

**Initiative „Kommunikation in der Softwareentwicklung“**

—

**Softwaresysteme als Kommunikationsthema**

## **Position Statements**

Dezember 2003

Herausgeber: Prof. Dr.-Ing. Siegfried Wendt, Dipl. Ing. Bernhard Gröne  
Arbeitsgruppe „Modellierung Software-intensiver Systeme“  
am Hasso-Plattner-Institut für Softwaresystemtechnik (HPI), Potsdam

## Inhalt

Einleitung.....	3
Das Anschreiben.....	4
Zusammenfassungen.....	4
<b>Die Position Statements</b>	
<b>SAP AG (Peter Zencke)</b>	
Communication in Software Development.....	7
<b>Beta Systems Software AG (Oskar von Dungern)</b>	
Ingenieurmäßige Entwicklung komplexer Software-Systeme.....	11
<b>Robert Bosch GmbH – Gasoline Systems (Taskin Ege)</b>	
Position Statement.....	13
<b>Intervista AG (Reinhard Irsigler)</b>	
Positionen-Papier.....	15
<b>IBM Deutschland GmbH (Rainer Hofmann)</b>	
Kommunikation mit Software im Kontext des IBM-SAP-Umfelds.....	16
<b>Aviso GmbH (Jürgen Weinreich)</b>	
Position Statement.....	18
<b>IZB Soft/ Sparkassen- und Giroverband (Michael Weigoldt)</b>	
Softwaresysteme als Kommunikationsthema.....	20
<b>Deutsche Telekom (Jürgen Brunn)</b>	
Position Statement.....	21
<b>Deutsche Post ITSolutions (Andreas Kersten)</b>	
Position Statement.....	23
<b>Lufthansa Systems Airline Services GmbH (Thomas Kullmann)</b>	
Kommunikation als Notwendigkeit im Betrieb von komplexen Softwaresystemen.....	24
<b>spm technologies (Guido Laures)</b>	
Softwaresysteme als Kommunikationsthema.....	25
<b>modulo3 GmbH (Michael W. Dietrich)</b>	
Software-Systembeschreibungssprachen vor FMC.....	27
<b>IBM Deutschland Entwicklung GmbH (Jörg Michael Thielges)</b>	
Kommunikation als Wertschöpfungsfaktor in der Softwareentwicklung.....	29
<b>Fraunhofer Gesellschaft (Herbert Weber )</b>	
Software-Entwicklung in Communicating Communities.....	30

## Einleitung

Immer wieder kommen aus der Praxis der Softwareentwicklung und -wartung Aussagen, nach denen die ernsthaften Probleme nicht technischer Art seien und ein Grundproblem darin besteht, dass die Kommunikation zwischen den Beteiligten nicht funktioniert. Die Wissenschaft aber befasse sich nur mit den technischen Problemen und lasse die Praktiker im Punkt der Kommunikation alleine stehen. Ende 2002 starteten Siegfried Wendt und Bernhard Gröne die Initiative KommSE, „Kommunikation im Software Engineering“ bzw. Softwaresysteme als Kommunikationsthema (Der Text der Initiative befindet sich im nächsten Abschnitt). Verantwortliche von rund 60 Unternehmen, die mit Software-Entwicklung und -Betrieb zu tun haben, wurden angeschrieben und um eine Stellungnahme zu diesem Thema gebeten. Dieser Band enthält die Position Statements, die bis Anfang Oktober 2003 eingegangen sind. Einige Unternehmen scheuten sich allerdings davor, eine Stellungnahme abzugeben, insbesondere wenn in diesen Unternehmen Software als Produkt oder als Kundenprojekt gemacht wird.

In ihrer Beziehung zu Entwicklung und Betrieb von Softwaresystemen kann man die Unternehmen in verschiedene Kategorien einteilen:

- **Software als eigenständiges Produkt**  
Mit dem Erzeugen und Vermarkten eines Softwareprodukts ist es für diese Firmen meist nicht getan. Neben Support kommen beim Kunden oft auch Einführungsprojekte und teilweise die Integration in ein bestehendes System dazu.
- **Software als Produktkomponente**  
Diese Firmen stellen Geräte her, in denen in zunehmenden Maße Software eine Rolle spielt. Während früher ein Ingenieur die Software alleine beherrschen konnte, sind heute größere Teams, zum Teil verteilt, damit befasst. Die Kommunikation zwischen Hardware- und Software-Ingenieuren ist hier von Anfang an wichtig.
- **Kundenspezifische Software als Projektergebnis**  
In dieser Kategorie ist die Kommunikation mit dem Kunden sehr wichtig, da dessen Anforderungen im Ergebnis, der Software, berücksichtigt werden müssen. Da die Kosten stark von den Anforderungen abhängen, müssen wiederum die Techniker und die Betriebswirte der Projektfirma sich verstehen.
- **Software für den internen Gebrauch**  
Die Firmen haben viele Standorte und daher zum Teil komplexe interne Systeme, für die sie eigene Abteilungen zur Entwicklung und Betrieb haben.
- **Beratung bei Softwareprojekten**  
Beratungsfirmen werden oft hinzugezogen, wenn Projekte ins Schlingern geraten und zu scheitern drohen. Die Analyse der Gründe, die ein Projekt gefährden, gehört damit auch zum Geschäft dieser Firmen.

Eine Einteilung einer Firma in genau eine Kategorie ist allerdings oft nicht möglich.

Eine weitere Kategorisierung betrifft die verteilte Entwicklung, die ihrerseits den Bedarf an Kommunikation potenziert. Bei weltweiter Verteilung kommen dann noch Sprach- und Kulturbarrieren dazu.

## Das Anschreiben

„Bei der arbeitsteiligen und besonders bei der verteilten Entwicklung softwareintensiver Systeme treten zwei große Probleme auf: Zum einen ist die Organisation der Entwicklung eine schwierige Aufgabe, zum anderen erfordert die arbeitsteilige Entwicklung eines komplexen Systems eine intensive Kommunikation, die häufig nur mit einem geringen Wirkungsgrad erfolgt.“

Tom De Marco schreibt 1997 in seinem Buch „Warum ist Software so teuer?“, „Bei der Software-Entwicklung geht es vor allem darum, miteinander zu reden und Dinge niederzuschreiben. Diejenigen, die die Disziplin am stärksten voranbrachten, waren in aller Regel die besten Kommunikatoren, nicht die besten Techniker ihres Fachgebiets.“

Mit dem Problembereich des Entwicklungsprozesses befasst sich heute das Software Engineering, während der Kommunikation über den Entwicklungsgegenstand – nämlich das komplexe System – im akademischen Umfeld bisher zu wenig Bedeutung zugemessen wird. Hier können Impulse aus der Industrie bewirken, dass die Wissenschaft in breiterem Rahmen nach Lösungen forscht.

Aus diesen Gründen haben wir zu Beginn dieses Jahres die Initiative „Kommunikation in der Softwareentwicklung“ gestartet. Ziel dieser Initiative ist die Gewinnung von Mitkämpfern, um die Probleme auf diesem Gebiet durch Erfahrungsaustausch, Verbreitung des erforderlichen Problembewusstseins und der Suche nach Lösungen anzugehen.

Es soll eine Sammlung von Positionsaussagen entstehen, mit der die akademische und politische Öffentlichkeit auf die Relevanz des Problemfeldes aufmerksam gemacht wird und worauf man sich bei der Planung anschließender Aktivitäten berufen kann. In ihren Textbeiträgen sollen die Autoren ihre Sicht des Problems darstellen, ihre Erfahrungen skizzieren und nach Möglichkeit andeuten, welche Aktivitäten sie zur deutlichen Verbesserung der Situation für geeignet halten. Diese Sammlung von Positionsaussagen soll im Laufe der kommenden Monate veröffentlicht werden.“

## Zusammenfassungen

### Firmen, die Software als eigenständiges Produkt herstellen.

P. Zencke von der **SAP AG** stellt fest, dass bei der weltweit verteilten Entwicklung bei der SAP ein immer größerer Kommunikationsbedarf entsteht, während die Möglichkeiten zur Kommunikation mit der Entfernung und Sprach- und Kulturbarrieren immer eingeschränkter werden. Die Lösung besteht für ihn in einer Kombination aus Technologie zur Unterstützung der Kommunikation und ihrer Einbeziehung in die Entwicklungsorganisation und der Firmenkultur. Die Fähigkeit zur Kommunikation ist erlernbar und dieser Lernprozess erfordert Zeit. Ein Vorbild hierfür ist die Auto-Industrie. Ebenso ist ein formales Modell für die Entwicklungsprozesse notwendig. Zencke denkt hierbei an ein Team Maturity Model.....7

