

Kostenfalle Software

Durch Plattformentwicklung zu beherrschbaren Entwicklungskosten

Seit vielen Jahren gilt Elektronik mit Anteilen bis zu 90% als Innovationstreiber Nummer 1 in der Automobilindustrie, Software trägt dazu als elementarer Funktionsträger elektronischer Systeme den größten Teil bei. Probleme bei der Softwareentwicklung führten jedoch in der Vergangenheit zu einer Reihe von Qualitätsmängeln. Hier haben sowohl OEM als auch Zulieferer unter anderem mit Prozessverbesserungsprogrammen (Stichwort: Automotive SPICE) reagiert. In der aktuell dramatischen Situation der Automobilindustrie, in der sich Zulieferer wachsendem Kostendruck durch OEM und einbrechenden Umsätze gegenüber sehen, müssen jetzt die Entwicklungskosten nachhaltig gesenkt werden. Der richtige Weg hierzu heißt Wiederverwendung durch Plattformen, vor allem im Softwarebereich.

Referenzen für Plattformentwicklung in der Softwareentwicklung

SIEMENS VDO
A u t o m o t i v e

TRW
Automotive



Ausweg aus der Kostenfalle in der Softwareentwicklung

Immer noch werden Synergien in der Softwareentwicklung bei den Zulieferern (Tier1) zu wenig genutzt, Wiederverwendung nicht systematisch betrieben oder unnötige Komplexität und damit Kosten durch Mehrfachentwicklung gleicher Funktionalität erzeugt. Nachdem bisher vor allem versucht wurde, die Qualitätskosten durch konsequente Prozessanwendung und damit Vermeidung von teurer Nacharbeit in der Entwicklung zu senken, muss in der Zukunft das Augenmerk auf der Senkung der Entwicklungskosten durch Wiederverwendung als Plattformkonzept und evolutionärer Weiterentwicklung liegen.

- Welche unterschiedlichen Ansätze zur Wiederverwendbarkeit von Software existieren?
- Welchen Umfang sollte die Wiederverwendung von Software haben, um maximalen Nutzen zu erreichen?
- Welche organisatorischen Maßnahmen sind notwendig sind, um den Erfolg einer Plattformstrategie zu sichern?

Unterschiedliche Ansätze zur Wiederverwendbarkeit von Software

Fragt man namhafte Tier 1 nach ihren Konzepten zum Software "re-use", verweisen sie wird in der Regel auf existierende Plattformen, die einen hohen Wiederverwendungsgrad ermöglichen. Untersucht man diese Konzepte näher, stellt man fest, dass damit häufig ein so genanntes "carry-over" Konzept gemeint ist, bei dem neue Entwicklungen auf bereits bestehende Entwicklungen für andere OEMs oder andere Fahrzeugplattformen desselben OEM aufsetzen. Dieser Ansatz ist jedoch von einem echten Plattformkonzept, das maximalen Nutzen bietet, zu unterscheiden:

Tabelle 1: Charakteristika eines echten Plattformansatzes

Charakteristika eines "carry-over" Ansatzes	Charakteristika eines echten Plattformansatzes
Design wird durch aktuelle Kundenanforderungen vorgegeben	Design wird durch Wiederverwendbarkeitsaspekte vorgegeben, pilotiert durch "Lead"-Kunde
Das zuletzt verifizierte Design ist Grundlage für das nächste neue Design	Plattform existiert unabhängig vom Projekt, oft gibt es eine generische Version zum Zwecke der Validierung
Prozesse für Verwendung oder Ableitung sind nicht definiert, ad-hoc Entscheidungen	Starke Prozesse für Plattformentwicklung ("road map") und an der Prozessschnittstelle Applikation / Plattform
Software ist nicht oder wenig konfigurierbar	Tool-basierte Konfigurierbarkeit individueller Kundenanforderungen
Arbeitsprodukte für "carry-over" sind nur der Software Code, nicht jedoch andere Arbeitsprodukte etwa aus Spezifikation oder Test	Plattform-Arbeitsprodukte decken das komplette V-Modell ab
Keine kundenunabhängige Plattformorganisation / kein Plattformbudget	Dedizierte Plattformorganisation, z. B. finanziert durch interne Lizenzgebühren der Kundenprojekte
Fehler ("bugs") werden in jedem Kundenprojekt einzeln gefunden	Fehler werden einmalig in der Plattform beseitigt und die Lösung wird an laufende und neue Projekte verteilt
Weiterentwicklungen kommen nicht allen Folgeprojekten zugute, „carry-over“ endet in disparaten Projektverästelungen	Kontinuierliche Weiterentwicklung der Plattform durch neue (innovative / repräsentative) Kundenanforderungen

