

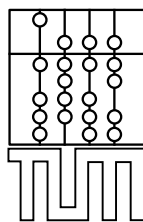
---

Kurzfassung des Abschlußberichtes der Vordringlichen Aktion des  
Bundesministeriums für Bildung und Forschung

# Entwicklung, Produktion und Service von Software für eingebettete Systeme in der Produktion



Arbeitsgruppe Prof. Gausemeier  
Heinz Nixdorf Institut  
Universität Paderborn



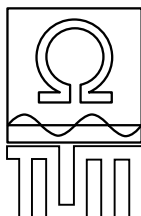
Lehrstuhl Prof. Broy  
Fakultät für Informatik  
TU München



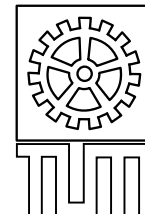
Arbeitsgruppe Prof. Schäfer  
Fachbereich  
Mathematik/Informatik  
Universität Paderborn

**EPS  
SES**

Entwicklung, Produktion und Service von  
Software für eingebettete Systeme in der  
Produktion"



Lehrstuhl Prof. Färber  
Fakultät für Elektro- und  
Informationstechnik  
TU München



Lehrstuhl Prof. Bender  
Institut für  
Informationstechnik im  
Maschinenwesen  
TU München



Abteilung Prof. Pfeifer  
Institut für  
Produktionstechnologie  
Fraunhofer Gesellschaft  
Aachen

Bernhard Schätz, Michael Fahrmaier, Michael von der Beeck,  
Peter Jack, Hans Kespohl, Ali Koç, Benito Liccardi, Sandra  
Scheermesser, Albert Zündorf

---



Dieses Forschungs- und Entwicklungsprojekt wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) innerhalb des Rahmenkonzepts „Produktion 2000“ gefördert und vom Projektträger Produktion und Fertigungstechnologien, Forschungszentrum Karlsruhe betreut.

Das vorliegende Dokument wurde im Rahmen der Vordringlichen Aktion „Entwicklung, Produktion und Service von Software für eingebettete Systeme in der Produktion“ erstellt. Die Durchführung der Arbeiten erfolgte durch die Projektpartner

- Lehrstuhl Prof. Dr. Manfred Broy, Fakultät für Informatik, Technische Universität München, Projektleitung
- Lehrstuhl Prof. Dr. Klaus Bender, Institut für Informationstechnik im Maschinenwesen, Technische Universität München
- Lehrstuhl Prof. Dr.-Ing. Georg Färber, Fakultät für Elektro- und Informationstechnik, Technische Universität München
- Arbeitsgruppe Prof. Dr. Jürgen Gausemeier, Heinz Nixdorf Institut, Universität Paderborn
- Forschungsabteilung Prof. Dr.-Ing, Dr. h.c. Tilo Pfeifer, Institut für Produktionstechnologie, Fraunhofer Gesellschaft Aachen
- Arbeitsgruppe Prof. Dr. Wilhelm Schäfer, Fachbereich für Mathematik und Informatik, Universität Gesamthochschule Paderborn

Die Durchführung der Arbeit wurde durch einen Steuerkreis überwacht. Mitglieder des Steuerkreises sind

- Dipl.-Ing. Knut Balzer, Robert Bosch GmbH, Zentralbereich Forschung und Voraentwicklung, Abteilung Produktionsinformatik und Logistik, Steuerkreissprecher
- Dr.-Ing. Lothar Borrmann, Siemens AG, Zentralabteilung Technik, Fachzentrum für Software- und Systemarchitekturen
- Dr.-Ing. Robert Grob, DaimlerChrysler AG, Zentralbereich Qualitätsmanagement
- Dipl.-Ing. Erik Mertens, Projektträgerschaft Produktion und Fertigungstechnologien PFT, Forschungszentrum Karlsruhe
- Dipl.-Ing. Claus Oetter, Fachgemeinschaft Software, VDMA Verein Deutscher Maschinen- und Anlagenbau e.V.
- Dr.-Ing. Rainer Stetter, SoftwareFactory, Industriekreissprecher

Die Ergebnisse der Arbeit wurden durch die Mitglieder eines Industriekreises überwacht. Die teilnehmenden Unternehmen sind im Anhang dieses Dokuments aufgeführt.



## 0 Kurzfassung

Dieser Abschnitt enthält auszugsweise die im Hauptteil ausführlich beschriebenen Ergebnisse der VA. Da aus den Ergebnissen einer Umfrage jedoch nur bereits existierende oder kurzfristige zukünftige Probleme ermittelt werden können, werden in einem weiteren Punkt die Ergebnisse der ebenfalls angewandten Szenariotechnik zusammengefasst. Mit Hilfe dieser Technik ist es möglich, aus vorhandenem Expertenwissen methodisch mögliche zukünftige Entwicklungen darzustellen und in leicht verständlicher Form zu bündeln. Die Szenarios zeigen mögliche Entwicklungstendenzen der nächsten acht Jahre auf und geben dadurch Mittel an die Hand, vom gegenwärtigen (durch die Umfrage ermittelten) Ist-Zustand auf mittel bis langfristige Probleme, Risiken und Verbesserungspotentiale hinzuweisen. Im letzten Punkt werden die entwickelten Einzellösungsschritte zu Themenfeldern gebündelt, in denen durchgeführte Aktionen den größtmöglichen Nutzen für eine breite Zielgruppe erreichen können (Aktionsfelder).

### 0.1 Executive Summary

Infolge der zunehmenden Bedeutung der Informationstechnologie in der Produktion als Wertschöpfungsfaktor, sowohl im Produktionsprozeß als auch im Produkt selbst, ist die ingenieurmäßige Softwareentwicklung gerade für die KMUs der Zukunft als dritte Kernkompetenz zusätzlich zu Maschinenbau und Elektrotechnik von entscheidender Bedeutung. Die Informationstechnologie erlaubt nicht nur schnellere und effizientere Realisierung von Produkten sondern auch die Umsetzung neuer Funktionalitäten und sichert so die internationale Wettbewerbsfähigkeit der KMUs.

Vordringlichstes Ziel ist daher die Integration der beteiligten Fachrichtungen Maschinenbau, Elektro- und Informationstechnik und Informatik. Dies bedeutet insbesondere das Zusammenführen ihrer Ansätze und Werkzeuge in eine anwendungsorientierte Vorgehensweise zur Entwicklung eingebetteter Systeme. Folgende Forschungs- und Entwicklungsaktivitäten erscheinen in dieser Hinsicht besonders vordringlich:

- Adaption und Transfer

Aufgrund der Eigenarten eingebetteter Systeme sind im Bereich Software-Engineering entwickelte Ansätze und Verfahren nicht immer unmittelbar einsetzbar. Benötigt wird daher die Anpassung vorhandener Vorgehensmodelle (inkl. Kostenmodelle) und Richtlinien an die Bedürfnisse der KMUs, einschließlich der Einführung von Konfigurationsmanagement mit Berücksichtigung der Variantenvielfalt und von auf die Anwendung für eingebettete Systeme zugeschnittenen Qualitätssicherung. Ebenso benötigt werden darauf aufbauend Maßnahmen zur Vermittlung der ingenieurmäßigen Softwareentwicklung sowie allgemein aktueller Entwicklungen und Lösungsansätze im Bereich der Informationstechnologie.

- Beherrschung der Komplexität

Die Bedeutung der Einzelfertigung im Anlagenbau einerseits sowie das Zusammenspiel von unterschiedlichen Fachgruppen im Bereich Produktionstechnik andererseits machen die Beherrschung der Komplexität im Entwicklungsprozeß zu einer wesentlichen Aufgabe. Themen von besonderer Dringlichkeit sind hier Definition und Einsatz von Bausteinen und Standardarchitekturen und Standardplattformen bestehend aus Hardware und Software und die Rückverfolgung und Sicherung der Abhängigkeiten im Entwicklungsprozeß über Produktversions- und Fachrichtungsgrenzen.