O. v. Dungern von **Beta Systems** hebt hervor, dass auch ein Produkt beim Kunden in einem Projekt eingeführt werden muss und dass im Kommunikationsprozess mit dem Kunden Zeichnungen die Verständlichkeit der Design-Dokumente und damit die Qualität der Kommunikation verbessern. Auch in der Entwicklung sind verständliche Darstellungen kein Hemmnis der Kreativität der Entwickler mehr, da es bei der herrschenden Komplexität erforderlich ist, dass mehrere erfahrene Entwickler gemeinsam einen Entwurf diskutieren, um möglichst viele Anforderungen abzudecken und Fehler erkennen – „Nur was sichtbar ist, ist verbesserungsfähig“. Obwohl Methoden und Notationen ausführlich beschrieben und praxiserprobt seien, werden sie in der Ausbildung an Hochschulen vernachlässigt.....11

### **Firmen, die Software als Teil ihrer Produkte machen**

Taskin Ege von der **Robert Bosch GmbH** sieht ein Defizit bei den Methoden der Software-Entwicklung, setzt aber große Hoffungen in das Reifegradmodell (CMM).....13

### **Firmen, die Kundenspezifische Software machen bzw. integrieren**

R. Isigler von der **Intervista AG** hebt die Wichtigkeit von Grafik für die Kommunikation aller am Lebensprozess von Software Beteiligten hervor.....15

R. Hofmann von **IBM Deutschland GmbH** führt die Komplexität der Kundensysteme auf die Vielzahl der verwendeten (für sich homogenen) Einzelkomponenten zurück. Um die Probleme beherrschen zu können, die oft erst bei der Integration beim Kunden auftauchen, werden Moderatoren benötigt, die die Kommunikation zwischen Anwendungs- und Systemarchitekten, Implementierern, Betreibern und Geschäftsprozessverantwortlichen unterstützen. Daneben sollen Tools eingesetzt werden, für die aber ein gewisser Schulungsbedarf nötig ist, und Standards beachtet werden, um die Integration zu erleichtern.....16

### **Firmen, die Software für den internen Gebrauch machen und betreiben:**

Für M. Weigold von **IZB Soft** sind technische Hilfsmittel zur Kommunikation alleine nicht ausreichend – auch das Finden der richtigen Abstraktion und die Verfügbarkeit am richtigen Ort ist wichtig, ebenso die Verwendung von Grafik. In diesen Punkten wird auch in der Ausbildung zu wenig gemacht.....20

J. Brunn von der **Telekom** sieht dagegen nur Probleme bei der Kommunikation zwischen den Fachleuten der Betriebswirtschaft und der Technik, weshalb in der Ausbildung neben Entwicklungsprozess auch Betriebswirtschaft und Projektmanagement vorkommen sollte.....21

A. Kersten von der **Deutschen Post ITSolutions** sieht für die Umstellung ihres System nur den Bedarf von Tools wie z.B. Knowledge Management Systeme.....23

T. Kullmann von **Lufthansa Systems** hebt den Bedarf für Kommunikation in ihrem Systemverbund hervor. Es fehle noch eine standardisierte Vorgehensweise und Beschreibungssprache. UML ist beispielsweise ungeeignet.....24

### **Firmen, die Softwareprojekte beraten**

Für G. Laures von **spm technologies** beginnen die Probleme schon mit der Aufteilung der Verantwortung auch fachlicher (betriebswirtschaftlicher) Seite, so dass ein Überblick über das System bzw. die Anforderungen an das System schwierig zu bekommen ist. Die Kommunikation im Entwicklerteam wird durch zu reglementierte oder formale Sprachmittel eher behindert. Da Text allein zu ungenau ist, sind grafische Baupläne hier hilfreich. In der Ausbildung sollten daher nicht nur formale Prozess- und Sprachmittel gelehrt werden, sondern auch der kreativen Akt beim Systementwurf unterstützt werden.....25

M. Dietrich von der **modulo3 GmbH** stellt fest, dass selbst Techniker trotz UML und MDA aneinander vorbei reden, von der Kommunikation mit den übrigen Beteiligten ganz zu schweigen. Die Forschung erkennt dieses Problem aber nicht.....27

### **Andere**

J. M. Thielges von der **IBM Deutschland Entwicklung GmbH** führt auch die schlechte Außenwirkung der Softwareentwickler und damit fehlende Wertschätzung ihrer Arbeit auf mangelnde Kommunikation und Diskussion zurück. Für Innovationen wird eine ganzheitliche Betrachtung der Systeme benötigt. Als Beispiele für Herausforderungen, die Kommunikation erfordern, nennt er Code-Wiederverwendung, Code-Wartung und Embedded Systems.....29