Umfang Wiederverwendung von Software

Softwareplattformen dürfen sich nicht auf den reinen Code beschränken, sondern umfassen idealerweise den gesamten Entwicklungszyklus. Dies bedeutet, dass neben den eigentlichen Softwaremodulen auch Software- und Schnittstellenspezifikationen, Softwaredesigns und die zur Softwareverifikation notwendigen Arbeitsergebnisse („work products“), also Modul-, Integrations-, und Software-Testpläne sowie Testprozeduren und Testfallspezifikationen wieder verwendbar sein müssen.

Beim Einsatz moderner Werkzeuge für Spezifikationserstellung, Design, ggf. Codegenerierung und Softwaretest muss dies auch berücksichtigt werden, um z.B. möglichst einfach kundenspezifische Werte (etwa Schwellwerte bei Diagnose) einführen zu können, ohne bei Ableitung der Plattform zu einem konkreten Projekt Inkonsistenzen zu erzeugen oder die „Traceability“ (d. h. die durchgängige Verfolgbarkeit der Anforderungen in Design, Code und Test und umgekehrt) zu verlieren.

Plattformkonforme Organisation und Prozesse

Der Hauptfehler bei der Implementierung von Softwareplattformen ist die unzureichende Ausgestaltung von Organisation und Prozessen, um dem Plattformgedanken Rechnung zu tragen. In vielen Fällen gibt es gar keine oder nur eine schwache Organisationseinheit, die für Plattformentwicklung verantwortlich ist. Oft werden deren Ressourcen als „Steinbruch“ für aktuelle Krisen in Applikationsprojekten missbraucht.

Der Grund dafür liegt wohl in erster Linie in der Befürchtung der Unternehmen, dass eine starke (und damit teure) Plattformorganisation Kosten nicht direkt auf Kundenprojekte umlegen kann und somit nicht mit den klassischen Mitteln des Projektcontrollings steuerbar ist. Allerdings muss man zugeben, dass diese Furcht nicht ganz unbegründet ist: Es gibt Beispiele teurer Plattformentwicklungen, die die in sie gesetzten Erwartungen kaum erfüllt haben. Diese Ursachen sind vielfältig: zu hohe kundenspezifische Anteile, die technologische Entwicklung überholt das Plattfordesign, fehlende Stabilität oder Modularität, unzureichende Konfigurierbarkeit oder komplexe, kaum handhabare Integration mit dem Applikationsanteil der Software

Zur Vermeidung dieser Probleme muss der Aufbau einer Plattformorganisation übergreifend geplant und mit Unterstützung des Top-Managements erfolgen.

In einem ersten Schritt werden die Anteile der Software mit dem höchsten Grad an Gemeinsamkeit über verschiedenen Kunden identifiziert. Dies umfasst typischerweise mindestens den bei AUTOSAR „Basic Software“ genannten Umfang, sollte aber deutlich darüber hinausgehen. Für die Entwicklung und Pflege dieser Software inklusive der dazugehörigen Arbeitsprodukte für

Anforderungen, Design und Test wird eine dedizierte Plattformorganisation aufgebaut.