- IT-Werkzeuge für die Produktion

Wesentliche Reibungsverluste entstehen durch die fehlende Integration der Werkzeuge der Fachrichtungen. Benötigt wird daher eine anwendungsorientierte durchgängige und fachrichtungsübergreifende Werkzeugunterstützung für die gesamte Systementwicklung. Unterthemen reichen hier vom einheitlichen Produktdatenmanagement über Konsistenzsicherung, bausteinorientiertes Arbeiten bis hin zur Generierung von Testfällen, Steuerungscode, Benutzeroberflächen oder Dokumentation.

- Integriertes Vorgehensmodell für die Produktion

Infolge der Kombination verschiedener Fachrichtungen und Aktivitäten zur Realisierung eines Produkts kommt dem Einsatz eines übergreifenden Vorgehensmodell für die ingenieurmäßige Entwicklung eine Schlüsselrolle zu. Hier zu behandelnde Unterthemen sind vor allem die fachrichtungs-, unternehmens- und phasenübergreifende Entwicklung, Systeminteroperabilität, ein aktives, entwicklungsbegleitendes Qualitätsmanagement, Kosten- und Risikomanagement sowie die Schaffung von Standards für Dokumente und Beschreibungstechniken.

Da die hier beschriebenen Themenfelder im Regelfall nicht unabhängig zu behandeln sind, sollen insbesondere solche Projekte berücksichtigt werden, die Aufgabenstellungen aus unterschiedlichen Themenfelder kombiniert behandeln.

## **0.2 Interdisziplinäre Zusammenarbeit**

Bei der Erstellung und Planung von Systemen mit eingebetteter Software ist die Zusammenarbeit verschiedener Disziplinen notwendig.

In kleinen Projektteams können nach einer Anfangsphase, die bei der Klärung unterschiedlicher Begrifflichkeiten behilflich ist, die der Herstellung eines einheitlichen Projektverständnisses im Wege stehenden Kommunikations- und Verständnisprobleme durch gegenseitiges Voneinanderlernen schnell behoben werden. Die unterschiedliche Sichtweise der beteiligten Ingenieurdisziplinen führt gerade in den frühen Entwicklungsphasen zu einer ganzheitlicheren Betrachtung und hilft in der Analysephase, wichtige Systemaspekte nicht zu vergessen.

In größeren Projekten ist die Zusammenarbeit zwischen Softwareentwicklern und Systementwicklern auf kurze Absprachen beschränkt. Hier ist auch eine hierarchische Abstufung zwischen den Entwicklungsdomänen zu beobachten. Während die Entwicklung des Gesamtsystems das Projekt dominiert und dabei Projektphasen, Meilensteine, Anforderungen und Randbedingungen bestimmt, wird die Software als der flexibelste Part verstanden, der sich den Gegebenheiten im Gesamtprojekt uneingeschränkt anpassen muss. Hier fehlt auf beiden Seiten das Verständnis für die Bedürfnisse der verschiedenen Entwicklungsprojektbestandteile.

Bei der Bewertung dieser Ergebnisse ist jedoch zu berücksichtigen, daß nur in wenigen Unternehmen Informatiker an der System- und Softwareentwicklung beteiligt sind; im Regelfall sind die „traditionellen“ Disziplinen Maschinenbau und Elektrotechnik an der Entwicklung beteiligt.

Die Analyse der Schnittstellen, die der softwareentwickelnde Unternehmensbestandteil mit anderen Bereichen hat, zeigt, daß eine Zusammenarbeit mit Vertrieb, Marketing und Produktplanung in den meisten Unternehmen vorkommt. Aus Einzelaussagen lässt sich entnehmen, daß gerade die Kommunikation mit dem Vertrieb und den betriebswirtschaftlichen Bereichen schwierig und unzureichend ist. Ursache ist mangelndes Wissen über die Eigenschaften von Software und überholte Vorstellungen in denen beispielsweise Anlagen noch immer nur aus Hardware bestehen und die Software lediglich ein Zubehör, aber kein integraler Bestandteil ist.

Aus diesem Grunde ist vielfach die Kommunikation der Komplexität nicht möglich, besonders in den frühen Phasen der Anforderung und ganz besonders, wenn es sich um neue Aufgabenstellungen handelt.

In der Softwareentwicklung sind überdies starke Nachwuchsprobleme zu verspüren. Aufgrund mangelnder Erfahrungen wird zudem ein Defizit im Bereich Projektmanagement deutlich. Gerade im universitären Bereich werden den zukünftigen Entwicklern kaum organisatorische Fähigkeiten und Methoden vermittelt. Insgesamt vermissen die Unternehmen bei den Berufsanfängern Wissen über organisatorische Zusammenhänge in einem Unternehmen und Erfahrungen in Teamarbeit.

### **Lösungsansätze:**

Eine Lösung der beschriebenen Probleme kann durch folgende Ansätze angestrebt werden:

- **Bereitstellung und Pflege von domänenübergreifenden Terminologiedatenbanken**  
Damit über die Disziplinen hinweg ein fruchtbarer Gedankenaustausch stattfinden kann, ist es wichtig mit Hilfe von Terminologiedatenbanken zum einen Begriffe mit mehreren Bedeutungen beziehungsweise einer Bedeutung, die durch mehrere disziplinspezifische Ausdrücke beschrieben wird, zu identifizieren und zum ändern in einer Art Glossarfunktion wirklich domänenspezifische Begriffe zu erklären.
- **Integrierte und praxisorientierte Ausbildung/Studiengänge**  
Abgesehen davon, daß ein Studiengang geschaffen werden könnte, der die besonderen Anforderungen an Projektingenieure, Entwicklungsleiter und Produktverantwortliche in Unternehmen, die eingebettete Systeme erstellen, vermittelt, sollten vorhandene Studien- und Ausbildungsgänge so angepasst werden, daß die entscheidenden Fähigkeiten vermittelt werden, welche die Entwicklung und Innovation solcher Systeme problemlos ermöglichen. Dazu gehört unter anderem die Förderung einiger sogenannter „soft-skills“, wie Kommunikations- und Teamfähigkeit. Weiterhin müssen Methoden gelehrt werden, wie Informationen schnell beschafft und für die jeweilige Situation so aufgearbeitet und präsentiert werden können, daß sie von anderen Teammitgliedern problemlos aufgenommen werden können. Die Integration von Praxissemestern oder die Einbindung von Studenten in industrielle Projekte muss tiefer in den Studiengängen verankert werden
- **Integrierte Projektmanagementmethoden**  
In jeder Disziplin gibt es spezifische Vorgehensweisen bei der organisatorischen Planung und Durchführung von Entwicklungsprozessen. Diese Projektmanagementmethoden sind um solche Ansätze zu erweitern oder zu ergänzen, die eine ideale Zusammenarbeit verschiedener Entwicklungsteams unterschiedlicher Disziplinen ermöglichen und verbessern.  
Mit der zu entwickelnden Art von Projektmanagementmethodik sollte vor allem der momentan schwer planbare Zeitbedarf für die Entwicklung von Software optimiert werden. Hier gilt es Frühwarnsysteme zu integrieren, die Alarm schlagen, sobald ein Projekt zeitlich oder bezüglich der erwarteten Ergebnisse nicht der Zielplanung entspricht.

## **0.3 Vorgehensweise**

Bei den meisten Unternehmen (ca. 75%) findet sich ein phasenorientiertes Vorgehen, das aus den Hauptphasen

- Anforderungsanalyse mit Erstellung der Anforderungsdokumente (Lastenheft, Pflichtenheft),

- Design mit Erstellung der Systemarchitektur und der Systemkomponenten oder des funktionalen Aufbaus des Systems, teilweise unterschieden in Grob- und Feindesign,
- Implementierung oder Umsetzung des Systems in Form von Hardware und Software (C-, C++-, SPS-Code etc.),
- Test (Modultest, Integrationstest, Inbetriebnahmetest, Systemabnahmetest) und
- Systembetrieb mit Service und Wartung (Fehlerbehebung, Einspielen neuer SW-Versionen)

besteht. Dabei sind nicht immer alle Phasen vorhanden; insbesondere die Designphase existiert nicht in allen Unternehmen in Form einer eigenständigen Phase. Auch sind die einzelnen Phasen und damit das gesamte Vorgehensmodell unterschiedlich stark ausgeprägt. Dabei liegt das Gewicht insbesondere auf den späten Entwicklungsphasen, vor allem der Implementierung sowie der Qualitätssicherung in Form von Tests. Nur in wenigen Unternehmen wird ein standardisiertes Vorgehensmodell wie das V-Modell mit den damit verbundenen Dokumenten eingesetzt.