---

Für H. Weber von der **Fraunhofer Gesellschaft** wurde zu viel Gewicht auf den (formalen) Software-Entwicklungsprozess gelegt, der in der Praxis eher eine untergeordnete Rolle spielt. Kommunikation und Wissensaustausch sind wichtig und erfordern in der Forschung ein Zusammenspiel verschiedener Wissenschaftsdisziplinen.....30



THE BEST-RUN BUSINESSES RUN SAP

# Communication in Software Development

## 1. Challenge of Communication

### Internal study at SAP confirms importance of communication

Results of a Doctoral Thesis completed at SAP (Professor Mellis, University of Cologne, Germany, 1997-1998)

*Conditions*

- Problems understood; concerns known
- Personally motivated initiative
- Personal relationship with persons involved
- Minimum number of persons involved
- Homogeneous group of people involved

*Parameters*

- Objective understood and agreed to
- Minimum degree of formalization
- Broad basis for consensus
- Personal communication as medium
- Real support from executive management
- Constructive interaction; some resistance

### Communication: Challenge for software development at SAP for three main reasons...

1. SAP operates with globally organized development units working in virtual teams; we cannot limit questions on communication to onsite situation, must keep distributed development projects in mind (more details later)
2. SAP has a division of labor between different development organizations, so that products move among these organizations – information flow is essential for work efficiency:
  - Project-oriented development (NAD) with “pools” of competency
  - Product-oriented maintenance (IBD/IMS)
  - Industry-oriented field organization (sales, professional services, support)
3. SAP development is often driven by customers, both in standard and custom development. In addition to multisite development within SAP, there are additional challenges due to different technical backgrounds: “Yes, we expected a product like this, but...”

## 2. Challenge depends on complexity and setup

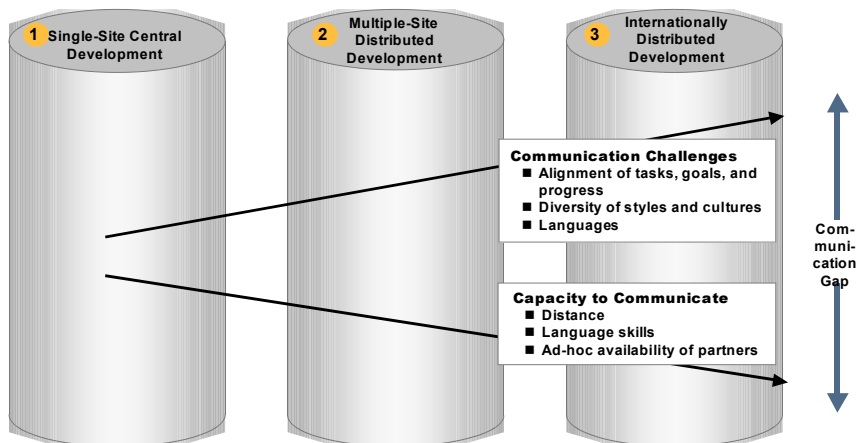


Figure 1 Communication challenges depend on sophistication of development setup

## **Distributed development projects: More challenges compared to single-site projects**

Distributed development (DD) projects fail for the same reasons as single-site development projects:

- Missed deadlines
- Quality problems
- Budget overrun
- Low employee morale
- Customer dissatisfaction

However, distributed development projects face two additional challenges:

- Distance
- Diversity

### **Distance as a challenge within projects**

**Distance**, the first challenge, has a limiting impact on information flow as soon as developers are not able to sit in the same room. Needless to say, the impact of distance is felt most strongly in projects distributed among remote locations that span the globe.

- While e-mail has made information exchange fast and easy, there is still a lot of information exchanged through more traditional and unstructured channels. Much of this information is not distributed systematically but happens by chance or circumstance.
- Another problem with distance is that the effects of mistakes in one remote part of the system are often not easily apparent until serious damage occurs. DD projects can be considered as feedback control systems with slower reaction times compared to centralized projects. In consequence, these projects are often less “forgiving” in terms of mistakes.

### **Diversity as a challenge within projects**

**Diversity**, the second challenge, has a variety of faces, for example, different...

- Languages
- Working styles
- Cultures
- Values

To deal with diversity, we need to account for, anticipate, and react to misunderstandings and communication gaps due to different cultural backgrounds, working styles, and language abilities.

### **Distance and diversity — the project dilemma**

Dilemma: We need to communicate more, but we have less capacity to communicate.

