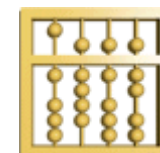

Software im Automobil

Potentiale, Herausforderungen, Trends

Manfred Broy



Technische Universität München
Institut für Informatik



Software im Automobil

Im Automobil finden sich gleichsam wie in einem Reagenzglas alle Herausforderungen an die moderne Informatik:

Sich schnell entwickelnde Nutzeranforderungen und Funktionalität, Multifunktionalität, Verteilung, großer Umfang, Vernetzung nach Innen und Außen, hohe Kritikalität und Zuverlässigkeitsanforderung, Einbettung in technische Systeme, Echtzeit, MMI, Security, Langelebige Systeme, Internationalisierung, Wartung, arbeitsteilige Entwicklung, Produktion (Down Load)

Schwerpunkte des Vortrags

- Fahrzeugtrends und die Bedeutung von Software
 - ◇ Potential für den Markt
 - ◇ Implikationen für Organisation, Management, Prozesse
- Gesamtkonzept Hard- und Software-Entwicklung
 - ◇ Integrierte Entwicklung und Entkoppelung
- Kostenmanagement und Benchmarking
 - ◇ Kostenfaktoren
 - ◇ Kostenmodelle
- Produktlebenszyklus und Wiederverwendbarkeit
- Nahtstellenmanagement:
Zulieferer – OEM – Service – Kunde
 - ◇ Prozesse
 - ◇ Schnittstellen

IT Trends

- Leistungszunahme
 - ◇ Hardware - Performanz
 - ◇ Software - Funktionalität □ Komplexität
- Übertragbarkeit von Lösungen: PLA
- Interoperabilität
- Vernetzung
 - ◇ Vernetzungsinfrastruktur
 - ◇ Logische Abhängigkeit □ Komplexität
- Konvergenz
- Sicherheit (Safety and Security)
- (Quasi)Standards

Software im Automobil

- Bis zu 100 Steuergeräte/Prozessoren
- Vernetzt durch mehrere Busse (CAN, MOST, ...)
- Zunehmend vernetzt nach Außen
- Bis zu 10.000.000 Lines of Code
- Komplexe Wartung: Software wird nachgeladen
- Komplexitätsbeherrschung schwierig (Echtzeit)
- Hohe Kritikalität
- Enormer Bedarf an
 - ◇ Zuverlässigkeit (Safety, Verfügbarkeit)
 - ◇ Sicherheit (Security)
 - ◇ Nutzerangemessenheit (MMI)

Fahrzeugtrends und die Bedeutung von Software

- Software als Innovationstreiber
 - ◇ Neue Funktionalität
 - ◇ Neuartige technische Lösungen
- Kundenorientierung
 - ◇ Personalisierung
 - ◇ Individualisierung
- Markttrends
 - ◇ Sicherheit und Vertraulichkeit
 - ◇ Komfort durch Zusatzdienste
- Neue Geschäftsfelder
 - ◇ Zusatzdienste
 - ◇ After Sales Geschäft

Neue Funktionalität

- Technische Fahrerassistenz
 - ◇ Adaption
- Personalisierung
- Nutzerdienste
 - ◇ Navigation
 - ◇ Verkehrsinformation
 - ◇ Flottenmanagement
- Integrierte technische Dienste
 - ◇ Energiemanagement
 - ◇ Fehlerdiagnose/management
- X by Wire

Schlüsselforderung: Zuverlässigkeit

- Aspekte der Zuverlässigkeit
 - ◇ Korrektheit
 - ◇ Einfachheit - Nutzertransparenz
 - ◇ Verfügbarkeit (Ausfallsicherheit)
 - ◇ Robustheit (Fail Safe, Unempfindlichkeit gegen Fehler in Umgebung oder Handhabung)
- Methoden zur Erreichung von Zuverlässigkeit
 - ◇ Anforderungen: Risikoanalysen
 - ◇ Entwurf: Fehlerbehandlungsverfahren
 - ◇ Realisierung: Qualitätssicherung
 - ◇ Wartung: Fehlerprotokollierung und Softwareupdates

Schlüsselforderung: Komplexitätsreduktion

- Entwicklung
 - ◇ Prozess
 - ◇ Dokumentation
- Produktion
 - ◇ Software Down Load
- Nutzung
 - ◇ MMI
- Wartung
 - ◇ Diagnose
 - ◇ Einfache Instandsetzung
- Verwaltung und Verkehrsführung

Herausforderung: Anforderungen identifizieren

- Entwurfsräume ausschöpfen
- Nutzererwartungen erfassen
- Implikationen abschätzen
- Abhängigkeiten analysieren
- Risiken vermeiden
- Anforderungen präzise dokumentieren
- Gesamtanforderungskonzept erstellen
- Systematisches Anforderungsmanagement (Änderungen, Verfolgung)
- Ganzheitliche Sicht (Kosten, Risiken, Folgeprobleme, Marktvorteile)

Architektur: System, Software, Hardware

Meilensteine eines eingebetteten System

- Domäne - Architektur der Funktionen (Nutzungsfälle)
 - ◇ Baum/Hierarchie der Nutzungsfälle
 - ◇ Varianten und Abhängigkeiten
- Komponentenzerlegung - Designarchitektur
 - ◇ Logisches Zusammenwirken
- Softwarearchitektur
 - ◇ Softwarekomponenten
 - ◇ Schnittstellen
 - ◇ Leistungsbedarf
- Plattform - Hardware und Software
 - ◇ Leistungsabgabe

Abhängigkeiten und Vernetzung

- Vernetzung der Funktionen
 - ◇ Funktionale Abhängigkeiten
 - ◇ Feature Interaction
 - ◇ Vernetzung nach außen
- Neue Funktionen durch Funktionskomposition