Allgemein ist bei jedem Unternehmen, unabhängig davon ob ein definiertes Vorgehensmodell vorhanden ist oder nicht, eine Analysephase in der praktischen Projektdurchführung vorgesehen; generell ist die Bedeutung dieser Phase anerkannt und damit ein entsprechendes Vorgehen definiert. Bei über 80% der befragten Unternehmen ist die Erstellung eines Pflichtenhefts vorgesehen sowie die Vorgaben für die Durchführung der Analyse oder die Erstellung des Pflichtenhefts vorhanden. Dabei unterscheiden sich jedoch die in der Analysephase vorgeschriebenen Aktionen wesentlich. Nur bei ca. 50% liegen genaue Vorgaben zur Durchführung der Analyse vor, einschließlich der Einplanung von Reviews.

Die Erstellung eines Lastenhefts ist nicht in allen Fällen vorgesehen (und auch nicht in allen Fällen vollständig möglich, beispielsweise bei innovationsgetriebenen Prozessen). Auch in den Fällen, in denen ein Lastenheft als Vertragsgrundlage eingesetzt wird, erfolgt die Erstellung des Lastenhefts im Regelfall noch nicht systematisch. Auch in den Fällen, in denen ein Lastenheft oder vergleichbares Dokument vorgesehen ist, wird das Pflichtenheft nur in Ausnahmefällen systematisch daraus abgeleitet. Eine Möglichkeit der Rückverfolgung von Anforderungen aus dem Pflichtenheft zu Anforderungen aus dem Lastenheft ist generell nicht gegeben.

Ähnlich wie im Fall der Analysephase - mit mehr oder weniger ausgeprägten Vorgaben einschließlich eines Pflichtenhefts - existiert prinzipiell auch eine eigenständige Designphase. Diese ist aber im Regelfall noch etwas schwächer ausgeprägt als die Analysephase: nur ca. 60% der Befragten legen im Design den Systemaufbau einschließlich funktionaler Anforderungen an die Systemkomponenten - mehr oder weniger vollständig - in Form des Feindesigns fest. Wie auch bei den bisher beschriebenen Entwicklungsdokumenten wird in den Designdokumenten der softwaretechnische Anteil im allgemeinen nur unvollständig, unpräzise abstrakt (d.h. ohne Implementierungsinformation) beschrieben. Eine genaue Beschreibung der Funktionalität (des Verhaltens) der Softwarekomponenten wird nur in Ausnahmefällen erstellt.

Die Implementierungsphase spiegelt die Vorgehensweise der Softwareentwicklung wie keine andere Phase wider. Für die Mehrheit der Unternehmen findet Softwareentwicklung ausschließlich in der Implementierungsphase statt, wohingegen für einzelne Ausnahmen diese Phase lediglich zur Konvertierung des erarbeiteten Design in eine physikalische Abbildung ist.

Die Ist-Situation stellt sich gerade in dieser Phase aufgrund der jahrzehntelangen gereiften Implementierungstechnik nicht in allen Punkten so schlecht dar. Alle Unternehmen setzen Standardentwicklungsumgebungen ein, die in ihrer Anwendung gut verstanden sind, jedoch als „stand alone“ Produkte noch weitgehend nicht mit Werkzeugen anderer Phasen verknüpft sind.



Es kommen Implementierungsrichtlinien zum Einsatz, deren Einhaltung in vielen Fällen fraglich beziehungsweise nicht nachvollziehbar ist.

Eine phasenübergreifende vollständige und durch ein Vorgehensmodell abgedeckte Wiederverwendung wird entweder überhaupt nicht oder nur vereinzelt betrieben. Dabei findet eine Wiederverwendung nur auf Basis von Klassenbibliotheken, GUI Bausteinen und SPS Code Modulen statt.

Betrachtet man die oben angesprochene automatische Generierung der ausführbaren Codestücke, so lässt sich feststellen, daß fast kein Unternehmen diese Technik ernsthaft einsetzt. Der generierte Code ist oft zu groß, zu langsam oder schlichtweg nicht generierbar, da die vorhergehende Designphase fehlt.

Harte als auch weiche Echtzeitbedingungen, die eine zentrale Rolle in der Entwicklung für Software für eingebettete Systeme spielen, werden zwar als wichtig angesehen aber an keiner Stelle formal erfasst beziehungsweise implementierungsspezifisch berücksichtigt.

Auch die Verwaltung von Hardwareressourcen und die Verlagerung der Komplexität in eine Modulare Struktur durch ein Realzeitbetriebssystem wird von wenigen eingesetzt. Auch hier ist die Akzeptanz durch Kostenüberlegungen oder aus Performanzgründen gedämpft.

Service und Wartung wird von den befragten Unternehmen sehr unterschiedlich behandelt. Komplex ausgestaltete Formen nimmt die Wartung bei Systemen an, die über Ferndiagnosemöglichkeiten verfügen und über Datenleitung steuerbar, instandsetzbar und wartbar sind. Systeme, die Telemonitoring, Teleservice und Prozessaufzeichnung ermöglichen, haben High-Tech-Standard und nutzen sämtliche Möglichkeiten der Kommunikationstechnologie. Eine weitere Abstufung bilden die Produkte der Unternehmen, die Logbuchfunktionalitäten enthalten oder Diagnoseergebnisse und Daten erfassen und speichern, die bei Fehlfunktionen analysiert werden können. Jedoch nicht bei allen Unternehmen, deren Systeme dies ermöglichen, werden die Diagnoseergebnisse systematisch analysiert und genutzt. Die letzte Stufe bilden die Systeme, die bei Fehlfunktionen oder Ausfall komplett ausgetauscht werden, ohne daß eine Ursachenanalyse stattfindet.

Die organisatorische Zusammenarbeit der Entwickler mit den Servicemitarbeitern sehr unterschiedlich gestaltet. Sie reicht von keinerlei Kommunikation bis dahin, daß die Entwickler in der Produkteinführungsphase und auch noch später selbst am Service beteiligt sind. Im zweiten Fall stellt sich die Frage nach einer ausreichenden Qualifikation der Servicemitarbeiter nicht, während es im anderen Fall zum Teil bemängelt wurde.

### **Lösungsansätze:**

Eine Lösung der beschriebenen Probleme kann durch folgende Ansätze angestrebt werden:

- Definition standardisierter Beschreibungen für den Softwareanteil eingebetteter Systeme als Grundlage für Lasten- und Pflichtenhefte

Während für den Hardwareanteil eingebetteter Systeme standardisierte Beschreibungsformen vorliegen, gilt vergleichbares für den Softwareanteil nicht. Die bisher bei der Softwareentwicklung allgemein eingesetzten Beschreibungsformen sind eher im Bereich der datenintensiven Systeme angesiedelt (z.B. E/R-Diagramme, Klassendiagramme). Die für die Aspekte eingebetteter Systeme geeigneten Beschreibungsformen unterscheiden sich je nach Ansatz und Werkzeughersteller (z.B. State Charts, State Flow Diagrams, ROOM State Diagrams, etc.). Die Definition geeigneter Beschreibungsformen bietet den Unternehmen die Möglichkeit, die Anforderungen an den Softwareanteil umfassend, präzise und trotzdem ausreichend abstrakt und kompakt zu beschreiben.