- Compared to a non-DD scenario, we need to communicate more to compensate for the potential communication gaps inherent to diverse environments.
- On the other hand, distance is a force that narrows the available spectrum and bandwidth of communication, thus limiting and slowing down the information flow among remote parties.
- That is why it is so important to balance distance and diversity in a DD project. Too much of both can easily induce or aggravate classic project mistakes.

We need to identify indicators and mitigation measures to prevent and handle the risks in DD projects.

### 3. Approach to overcome challenge

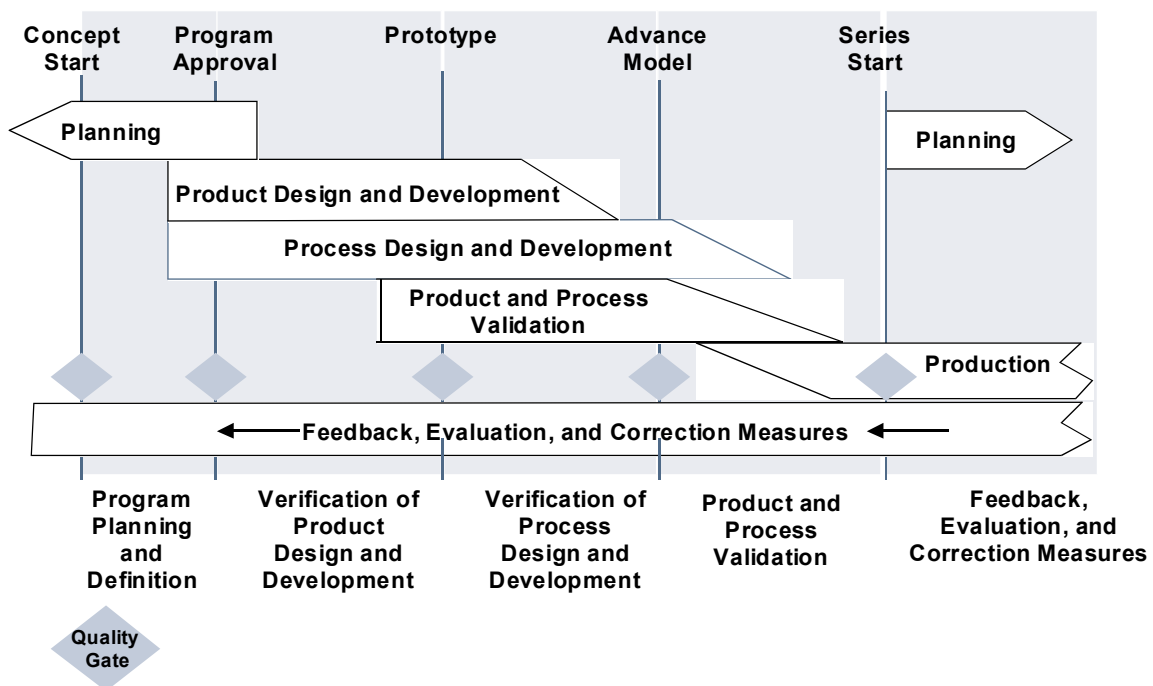
#### Solving problems requires technology and sociology

**Technical Approach:** The good news is that all variables are within the control of the organization – to a large extent. Communication needs can be kept in check by limiting complexity and interdependency among the various parts of the system – by organizational design.

**Sociological Approach:** It is also important to consider that building up communication capabilities is a process that takes time. Groups involved in distributed projects will go through successive stages of increased maturity with respect to communication capabilities. For example, trust and interpersonal chemistry play critical roles in how well a distributed group communicates – and these abilities do not usually appear overnight.

#### Technology approach requires a process model for collaboration

The discrete industry – specifically the automotive industry – underwent a change in mindset in the past few years. To reduce time-to-market, they developed a process of collaboration among car manufacturers and suppliers to develop a new version of a car and its components in a collaborative project approach – described in the advanced quality planning process, or APQP (see figure below).



**Figure 2 Automotive Industry's Advanced Product Quality Planning (APQP)**

Multilateral projects require shared procedures derived from a complete and comprehensive plan, and carried out through individual work packages. The model must include regular synchronization, so that projects are kept on track as follows:

- Work packages are synchronized with each other
- Work packages are synchronized with the overall plan

Such projects require setting clear goals and a stringent monitoring process to achieve the objectives.

#### Meta model required for objects and services

Projects – especially distributed ones – need a meta level of communication that leaves little room for misunderstandings because of language, culture, business background, and so on.