Konsequenzen für die Entwicklung

- Gesamtarchitektur
 - ◇ Funktionen/Use Cases
 - ◇ Komponentenzerlegung
 - ◇ Wirkungszusammenhänge
- Mehr Augenmerk auf die Integration

Qualitätsmanagement

- Anforderungen
 - ◇ Validierung
- Komponentenarchitektur
 - ◇ Zerlegung
 - ◇ Schnittstellen
 - ◇ Integration
- Komponenten
 - ◇ Test
 - ◇ Modelchecking
 - ◇ Verifikation
- Fehlermanagement
 - ◇ Im Entwicklungsprozeß
 - ◇ Im Softwareprodukt

Fehlermanagement in Software

- Risikoanalyse
 - ◇ Welche Fehler können auftreten
 - ◇ Welche Konsequenzen
- Fehlermodellierung
 - ◇ Wie sehen die Fehler genau aus
- Fehlermanagement
 - ◇ Wie auf Fehler reagieren
 - ◇ Wie Mindestmaß an Funktionalität erhalten
- Protokollierung und Dokumentation
- Fehlermanagement in der Wartung
 - ◇ Diagnose
 - ◇ Beseitigung

Fehlertoleranz

- Redundanz
 - ◇ Hardwareredundanz
 - Schutz vor Hardwaredefekten
 - ◇ Softwareredundanz
 - Idee: Software mehrfach auslegen
 - Problem: Software hat keine Materialfehler sondern logische Fehler
- Defensives Programmieren
 - ◇ Zusatzüberprüfungen
 - ◇ Fehleranfällige Konstruktionen vermeiden
- Fehlermanagement
 - ◇ Zentrale Erfassung
 - ◇ Wiederanlauf

Gesamtkonzept Hard- und Software-Entwicklung

- Die Rolle der Software und der Hardware
 - ◇ Modelle
 - ◇ Modelle des Zusammenwirkens
 - ◇ Optimierung
- Software - Funktionalität
- Hardware - Ressourcen
- Offene Systeme - Portabilität und Erweiterbarkeit
 - ◇ Schichtenarchitekturen
 - ◇ Plattformen
 - ◇ Softwarearchitekturen
 - ◇ Flexibles Hardware/Software-Mapping
- Kostenoptimierung

Kostenmanagement und Benchmarking

- Kostenwirkfaktoren
 - ◇ Entwicklungskosten
 - ◇ Produktionskosten
 - ◇ Wartungskosten (Pflegekosten im Feld)
 - ◇ Folgerisiken
- Wechselspiel Hardware/Softwarekosten
- Nutzenwirkfaktoren
 - ◇ Gewinn an Funktionalität
 - ◇ Gewinn an technischer Qualität
 - ◇ Kostenverringderung
 - ◇ Image/Vermarktung

Zusammenhänge ungenügend verstanden

Implementierung

- Korrektheit
 - ◇ Echtzeit
 - ◇ Safety
- Performanz
- Plattformunabhängigkeit
 - ◇ Portabilität

Wartung

- Änderungsmanagement
 - ◇ Fehlermanagement
 - ◇ Funktionale Erweiterung
 - ◇ Hardwareerneuerung
- Software Download
- Konfigurationsmanagement
- Kompatibilitätsmanagement

Software-Produktlebenszyklus

- Projektierung
 - ◇ Welche Funktionalität
 - ◇ Welche Kosten/Nutzen
- Entwicklung
 - ◇ Strukturierung
 - ◇ Realisierung
- Produktion
- Wartung
 - ◇ Versions- und Konfigurationsmanagement
 - ◇ Softwaredownload
 - ◇ Softwarelogistik
- Weiterentwicklung

Wiederverwendbarkeit

Die steigenden Investitionskosten für Software erfordern einen Investitionsschutz: Die Wiederverwendung von Entwicklungsergebnissen

Schwieriges Thema:

- Erfahrungen mit Wiederverwendung in der Softwaretechnik zwiespältig
 - ◇ Ad Hoc Wiederverwendung
 - ◇ Systematische Wiederverwendung
- Produktlinien

Nahtstellenmanagement: Zulieferer – OEM – Kunde

Schlüsselprobleme:

- Anforderungen unscharf
- Schnittstellen nicht systematisch festgelegt
- Kein definierter Prozeß Zulieferer/Auftraggeber

Lösungsansätze

- Präzise technische Dokumentation
- Übergreifender Prozeß
 - ◇ Integrierte QS
 - ◇ Gemeinsames Prozeß/Produktverständnis

Spannungsfeld: Flexibilität und Abstimmung

IT Sicherheit - Security

Mit Vernetzung wachsen die Sicherheitsanforderungen:

- Identifizierung
- Authentifizierung
- Autorisierung
- Privacy

In Hinblick auf

- Nutzer
- Wartung
- Softwareinfrastruktur

Zusammenfassung

Software im Fahrzeug bleibt ein Gebiet hoher Dynamik:

- Viele Problemfelder
- Schneller Technologiefortschritt
- Neue Anwendungsfelder
- Neue Technologien
- Schnelle Lernprozesse erforderlich
 - ◇ Markt
 - ◇ Organisation
 - ◇ Prozesse
 - ◇ Lösungen
 - ◇ Werkzeuge

Antworten der Softwaretechnik

- Konsequente Modellorientierung
 - ◇ Anforderungen
 - ◇ Architektur/Schnittstellen
 - ◇ Plattformunabhängige Modellierung
 - ◇ Test
- Strikte Architekturzentrierung
 - ◇ Modularität
 - ◇ Concurrent Engineering
 - ◇ Schichtung
- Vereinheitlichung/PLA