Diese Plattformorganisation folgt – neben den Standardprozessen z. B. gemäß Automotive SPICE – weiteren plattformspezifischen Prozessen und Regeln:

Tabelle 2: Kernprozesse für eine Plattformorganisation

Prozess	Zweck
Ableitungs- oder Anwendungsprozess	Vorgaben für Applikationsprojekte, wie der Plattformcode anzuwenden und auf spezielle Projektbedürfnisse anzupassen ist
Änderungsmanagement	Vorgaben für Applikationsprojekte, wie Änderungen veranlasst werden können, um die Plattform z. B. an neue Kundenanforderungen anzupassen oder Kostenvorteile zu erreichen
Kompatibilitätsmanagement	Regeln, wie die Plattform Auf- und Abwärtskompatibilität mit anderen Komponenten aufrecht erhält oder auch bricht
Plattformentwicklung ("road map")	Regeln für die Weiterentwicklung der Plattform während ihres Lebenszyklus u. a. als Richtschnur für Applikationen, wann die Entwicklungsschritte in deren Projektphasen zu integrieren sind

Aus kommerzieller Sicht sollte sich eine solche Plattform nach einer Phase der Anschubfinanzierung durch „Lizenzumlagen“ der Applikationen finanzieren, die die Plattform verwenden.

Der zweite Schritt hat das Ziel, den Antagonismus zwischen Plattform und Applikation gänzlich aufzuheben. Das bedeutet, dass letztlich die gesamte Software inklusive der kundenspezifischen Teile „Plattform“ ist, die dann in verschiedene Subsysteme zerfällt. Die Aufgabe der Applikationsprojekte ist demnach „nur noch“, die Kundenanforderungen zu analysieren, sich daraus ergebende neue Anforderungen an die entsprechenden Subsysteme zu stellen, und – nach der Implementierung in Verantwortung der Plattformorganisation – die Subsysteme nach Kundenanforderungen zu konfigurieren, zu integrieren und zu verifizieren. Die Aufteilung der Subsysteme und die Definition der Subsysteminteraktion ist Aufgabe der Softwarearchitekten, die Teil Plattformorganisation sind. Diese agieren als „Design Authority“, und haben das Recht und die Pflicht, die Softwarearchitektur stabil aufrecht zu erhalten.

Praxisbewährtes Vorgehen – nachweisbare Erfolge

Zulieferer, die konsequent den beschriebenen Weg zu Wiederverwendung in der Softwareentwicklung beschreiten, werden ihre Entwicklungskosten nachhaltig senken können und deutlich mehr Projekte „on time and budget“ realisieren.

Dies stärkt wiederum die Wettbewerbsfähigkeit und trägt dazu bei, den eigenen Marktanteil ausbauen und sich in der Spitzengruppe der Elektronikzulieferer positionieren – als Unternehmen, das Software und Systeme entwickelt, die den Kundenwünschen wirklich gerecht werden.

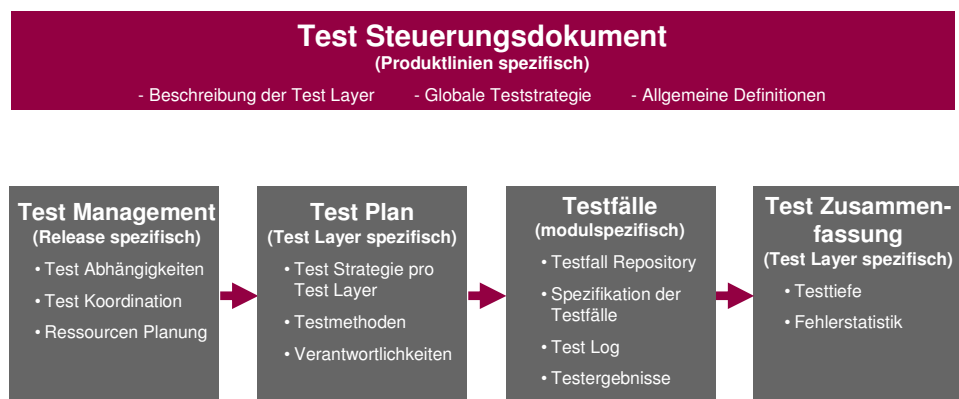
Dass es sich hier nicht um ein theoretisches Konstrukt, sondern um eine praxisbewährte Vorgehensweise handelt, zeigt ein kurzer Blick auf ein Projekt, das PA bei einem multinationalen Zulieferer unterstützt hat. Das Management des Unternehmens hatte die Notwendigkeit erkannt, einen Paradigmenwechsel zu vollziehen und den bislang verfolgten "carry-over"-Ansatz durch einen echten Plattformansatz abzulösen.