- Definition standardisierter, produktbereichspezifischer Standardarchitekturen und „hoher“ Schnittstellen für den modularen Aufbau von Systemen aus HW/SW-Bausteinen

Obwohl gerade in vielen Bereichen der Entwicklung eingebetteter Systeme die Verwendung von Bausteinen und Standardarchitekturen in der ingenieurmäßigen Tradition der Systementwicklung liegt, finden sich standardisierte Schnittstellen eher im Bereich der Hardware oder der Software, weniger jedoch bei deren Kombination. Die Verwendung einheitlicher hoher Schnittstellen (z.B. einheitliche Fehlerbehandlung, einheitliche Steuerung) für ganze Baugruppen erlaubt eine bausteinorientierte Entwicklung des Systems durch Zusammenstellen solcher Komponenten. Der Einsatz einheitlicher Architekturen und Schnittstellen ist besonders für die Kombination von Komponenten verschiedener Hersteller von Bedeutung. Die Verwendung hoher Schnittstellen erlaubt eine einfachere Integration von verschiedenen Elementen und Versionen vergleichbarer Baugruppen.

- Definition eines integrierten Vorgehensmodells einschließlich geeigneter Beschreibungen und Werkzeugunterstützung

Für den Einsatz bei der Entwicklung eingebetteter Systeme wird ein Vorgehensmodell benötigt, das die Besonderheiten solcher Entwicklungen (z.B. stark unterschiedlicher Projektumfang, intensiver Einsatz von Bausteinen und Standardarchitekturen, interdisziplinäre Entwicklung, Berücksichtigung von strengen Anforderungen wie Echtzeitkriterien) berücksichtigt. Wesentliche Ziele des integrierten Vorgehensmodells sind die Konsistenzüberprüfung beschriebener Systeme und Komponenten, der Einsatz von Simulation, die Möglichkeit der Generierung von ausführbarem Code, von Testfällen sowie von Dokumentation sein. Dabei darf sich die Werkzeugunterstützung wie bisher nicht nur auf die Behandlung der späteren Phasen (ab Design) beschränken, sondern muss bereits ab der Anforderungsanalyse ansetzen und insbesondere die Möglichkeit der Dokumentation von Entwurfsentscheidungen sowie der Rückverfolgung von Anforderungen berücksichtigen. Ein solches Vorgehensmodell muss möglichst übergreifend für den Bereich der eingebetteten Systeme definiert werden. Dies ist vor allem für eine unternehmensübergreifende Entwicklung von Systemen von Bedeutung, bei der einzelne Arbeitspakete von unterschiedlichen Unternehmen entwickelt werden.

- Verbesserung der Vorgehensweise in der Implementierung

Die Punkte Entwicklungsumgebung, Programmierrichtlinien und Wiederverwendung werden im Abschnitt 5.1.2 II „Definition eines integrierten Vorgehensmodells einschließlich geeigneter Beschreibungen und Werkzeugunterstützung“ behandelt. Eine Bearbeitung dieser Thematik kann nur in einem phasenübergreifenden Kontext bearbeitet werden und wird hier nicht näher aufgeführt.

Die Einhaltung von Echtzeitbedingungen verlangt zum einen nach einer formalen Spezifikation dieser Randbedingungen und zum anderen nach einer „Messbarkeit“ dieser Kennwerte. Daher werden Vorgehensweisen (Benchmarking Methoden) benötigt, die Leistungskennwerte von Software (und in Zukunft auch Softwarebausteine) transparent darlegen.

- Optimierung automatischer Codegenerierung

Ein zweiter zentraler Lösungsansatz ist die Verbesserung der „Last Mile“ Phase, d.h. die Optimierung der automatischen Codegenerierung. In diesem Fall sind die existierenden Lösungen nicht auf den Spezialfall der eingebetteten Systeme abgestimmt. Für eine mittel- bis langfristige Lösung müsste eine Beschreibung der verwendeten Hardwarearchitektur existieren, die eine automatische Generierung von Compilern für diese Architektur erlaubt.

- Automatisiertes Design - Intelligente Bausteine, selbstorganisierende adaptive dynamische Systeme und Middleware für eingebettete Systeme

Einige Zukunftsprojektionen haben gezeigt, dass bereits innerhalb der nächste acht Jahre autonome, dezentral vernetzte intelligente mechatronische Systeme eine zunehmend stärkere Rolle spielen könnten, weshalb es angeraten erscheint, sich bereits jetzt durch die Beschäftigung mit den dafür notwendigen technischen und methodischen Grundlagen einen Vorsprung zu verschaffen.

Neben dem Aufbau einer adäquaten Vertriebsstruktur für Bausteine (auffinden, zertifizieren von Echtzeit und Sicherheitseigenschaften, automatische Lizenzierung usw.) kommt der Middleware, d.h. der softwaretechnologischen Infrastruktur für den Betrieb dieser Art von Systemen und Bausteinen eine gesteigerte Bedeutung zu. Bestehende Ansätze wie SUNs Jini oder Microsofts UPNP müssen auf ihre Tauglichkeit untersucht und auf die Bedürfnisse von eingebetteten Systemen angepasst beziehungsweise in diese Richtung erweitert werden.

## 0.4 Qualitätssicherung

Eine der wichtigsten Aufgaben der Unternehmen liegt in der Sicherung der Produktqualität. Vor dem Hintergrund, dass heute - und in Zukunft verstärkt - die Software eine Kernkomponente von innovativen Produkten darstellt, hat die SW-Qualität einen hohen Einfluss auf die Qualität des Gesamtprodukts. Die Herausforderung an die Qualitätssicherung liegt darin, eine hinreichende Fehlerfreiheit der Software bei akzeptablem Zeit- und Kostenaufwand zu garantieren. Wobei die Komplexität der Qualitätssicherung von eingebetteten Software höher ist als bei herkömmlicher Software, da das eingebettete System wesentliche Unterschiede aufweist, wie z. B. die Echtzeitfähigkeit und die enge Verzahnung von Software, Hardware und Gerätemechanik.

Fast alle Unternehmen besitzen ein definiertes bzw. standardisiertes Qualitätsmanagementsystem, jedoch wird dabei die Softwareentwicklung meist außer Acht gelassen bzw. nur sehr grob beschrieben. Auch fehlt es oft an einer kompetenten Organisationseinheit für die SW-Qualitätssicherung. Daher verwundert es nicht, dass die SW-Qualitätssicherung oft wenig geplant und unsystematisch vorgenommen wird.

Obwohl alle Unternehmen sich prinzipiell bewusst sind, dass Fehler möglichst frühzeitig entdeckt werden sollten, wird der meiste Aufwand dennoch erst am Ende der Entwicklung in den Testphasen (Modultest, Integrationstest, Systemtest) betrieben, was dazu führt, dass Projektverzögerungen sich unmittelbar auf die Testzeit auswirken. In den frühen Phasen (Analyse, Design) werden bestenfalls Reviews angewendet, und diese meist auch sehr unsystematisch und ohne eine explizite Planung.

Das Testen erfolgt meist unsystematisch durch den Entwickler selbst und ist dabei zusätzlich erfahrungsabhängig. Erschwerend kommt hinzu, dass das Ermitteln von Testfällen nicht methodisch erfolgt und ein geeignetes und einheitliches Beschreibungsmittel für Testfälle fehlt. Auch ein Werkzeugeinsatz für die einzelnen Testaktivitäten findet nur in geringem Maße statt. Dies lässt sich aber hauptsächlich dadurch begründen, dass die meisten kommerziell verfügbaren Werkzeuge zum Testen die besonderen Anforderungen von eingebetteten Systemen nur unzureichend erfüllen.

