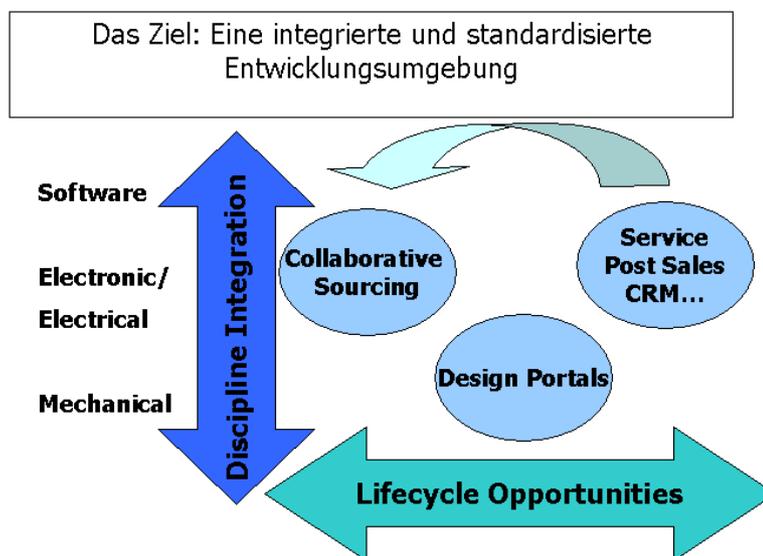
Viele Prozesse – viele Komponenten – viele Fehlermöglichkeiten

Weitere Initiativen wie AUTOSAR, ein Gremium, das sich mit der Software Architektur befasst, sowie der die Komponenten adressierende Standard OSEK unterstützen auf der Ablauebene. Es kann aber sein, dass die Ingenieure bei der Integration der diversen Ansätze (die verschiedene Ebenen adressieren) am Ende überfordert sind – denn es gilt, mindestens 4 verschiedene Schnittstellen und Bussysteme (Flex-Ray, CAN, LIN, MOST) nach den obigen Vorgaben zu implementieren, zu auditieren und vollständig zu testen. Der letzte Punkt ist kritisch, denn vollständige Tools, die Gesamtsysteme simulieren und testen sind offensichtlich noch nicht vollständig ausgereift.

Vorhandene IT-Tools bieten begrenzten Nutzen

Die großen Herausforderungen kommen also erst noch. Ohne Zusammenarbeit aller Hersteller von Autos und der Zulieferer auf hohem Niveau kommen die bisherigen Bemühungen schnell an eine Grenze. Meines Erachtens verlassen sich Ingenieure auch zu sehr auf das, was ihnen an Tools von (meist US) Herstellern vorgesetzt wird. Das betrifft mehr oder weniger die übergreifenden IT-Werkzeuge oder Tools für Product-Lifecycle-Management..

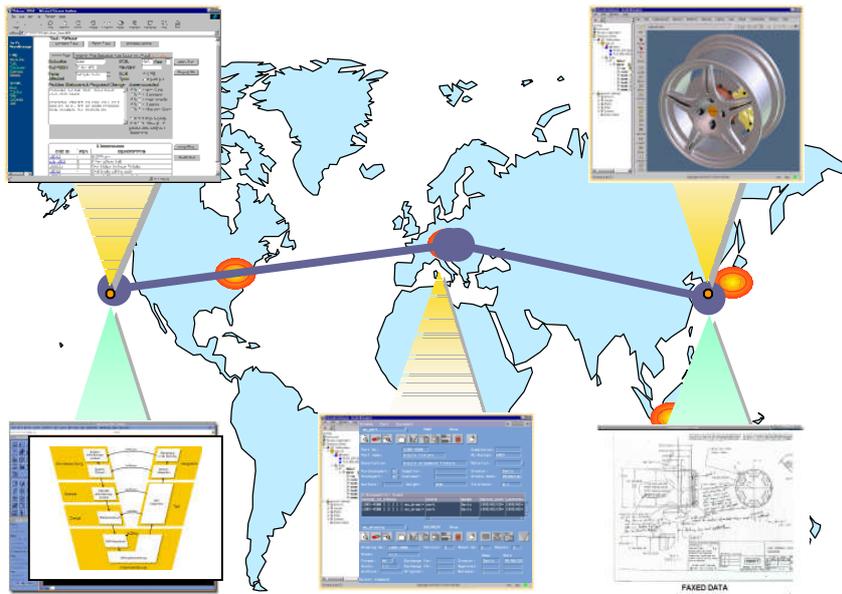
Es kommt immer wieder vor, dass Software Hersteller von Tools diese ohne Rücksprache ändern, mit unzuverlässigen Patches arbeiten oder generell am Bedarf der Autohersteller vorbei entwickeln.



Ingenieure und Informatiker sind zu einem großen Teil ihrer Zeit damit beschäftigt, mit den Unzulänglichkeiten der existierenden Systeme fertig zu werden – und haben noch kaum ein Gefühl dafür, was mit den neuen Prozeßstandards und Schnittstellen (z.B. in STEP) auf sie zu kommt.