Parallel wurden zwei Arbeitspakete durchgeführt:

Es wurde der Kern der Plattform in Angriff genommen: die gemeinsame System- und Softwarearchitektur. Dazu wurden - auf Basis einer konkreten Produktfamilie elektronischer Steuergeräte - die bestehenden Anforderungen analysiert und generalisiert, um anschließend in einem Top-Down-Ansatz die System- und Softwarearchitektur bis hin zu Teilen des Softwarefeindesigns mittels UML 2 zu modellieren. Diese Projektstufe zeigte bereits die Herausforderungen, die es bei einer Plattformentwicklung zu meistern gilt: insbesondere die Divergenz der Kundenanforderungen, aber auch das Beharrungsvermögen der Entwicklungsorganisation.

Als weiteres Arbeitspaket wurde in einem konkreten Applikationsprojekt der Plattformgedanke im Testprozess eingeführt: Für Software-, Integrations- und Systemtest wurden eine Testarchitektur mit Prozessen und Prozessrollen definiert, die generisch genug waren, um zukünftigen Plattformanforderungen gerecht zu werden, aber auch konkret genug, um in dem laufenden Kundenprojekt erfolgreich eingesetzt werden zu können.

Grafik 1: Standardisierte Testprozesse



Beide Arbeitspakete wurden erfolgreich abgeschlossen - begleitet und testiert durch Audits gemäß Automotive SPICE. Weitere Projektphasen wurden geplant und werden sukzessive umgesetzt:

Corporate headquarters

123 Buckingham Palace Road
London SW1W 9SR
United Kingdom
Tel: +44 20 7730 9000
Fax: +44 20 7333 5050
E-mail: info@paconsulting.com

Deutschland

Eschersheimer Landstraße 223
60320 Frankfurt am Main
Kontakt: Karsten Groß
Tel: +49 69 71 70 20
Fax: +49 69 71 70 22 63
E-mail: marketing.de@
paconsulting.com

Thereseinstraße 6-8
80333 München
Tel: +49 89 288 90 540
Fax: +49 89 288 90 507

www.paconsulting.com

PA Consulting Group ist
eine international führende
Management-, System-
und Technologieberatung.
1943 gegründet, ist das
Unternehmen weltweit in
mehr als 35 Ländern aktiv.

Mehr Informationen über
PA in Deutschland unter
www.paconsulting.com/deutsch

Vertreten in

Argentinien, China, Dänemark,
Deutschland, Großbritannien,
Indien, Irland, Malaysia,
Neuseeland, Niederlande,
Norwegen, Schweden,
Vereinigte Arabische Emirate,
Vereinigte Staaten

© PA Knowledge Limited 2009.
All rights reserved.

- Fertigstellung des Feindesigns
- Überarbeitung und Generalisierung der Softwaremodule (Kodierung)
- Ableitung eines ersten Kundenprojekts aus der neuen Plattform
- Konsequente Einführung einer Plattformorganisation

Das Projekt ist noch nicht beendet. Doch weisen die bisher bereits erzielten Resultate unzweifelhaft darauf hin, dass der Unternehmensbereich in eine neue Ära der Softwareentwicklung aufgebrochen ist und der generelle Ansatz nicht mehr in Frage gestellt wird. Im Endeffekt führt all das zu beherrschbaren und geringeren Entwicklungskosten bei qualitativ besseren software-basierten Produkten für die Automobilindustrie – zumindest, wenn weitere Zulieferer einen solchen Entwicklungsansatz verfolgen.