### Lösungsansätze:

Die Besonderheiten eingebetteter Software unterscheiden sich grundsätzlich von denen der gewöhnlichen Softwaresysteme. Die dadurch bedingte zusätzliche Komplexität bedarf im

Bereich der Qualitätssicherung neuer Ansätze und Lösungen, die im folgenden kurz beschrieben werden:

- **Evaluierung von Werkzeugen und Methoden für die Qualitätssicherung**  
Da der Markt der verfügbaren Werkzeuge und Methoden sehr unübersichtlich ist und KMUs gewöhnlich nicht die Kapazitäten haben um deren Tauglichkeit im Bereich der Entwicklung Eingebetteter Systeme zu evaluieren, sollte diese Evaluierung innerhalb eines Projektes bearbeitet werden.
- **Bewertung von QS-Maßnahmen**  
Die Qualität und Effektivität von QS-Maßnahmen sind oftmals schwer zu bewerten, bzw. es gibt wenig Aussagen darüber, was eine bestimmte Maßnahme an Nutzen bringt und unter welchen Rahmenbedingungen man diese einsetzen sollte. Daher wäre es sinnvoll, Maßzahlen für QS-Methoden zu ermitteln.
- **Leitfaden für Unternehmen zur Auswahl und Einführung von QS-Maßnahmen**  
Welche QS-Maßnahmen für ein Unternehmen sinnvoll sind, kann man nicht pauschal definieren, d.h. nachdem man Bewertungskriterien für QS-Maßnahmen gefunden hat, müssen diese an den Gegebenheiten und unterschiedlichen Anforderungen in den Unternehmen gespiegelt werden. Ein Leitfaden sollte erarbeitet werden, welcher den Aufwand für die Unternehmen bei der Auswahl und Einführung entscheidend minimieren würde.
- **QS-Modell für Eingebettete Systeme**  
Ausgehend von vorhandenen Vorgehensmodellen (für Standardsoftware) sollte ein an die Anforderungen von Eingebetteten Systemen angepasstes QS-Modell entwickelt werden, welches an die Projektanforderungen adaptierbar ist.
- **Methoden für die entwicklungsbegleitende QS**  
Der größte Aufwand der Qualitätssicherung liegt heute am Ende des Entwicklungsprozesses, nämlich beim Integrations- und Systemtest. Dies liegt aber im kritischen Projektpfad und stellt somit hohe QS-Risiken, daher bedarf es den Einsatz neuer Technologien um in den frühen Entwicklungsphasen effektivere Prüfungen durchführen zu können.
- **Systematische Ableitung und Beschreibung von Testfällen**  
Das Ermitteln und Beschreiben von Testfällen für Eingebettete Systeme stellt eine große Herausforderung dar, welche es noch zu bewältigen gilt. Ein sinnvoller Ansatz dazu wäre, eine geeignete Beschreibungstechnik zu entwickeln, welche die Besonderheiten von Eingebetteten Systemen berücksichtigt und hinreichend einfach anwendbar ist.
- **Testautomatisierung und Wiederverwendung**  
Die Eigenschaften eingebetteter Systeme und die Anforderung nach einer hohen Qualität ziehen umfangreiche Tests und komplexe Testbedeutungen nach sich. Um die praktische Anwendbarkeit zu gewährleisten, muss ein besonderer Augenmerk auf die Wiederverwendung und Automatisierung gelegt werden.

## 0.5 Konfigurationsmanagement

Die durchgeführten Interviews haben ergeben, daß etwa 25% der befragten Unternehmen gar kein Konfigurationsmanagementsystem einsetzen, etwa 50% der Unternehmen setzen einfache Konfigurationsmanagementsysteme, wie PVCS, CVS, SourceSafe, etc., ein und weitere 25% sehr moderne Systeme wie ClearCase.

Konfigurationsmanagementsysteme sind in der Softwareentwicklung als ein unentbehrliches Hilfsmittel zur Koordination von Entwicklerteams seit einigen Jahren weitgehend etabliert. Sie haben sich insbesondere zur Verwaltung von Quelltexten sehr bewährt. Durch den Einsatz von Konfigurationsmanagementverfahren, wie der Verwaltung von Zugriffssperren, der Verwendung von optimistischen Sperrverfahren und Entwicklungs-Sandboxes, oder gar einer einfachen Work-Flow-Unterstützung, können viele Reibungsverluste bei der Softwareentwicklung vermieden werden, die Produktivität der Entwickler kann entscheidend verbessert werden und auch große Projekte mit mehr als 20 Entwicklern und mehreren 100000 Quelltextzeilen werden beherrschbar.

Für den Bereich der eingebetteten Systeme sind diese Ergebnisse aber leider nur eingeschränkt übertragbar. Weitergehende Konfigurationsmanagementtechniken, wie optimistische Sperr- und Mischverfahren, stehen im allgemeinen nur für rein textuelle Dokumente zur Verfügung. Desweiteren basieren diese Verfahren auf einer weitgehenden automatischen Konsistenzprüfung wie sie bei der Softwareentwicklung vom Compiler vorgenommen wird. Dabei wird mit hoher Sicherheit festgestellt, ob parallel zu einander durchgeführte Änderungen verschiedener Entwickler an unterschiedlichen (oder sogar an den selben) Dateien sich widersprechen oder ob diese nebenläufigen Änderungen Konflikte erzeugen, die der manuellen Überprüfung bedürfen.

Im Bereich der eingebetteten Systeme stellen Quelltexte nur einen (kleinen) Teil der gesamten Entwicklungsdokumente dar. Ebenso wichtig für die gesamte Funktionalität sind CAD-Zeichnungen, Funktionsskizzen, Schaltpläne für die Elektronik, Bauteilspezifikationen, Anschlusspläne, Platinenlayouts und Bestückungspläne. Die Mehrzahl dieser Dokumente liegt in werkzeug- oder domänenspezifischen Dateiformaten vor, die sich einer zentralen Konfigurationsverwaltung und zentralen Mischverfahren entziehen. Schlimmer noch, eine übergreifende automatische projektweite Konsistenzsicherung fehlt im Bereich der eingebetteten Systeme aufgrund der vielen proprietären Dateiformate bisher vollständig. Das bedeutet, wenn ein Entwickler A eine mechanische Komponente modifiziert dann müssen die Auswirkungen dieser Änderung auf Verdrahtungs- und Schaltpläne und die daraus entstehenden Auswirkungen auf Softwarekomponenten manuell an die jeweils zuständigen Entwickler weitergeleitet und von denen manuell nachgezogen werden. Dabei kommt es oft vor, dass Änderungsvorschläge von davon betroffenen Entwicklern aus anderen Bereichen später abgelehnt oder modifiziert und zurückgegeben werden. In einem großen Projekt mit vielen Entwicklern werden täglich dutzende solcher Änderungsprozesse in Gang gesetzt (teilweise widerrufen und neu initiiert). Dies führt schnell zu sich überholenden und widersprüchlichen Änderungsvorgängen und nach kurzer Zeit weiß niemand mehr, welcher Änderungsstand von welchem Dokument welche Änderungen schon enthält und mit welchem Stand der anderen Dokumente zusammen passt. Dies verursacht hohe Reibungsverluste.

### **Lösungsansätze:**

Zur Lösung der Probleme im Bereich Konfigurationsmanagement für eingebettete Systeme sollte man kurz- bis mittelfristige Schritte und mittel- bis langfristige Ansätze betrachten:

- Adaption und Ausweitung des Konfigurationsmanagements

Kurz- bis mittelfristig sollten die bestehenden Möglichkeiten und Techniken des Konfigurationsmanagements und die Erfahrungen aus dem Bereich der reinen Softwareentwicklung so weit wie möglich auf das Gebiet der eingebetteten Systeme übertragen werden. Hierzu gehört die flächendeckende Einführung von Konfigurationsmanagementsystemen, die Ausdehnung des Konfigurationsmanagements von der Quelltextverwaltung auf die Verwaltung aller möglichen Projektdokumente und vor allem der vermehrte Einsatz von optimistischen Sperrverfahren und Sandbox-Techniken. In diesem Bereich ist aber auch die Schulung der

Mitarbeiter von zentraler Bedeutung. Konfigurationsmanagementsysteme greifen direkt oder indirekt in die Organisation und Arbeitsweise von Projekten und ihren Mitarbeitern ein. Dies führt zur Umstellung von gewohnten Arbeitsabläufen und erfordert eine umfangreiche Vorbereitung der einzelnen Mitarbeiter vor und bei der Einführung neuer Konfigurationsmanagementverfahren.