Aus: Die neuen Werkzeuge der Produktentwicklung
Vortrag E.P. Selbach
Automotive Harness Forum 2004

Das nächste offene Thema – ebenfalls unzureichend adressiert- ist Engineering Collaboration. Das ist das Schlagwort für die weltweite Zusammenarbeit heterogener Entwicklungsteams an einem Gesamtsystem. Auch für diesen Prozeß benötigt man Standards, Vorgaben und ein gemeinsames Verständnis. Das wird vor



Werkbild: CoCreate

allein die Ingenieurebenen einschliessen, aber später auch die gesamte Lieferkette und den Lieferzyklus integrieren. (Schlagwort *Product Lifecycle Management*)

Ingenieurdisziplinen wie Electronic Design, Software und Konstruktion/Mechanik werden enger verzahnt sein, so daß die Grenzen fließend werden. Dazu gehört

auch, dass ein Großteil der Basisentwicklung aus Billiglohnländern kommen wird – begleitet von hohen Anforderungen an die technische und vor allem auch soziale Kompetenz der Entwickler (wenn man unterstellt, dass die Sprachkenntnisse bereits befriedigend sind...)

Die Anforderungen sind also bereits hoch, sie werden aber, wie gezeigt, extrem steigen, und das nicht nur linear, sondern exponentiell.

Die Lösung kann nicht von den Entwicklern alleine kommen – das ist ein strategisches Thema

Deshalb wäre es sinnvoll, die ingenieurmäßigen Anstrengungen von strategischen Initiativen, die auch politisch fundiert sein müssten (z.B. durch Einbindung der EU Kommission) als Chefsache zu behandeln

Programme zur Kostensenkung sind in den meisten Unternehmen Chefsache, selbst eine SAP-Einführung oder CRM und SRM Entwicklungen laufen zumindest auf der Ebene des CFO - häufig sogar des Firmenvorstands. Derartige Programme adressieren den Bereich der indirekten Kosten, der für die Bilanz, verglichen mit dem Potential einer Umsatzsteigerung, relativ geringe Bedeutung hat. Im positiven wie im negativen Sinne wird die Entwicklung der Elektronik und die Einbindung der Werkzeuge und Prozesse in die gesamte Informationstechnik den größten Einfluß auf den wirtschaftlichen Erfolg der Autokonzerne und ihrer Zulieferer haben.

Steigen die Fehlerquoten mit der Zunahme der Software, ist die automobiler Zukunft Deutschlands in Gefahr. Bringt die Software aber den gewünschten Effekt und reduziert man gleichzeitig die Ausfallquoten, bringt das den größten Gewinn.

Strategische Initiativen im Bereich der Softwareentwicklung für Elektronik gehen häufig auf das Konto der Entwicklungsabteilung, deren Ressourcen bei weitem nicht

ausreichen, die Tool Hersteller auf der Top-Ebene zu beeinflussen, es sei denn, es handelt sich um lokale Anbieter von Nischenprodukten.

OEM und Zulieferer würden sich selbst einen großen Gefallen tun, wenn sie die Entwicklungsabteilungen hier entlasten und dem Thema die gebührende Aufmerksamkeit widmen würden.

Man darf nicht außer acht lassen, dass einige Autokonzerne insbesondere japanischer Provenienz Qualität mit dem Verzicht auf ausgefuchste Elektronik erkaufen. In einigen Jahren werden diese nachziehen müssen und dann möglicherweise mit den gleichen Problemen zu kämpfen haben, die heute bei deutschen Herstellern auftauchen. (Wahrscheinlicher ist aber, dass sie die Erfahrung der lokalen OEM ausnutzen und vermeiden werden, dieselben Fehler nochmals zu machen).

Makroökonomische Randbedingungen

Das Problem hat noch eine weitere Dimension. Der VDI beklagt nun schon seit längerer Zeit, dass die Zahl der Absolventen in ingenieur- und naturwissenschaftlichen Fächern dramatisch sinkt. Es ist abzusehen, dass genau dann, wenn der Software Anteil im Auto seinen höchsten Stand erreicht und die Anforderungen an die Entwickler aus jetziger Sicht ein Maximum erreicht haben, noch weniger Ingenieure, Informatiker, Mathematiker, Physiker, Chemiker usw. zur Verfügung stehen.

Das Problem ist also nicht nur aus Sicht der Entwicklung sondern aus einer ganzheitlichen Betrachtung zu sehen. Hier ist auf makroökonomischer Ebene der nächste Stolperstein zu finden.

Es sind derzeit nicht genug Entwickler verfügbar und es ist abzusehen, dass es in einigen Jahren zu dramatischen Engpässen bei den Ressourcen kommen wird.

Nun beschäftigen indische Entwicklungsunternehmen mittlerweile eine große Zahl an Entwicklern, die auch bereits lange Erfahrung in Automobilprojekten gewonnen haben. Indien selbst macht derzeit mit seiner strategischen CAR Initiative einen großen Sprung, um State-Of-the-Art Entwicklungskapazitäten für weltweite Aktivitäten verfügbar zu machen.

All das Entwickler Know-How, was Deutschland dringend benötigt, ist dort vorhanden.

Jetzt ist die Zeit, das alles zu adressieren und die notwendigen Schritte einzuleiten.

Unter den gegebenen Umständen wäre es also sinnvoll, sich dort das Wissen zu holen, wo es bereits verfügbar ist und wo auch die Märkte hin tendieren. Und das ist in diesem Fall Indien. Indische Dienstleister haben uns im Bereich Software-Prozesse und Qualitätsmanagement ebenso längst überholt, wie im Bereich Knowledge Management, auch wenn es hierzulande vielen schwer fällt, dies zu akzeptieren.

Eine Zusammenarbeit mit indischen Dienstleistern, die Erfahrung im Automobilsektor haben, stellt demnach die Zukunftsfähigkeit der deutschen Automobilwirtschaft sicher, wenn sie professionell aufgesetzt wird.