- **Interdisziplinäre Konsistenzprüfungen**

Mittel- bis langfristig müssen die zentralen Probleme des Konfigurationsmanagements im Bereich der eingebetteten Systeme angegangen werden. Dies sind die projektweite automatische Konsistenzprüfung (wie sie im Bereich der Softwareentwicklung durch den Compiler weitgehend sicher gestellt wird). Eine solche projektweite Konsistenzprüfung muss insbesondere disziplin- und dokumenttypübergreifend funktionieren, damit auch die Arbeit von Maschinenbauern, Elektrotechnikern und Informatikern gezielt koordiniert werden kann. Hierzu ist es erforderlich, dass die verschiedenen Hersteller von Entwicklungswerkzeugen dazu bewegt werden, ihre Dateiformate offen zu legen beziehungsweise Standarddateiformate zu verwenden. Basierend auf solchen Industriestandards für die verschiedenen Dokumentarten im Bereich der eingebetteten Systeme können dann allgemeine, domänenübergreifende Konsistenzprüfungswerkzeuge entwickelt werden. Die Entwicklung von domänenübergreifenden Konsistenzprüfungswerkzeugen kann aber auch so schon für einige verbreitete Werkzeuge und Dokumentarten in Angriff genommen werden. Mit der projektweiten Konsistenzprüfung einher sollte eine verbesserte Anbindung der verwendeten Entwicklungswerkzeuge an die Konfigurationsmanagementsysteme erreicht werden. Dies kann in einfachen Fällen durch geeignete Benachrichtigungs-Mechanismen geschehen, wodurch das Konfigurationsmanagementsystem Kenntnis darüber erhält, wer mit welchem Werkzeug auf welche Dokumente zugreift und möglicherweise entsprechende Zugriffssperren vergeben kann. Zur Unterstützung von optimistischen Sperrverfahren und Sandbox-Techniken ist aber auch die Bereitstellung von geeigneten Mergeverfahren für die verschiedenen Dokumentarten wünschenswert.

## **0.6 Beschreibungstechniken und Werkzeuge**

Über alle Klassifikationen hinweg werden nur isolierte Werkzeuge und isolierte Beschreibungstechniken verwendet, hauptsächlich zur Veranschaulichung beziehungsweise einfacheren Darstellung bestimmter Sachverhalte sowohl in der Kommunikation der Entwickler untereinander, als auch bei der Erstellung von Dokumentationen.

Die eingesetzten Maßnahmen sind unzureichend, da es den Entwicklern meist völlig freisteht, welche Beschreibungsmittel und welche Werkzeuge sie einsetzen und wenn doch Vorschriften existieren, dann ist meist der Grad, bis zu dem sie angewendet werden müssen nicht festgelegt.

Die Verwendung von Beschreibungstechniken für über Dokumentation und Kommunikation hinaus gehende Zwecke, wie beispielsweise phasenübergreifende Generierung, hängt stark mit der Verfügbarkeit und Anwendbarkeit entsprechender Werkzeuge zusammen. Die Mehrzahl der Interviewten setzen jenseits von Textverarbeitung und Entwicklungsumgebung keine weitergehenden Werkzeuge ein. Weitere Ergebnisse waren:

- Einsatz von Beschreibungstechniken hauptsächlich in der Analyse und im Grobdesign
- Favorisierte Beschreibungstechniken sind Automaten, Architekturdarstellungen und Klassendiagramme (nur zusammen mit Werkzeug), Einsatz von UML nur bei Struktur (1/3), nicht bei Verhaltensbeschreibung.

- Eine Modularisierung der Systementwicklung findet wenn überhaupt nur informell statt (nicht durch besondere BT oder Werkzeuge unterstützt)
- Kein effektiver Einsatz von Codegenerierung
- Echtzeitanforderungen spielen eine wichtige Rolle, werden aber oft nicht methodisch erfasst

Ursachen sind einerseits fehlende Kenntnis der Möglichkeiten, oder Ablehnung moderner Techniken, andererseits die fehlende Durchgängigkeit der Werkzeuge, deren schlechte Qualität, unwirtschaftlichkeit und mangelnde Berücksichtigung der Anforderungen von eingebetteten Systemen.

Der Systementwurf insgesamt findet also weitestgehend informell und wenig detailliert statt. Viele Entwurfsentscheidungen sind schlecht oder gar nicht dokumentiert, weshalb es auch wenig verwunderlich erscheint, daß Wiederverwendung, oder bausteinorientierte Entwicklung so gut wie keine Rolle spielt. Weitere ermittelte Ursachen waren:

- Unterschiedliche Endkundenanforderungen
- Wenig Werkzeuge für Analyse und Test
- Keine phasenübergreifende Vorwärts- und Rückwärtsverfolgbarkeit
- Übergewicht Feindesign/ Implementierung

Abschließend ist festzustellen, daß nur etwa 30% der Befragten alle wesentlichen Eigenschaften des Systems beschreiben. Nur etwa 5% der Befragten liefern eine vollständige nicht informelle Beschreibung ab, etwa die gleiche Anzahl Befragter gab an, sich komplett auf informelle Beschreibung zu stützen, oder überhaupt nichts zu dokumentieren.

Eine vollständige integrierte und durchgängige Beschreibung etwa für automatische Überprüfung der Stimmigkeit findet sich nirgends

### **Lösungsansätze:**

Für die Lösung der vorherrschenden Probleme im Bereich des Werkzeugeinsatzes und der Beschreibungstechniken bieten sich folgende Ansätze an:

- Bestehende Vorurteile und Widerstände abbauen - Wirksamkeit beweisen

Die Realisierung eines gewünschten Vorgehens steht und fällt mit der Akzeptanz der einzusetzenden Mittel und genau hier liegt eines der grundlegendsten ermittelten Probleme.

Eine mittelfristige Lösungsmöglichkeit besteht daher darin, die Wirksamkeit und möglicherweise erst zukünftige Bedeutung zu beweisen. Hier ist auch die Wissenschaft in der Pflicht, nicht nur Grundlagen in Form von Beschreibungstechniken und Methoden zu erforschen, sondern auch deren Wirksamkeit im realen Einsatz zu erproben. Möglicherweise müssen auch erst noch Techniken erfunden oder verbessert werden, die diese Art von Wirksamkeitsbeweisen methodisch ermöglichen.

- Bestehende Nischen- und Speziallösungen nutzen

Einige Firmen haben für ganz spezifische Anforderungen bereits eigene Methoden oder Werkzeuge entwickelt. Es fehlen jedoch die Mittel, diese improvisierten und aus pragmatischen Ansätzen entstandene Produkte kurzfristig so weit zu verbessern, daß sie auch von anderen genutzt werden könnten, die ähnliche Anforderungen haben.

- Erfahrungsaustausch und Evaluierung bestehender Methoden und Werkzeuge

Ein großes kurzfristiges Verbesserungspotential liegt in der Ausnutzung von Synergieeffekten bei jeweils firmeninternen Evaluierungen bestehender Methoden und Werkzeuge. Momentan werden jeweils eigene beschränkte Mittel nur dafür eingesetzt, einen kleinen Ausschnitt der vorhandenen Möglichkeiten für zu begutachten. Gesammelt und allen zugänglich gemacht, ergeben die vielen Einzelerfahrungen einen guten Überblick über den gesamten Markt der Möglichkeiten.

- Multidisziplinäre Werkzeug & Methodenintegration durch mehr Abstraktion

Langfristig erscheint der Einsatz von Forschungsmitteln für die bessere Berücksichtigung von domänenspezifischer Besonderheiten in Beschreibungstechniken oder in Werkzeugen wenig sinnvoll, da der volkswirtschaftliche Nutzen im Vergleich zu den eingesetzten Mitteln nur sehr gering sein wird. Eine rein technische Integration und die Beschränkung auf die Berücksichtigung domänenspezifischer Besonderheiten würde beim augenblicklichen Zersplitterungsgrad der Domänenanforderungen unweigerlich zu noch umfangreicheren und teureren Werkzeugen führen, und damit nicht ausreichend zur Steigerung der Produktivität der Anwender führen, da die Komplexität der Werkzeuge überproportional gegenüber der Steigerung möglicher Anwender wachsen würde.

Zunächst sollte statt dessen versucht werden, den Entwicklungsprozess für Software auf das gleiche ingenieurmäßige Niveau zu heben, wie es in anderen Disziplinen z.B. bei der Hardwareentwicklung zu finden ist. Sind die Grundlagen geschaffen, die eine Entwicklung bedarfsgerechter integrierter Werkzeuge mit vertretbarem Aufwand und für einen größeren Markt erlauben, reicht der bereits existierende Bedarf bei weitem aus, um durch die Kräfte des Marktes zu qualitativ hochwertigen und günstigen Werkzeugen zu kommen.

In einem zweiten Schritt sollte versucht werden, das Abstraktionsniveau noch weiter anzuheben und die verschiedenen Disziplinen, die von der Softwareentwicklung für ES berührt werden unter einem gemeinsamen Vorgehensmodell zu vereinen, beispielsweise einem digitalen Produktmodell, das alle Aspekte eines Produktlebenszyklus in sich vereint und die verschiedenen Einzelvorgehensmodelle miteinander integriert.

## **0.7 Strategische Szenarien - die Anforderungen von morgen**

Von strategisch hoher Bedeutung ist das Ermitteln der Anforderungen von morgen für die Entwicklung, Produktion und den Service von Software in eingebetteten Systemen sowie die Erschließung zukünftiger Erfolgspotenziale. Die Betrachtung und Lösung der heute offenstehenden Probleme reicht angesichts der sehr dynamischen technologischen Entwicklung mit Sicherheit nicht aus, die Herausforderungen der Zukunft zu bewältigen. Im Rahmen der VA wurden daher Zukunftsszenarien entwickelt, um heute wahrnehmbare Entwicklungen zu antizipieren und zu schlüssigen Zukunftsbildern (Szenarien) zu verknüpfen. Wichtig ist in diesem Zusammenhang, eine größere Anzahl von Einflussfaktoren aus den Bereichen Technologie - insbesondere Softwaretechnologie und Kommunikationstechnologie -, Branchen und Märkte und deren Vernetzung zu berücksichtigen sowie mehrere Entwicklungen eines Einflussfaktors ins Kalkül zu ziehen. Als Betrachtungszeitraum für diese Entwicklungen liegt den Szenarien das Jahr 2008 zugrunde. Für die einzelnen Anwendungsdomänen wurden Schlüsselfaktoren bestimmt, die besonders stark eingebunden sind in die Vernetzung der Einflussfaktoren. Für diese Schlüsselfaktoren wurden anschließend Zukunftsprojektionen für das Jahr 2008 im Rahmen von Workshops ermittelt. Die anschließende Konsistenzanalyse führt zu den einzelnen Szenarien. Die wesentlichen Stoßrichtungen der Szenarien ergeben sich aus dem Vergleich der Unterschiede der Szenarien einer Anwendungsdomäne.



## **Anlagen- und Maschinenbau:**

Die Szenarien für den Anlagen- und Maschinenbau verdeutlichen unterschiedliche Zukünfte. Neben der Entwicklung der Informations- und Kommunikationstechnologie hängt der Erfolg zukünftiger Anlagen und Maschinen im wesentlichen an effizienten Methoden und Werkzeugen des Softwareengineerings, der Modularisierung der Produkte sowie dem Vorhandensein und Einhalten von weitreichenden Standards. Differenzierungen sind vor allem hinsichtlich betroffener Komponenten und Anwendungen zu erkennen. Sicherheitsrelevante Funktionsgruppen mit Echtzeitanforderungen können z.B. erst langsam aufgrund der notwendigen Zuverlässigkeit IT-mäßig vernetzt werden. Kostspielige IT-Anwendungen für den Sondermaschinebau sind eher denkbar bei vergleichsweise hohen Entwicklungskosten der Softwarefunktionalität.

Neben den Produkten selbst sind aber auch die Umfeldentwicklungen wie Verfügbarkeit von effizienter Softwareentwicklungsumgebungen oder wirksame Einhaltung des Patentrechts zu betrachten. Sie stellen entscheidende Weichen hinsichtlich der Konsistenz der möglichen Zukunftsbilder. Darüber hinaus muss es der Branche gelingen, ingenieurmäßiges Standesdenken der Disziplinen durch Offenheit, breitere Bildung sowie Verknüpfung von Informatik und Maschinebau entgegen zu treten. Letztendlich aber entscheiden die Kunden mit ihren Qualitätsanforderungen über den Maßstab der Entwicklungsanstrengungen.

## **Automatisierungs- und Produktionstechnik:**

Die Szenarien für die Automatisierungs- und Produktionstechnik verdeutlichen unterschiedliche Zukünfte. Die konsistenten Entwicklungen der Schlüsselfaktoren zeigen ein gespaltenes Bild, in dem vor allem die Spannungsdreiecke „Qualität-Produktkomplexität-Preis“ und „Beherrschung von Software-Zuverlässigkeit-Anbieter“ die entscheidenden Rollen spielen. Insbesondere die Beherrschung von Software gepaart mit den rasanten Entwicklung auf dem Gebiet der Sensorik und der Kommunikationstechnik erscheinen als Notwendigkeit für die Verbreitung von autonomen mechatronischen vernetzten Systemen und damit zu einer Modularisierung sowie auch Standardisierung von Lösungselementen. Ihre Durchdringung in den Bereich der echtzeitfähigen Systeme hängt dabei im wesentlichen von der zuverlässigen und wirtschaftlichen technischen Machbarkeit ab. Dahingegen kann eine Software-Krise im Bereich der eingebetteten Systeme viele wirtschaftlich notwendige Innovationen behindern bzw. verzögern. Aus wettbewerblicher Sicht hätte dies starke Konzentrationsbemühungen der Anbieter zu Folge. Gar könnten die Kunden im Business-to-Business bewusst auf weniger Informationstechnologie setzen, um ihre Produkte noch an Endkunden vertreiben zu können.

## **Verkehrstechnik:**

Die Szenarien für die Verkehrstechnik verdeutlichen im wesentlichen drei unterschiedliche Zukünfte. Sie sind bestimmt durch das Spannungsfeld Hi-Tech-Einsatz, Sicherheitsstreben und Akzeptanz gegenüber technischen Systemen. Je nach Dominanz dieser Faktoren werden die Zukunftsbilder maßgeblich beeinflusst. Eine starke Stellung des Hi-Tech-Einsatzes führt zu einer Fortbewegung, die ständig durch alle Möglichkeiten der mobilen Informationsversorgung und Kommunikationstechnik begleitet wird. Deren Verbreitung setzt aber die sichere Beherrschung von IuK-Technologien sowie der Software-Entwicklung voraus. Überwiegt das Sicherheitsstreben der Gesellschaft werden Innovationen gebremst, standort-immanente Nachteile Deutschlands gestärkt und folgerichtig internationale Wettbewerbsfähigkeit eingebüßt. Als dritte Variante kann die Akzeptanz technischer Systeme die Zukunft der Verkehrstechnik entscheidend bestimmen. Gelingt der Umgang mit der Komplexität der Systeme und der Softwareentwicklung nicht zuverlässig, so kann dies zu einem Grundmißtrauen in der Gesellschaft führen. In Fragestellung des Verkehrsaufkommen und Forderung nach Transparenz können

konsistente Folgen darstellen. In den beiden letzten Fällen kann die Verkehrstechnik aber nur schwer als Motor für die gesamtwirtschaftliche Entwicklung dienen. Ihre Vorreiter Rolle wäre in Frage gestellt.

### **Fazit:**

Bei dem Übergang von den Ergebnissen der Szenario-Untersuchung zur Strategieentwicklung sind grundsätzlich zwei Sichtweisen möglich. Einerseits kann eine Strategie zukunftsrobust angelegt werden. In einem solchen Falle wären alle Maßnahmen gleichermaßen gut für eine der alternativen Zukünfte geeignet. Andererseits kann eine Strategie fokussiert angelegt werden, nämlich dann, wenn auf bestimmt notwendige Voraussetzung für eine oder mehrere Zukünfte abgehoben wird. Dies birgt selbstredend die Gefahr der Fehlentscheidung in sich. Bei den hier betrachteten eingebetteten Systemen besteht der Fortschritt auf der technischen und wirtschaftlichen Machbarkeit. Ihm entgegen treten aber Kundenanforderungen, die aus vielen unterschiedlichen Faktoren bestimmt werden. Diese können sich progressiv oder degressiv auf den Fortschritt auswirken. Vor diesem Hintergrund erscheint eine zukunftsrobuste Strategie nicht wirkungsvoll. Vielmehr sind internationale Faktoren ebenfalls ins Kalkül mit einzubeziehen. Daher wird hier eine Technologie-orientierte, fokussierte Strategie empfohlen.

Zusammenführend lässt sich feststellen, dass die Software-Technologie eine Schlüsselrolle hinsichtlich der Konsistenzfähigkeit der Szenarien einnimmt. Ihre Beherrschung in puncto Effizienz, Qualität, Ausbildung, Methoden und Werkzeuge sind unabdingbare Voraussetzung für die Erschließung der Erfolgspotenziale von morgen. Dabei spielen selbstredend bei eingebetteten Systemen aufgrund ihrer „Hardware-Nähe“ zu benachbarten Forschungsgebieten wie Mechatronik, Sensorik oder Kommunikationstechnik enge Verknüpfungen von denen die eingebetteten Systeme profitieren wie auch entsprechend gehemmt werden können.

Ohne Software-Technologie sind allerdings die Erfolgspotenziale, die sich durch die rasch entwickelnde Informations- und Kommunikationstechnik für den Maschinenbau und artverwandten Branchen ergeben nicht erschließbar.

## **0.8 Aktionsfelder**

Im folgenden werden sechs mögliche Vorschläge für Aktionsfelder aufgeführt, die sich aus den Lösungsansätzen und Szenarien, die im Rahmen der Vordringlichen Aktion ermittelt wurden, ableiten lassen. Die in Abschnitt erarbeiteten Lösungsansätze stellen technische Lösungen für die identifizierten Problemfelder dar, benötigen aber zu ihrer Realisierung Umsetzungsmaßnahmen. Die im folgenden beschriebenen Aktionsfelder stellen jeweils Umsetzungsmechanismen für ganze Bündel solcher Lösungsansätze da:

- **Adaption und Transfer:** Aufgabe ist die Einführung eines nach dem heutigen Stand der Technik und Wissenschaft möglichen Entwicklungsprozesses in kleinen und mittelständischen Unternehmen. Dabei soll diese Einführung in Form von geförderten Projekten mit repräsentativen Pilotunternehmen durchgeführt werden. Hauptaufgaben bei der Einführung eines Entwicklungsprozesses sind die Anpassung vorhandener Vorgehensmodell an die Bedürfnisse der Unternehmenszielgruppe der Vordringlichen Aktion, die Einführung von Konfigurationsmanagement und die Definition eines Qualitätssicherungsleitfadens für eingebettete Systeme.
- **Kompetenzzentrum:** Ziel dieses Aktionsfelds ist der Aufbau einer Einrichtung zur Koordination von Aus- und Weiterbildung der Softwareentwicklung für die Anwendungsbereiche eingebetteter Systeme. Wesentliche Funktion eines solchen Kompetenzzentrum ist es, als

zentrale Anlaufstelle, besonders für klein - und mittelständische Unternehmen, in software-technologischen Fragen zu dienen. Die Aufgabenstellung eines solchen Kompetenzzentrums umfasst insbesondere die Vermittlung der ingenieurmäßigen Softwareentwicklung, die Funktion einer Anlaufstelle für Transfer aktueller Entwicklung im Bereich Softwareentwicklung für eingebettete Systeme, einer Innovationsdrehscheibe für Diskussion aktueller Trends und Herausforderungen sowie eines Beratungsgremiums zur Unterstützung von Studiengangreformen.

- **Koordination Werkzeugentwicklung:** Aufgabe dieses Aktionsfeldes ist es, eine Kontaktplattform für Werkzeughersteller und KMUs zu schaffen. Diese Kontaktplattform soll als vermittelnde und moderierende Schnittstelle zwischen den ansonsten gegenüber Werkzeugherstellern nicht ausreichend repräsentierten kleinen und mittleren Unternehmen dienen. Die Anpassung an spezifische Bedürfnisse kann beispielsweise in Form von geförderten Pilotprojekten mit KMUs des Anwendungsbereichs erfolgen.
- **Unternehmensübergreifende Standardformen:** Schwerpunkt dieses Aktionsfeldes ist die Schaffung und Bewertung von Standards im Bereich eingebetteter Software. Hierbei steht nicht die Schaffung von Normen, sondern die Erstellung von direkt umsetzbaren Vorlagen und Grundkonzepten für die Unternehmen im Vordergrund. Wesentliche Teilaufgaben sind hierbei die Erfassung von Best-Practice-Ansätzen, die Definition von Standards für Dokumenten- und Beschreibungsformen (z.B. Lasten- und Pflichtenheft, Prüfpläne), von Standardsystemarchitekturen und Standardsystembausteine (z.B. Kommunikationsstruktur, Fehlerbehandlung) und von Standards für Plattformen (z.B. Echtzeitbetriebssysteme) und Standardrealisierungsarchitekturen (HW und SW).
- **Integriertes Vorgehensmodell:** Ziel ist hier die Definition eines fach- und unternehmensübergreifenden Vorgehensmodells. Dieses langfristig angelegte Aktionsfeld stellt daher einerseits Bündelung der Ergebnisse der vorausgehenden Aktionsfelder dar sowie andererseits die Integration der Ergebnisse und die Erweiterung um phasen-, fach- und unternehmensübergreifende Aspekte. Dabei fallen folgende Einzelaufgabenstellungen an: Phasenübergreifende Entwicklung (Generierung, Rückverfolgung), Unternehmensübergreifende Entwicklung, Domänenübergreifende Entwicklung (Informatik, E-Technik, Maschinenbau), Konfiguration und Konsistenzsicherung, Kostenmanagement, entwicklungsbegleitende Qualitätssicherung, Bausteine und Standardarchitekturen, eine durchgängige Werkzeugkette sowie der Rückfluss in den Prozess (Wartung).

Ziel dieses Aktionsfeldes ist die Bereitstellung neuer Technologien der Informationsverarbeitung für die Anwendung im Maschinen/Anlagenbau, der Produktions/Automatisierungs/Verkehrstechnik, z.B. Netzwerktechnologien (inkl. Internet, Jini, etc.), intelligente Bausteine und adaptive Systeme, OO-Technologien (inkl. CORBA etc.), Virtuelle Produktentwicklung. Verbunden mit diesen Themenstellungen ist naturgemäß deren Integration in die anderen Aktionsfelder, insbesondere - wegen des langfristigen Charakters - in das integrierte Vorgehensmodell. Damit gehen jeweils spezifische Fragestellungen einher, beispielsweise deren Übernahme in Standardarchitekturen oder Bausteine, Beschreibungsformen und Werkzeuge, Fragen der Qualitätssicherung oder Konfiguration.

Die vorgestellten Aktionsfelder stellen jedoch **keine** Projektdefinitionen dar. Es lassen sich je nach Aufwand und Ressourcen unterschiedliche Schnitte durch diese Themenfelder legen, um entsprechende Realisierungsprojekte zu definieren. Ein Beispiel für einen Schnitt ist das Projekt „Einführung eines bausteinorientiertes Vorgehensmodell im Bereich Anlagenbau“, mit den betroffenen Aktionsfeldern Integriertes Vorgehensmodell mit Schwerpunkt Bausteinorientierung, Unternehmensübergreifende Standardformen mit Schwerpunkt

Standard-SW-Architektur für den Anlagenbau, Adaption und Transfer unter Verwendung des entwickelten Vorgehensmodells sowie die Überführung in ein Kompetenzzentrum nach Erprobung des entwickelten Vorgehensmodells mit Pilotunternehmen.