When project members differ throughout the project life cycle (development, maintenance, implementation projects), there is no chance for context-oriented communication.

→ We need formal written procedures that are independent of technical and sociological misunderstandings that support both collaboration and the transfer of know-how.

### Team Maturity Model

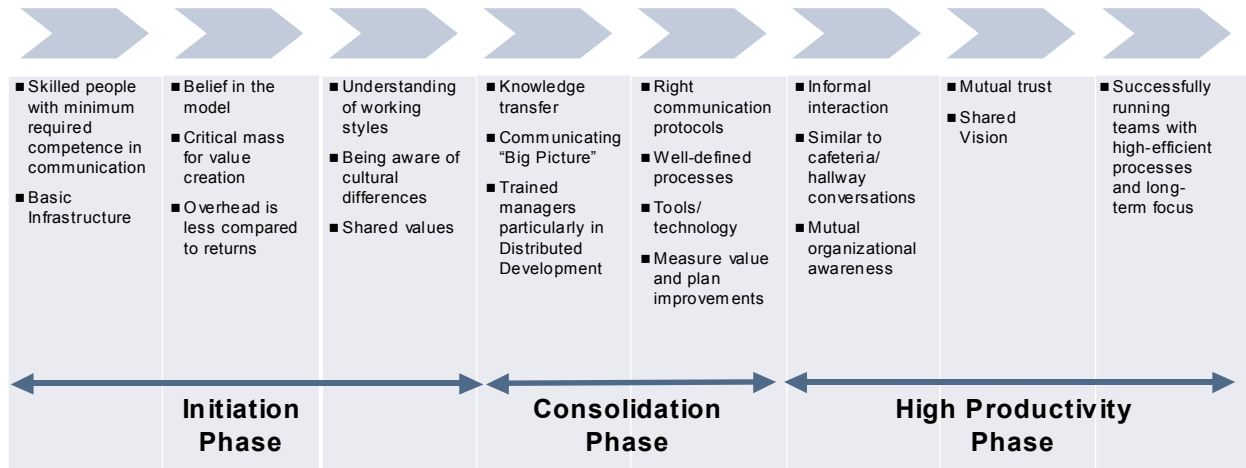


Figure 3 Relevant Soft Factors: Team Maturity Model

*Dr. Peter Zencke*

*Member of the Executive Board, SAP AG*

## Ingenieurmäßige Entwicklung komplexer Software-Systeme

Beta Systems Software AG mit Sitz in Berlin ist ein erfolgreicher Anbieter von Standard Software-Systemen für Großrechenzentren. Alle Kunden installieren von identischen Datenträgern. Die Produkte basieren auf einer gemeinsamen Architektur und nutzen eine umfangreiche Bibliothek von Reuse-Bausteinen. Anforderungen werden in Kundenarbeitskreisen abgestimmt und solche, die nicht im Interesse der Mehrheit der Kunden sind oder jenseits dokumentierter, releasefähiger Schnittstellen realisiert werden können, werden nicht umgesetzt. Dieser hohe Abstraktionsgrad ist ein wesentlicher Beitrag zum wirtschaftlichen Erfolg unserer Produkte in der Vergangenheit – in Zukunft sind jedoch zusätzliche Maßnahmen erforderlich.

In unseren Kundenbeziehungen – es handelt sich zumeist um Großkunden mit hohen Ansprüchen an Qualität und Zuverlässigkeit, sowohl hinsichtlich der Software selbst als auch hinsichtlich der Einführungsprojekte und Wartung – haben wir gelernt, dass heute die Einbettung unserer Produkte in unternehmensweite Prozesse und Softwarelösungen erforderlich ist. Dazu muss Standard-Software zumeist auf dem Wege eines individuellen Projektes ins jeweilige IT-Umfeld integriert werden. Im Vorfeld unterstreicht eine klare Kommunikation über die Projektinhalte mittels akzeptierter Design-Dokumente (Text mit „Zeichnungen“) die Kompetenz und stärkt das Vertrauen in den Anbieter. Zum Vertragsabschluß und während des Projektes ist die so gewonnene präzise Spezifikation der Gesamtlösung die Voraussetzung für eine ziel- und zeitgerechte Realisierung.

Hingegen zeigen zahlreiche Untersuchungen eine im Vergleich zu anderen Technik-Disziplinen sehr hohen Anteil an Software-Projekten, die entweder zu lange dauern und zu teuer sind, teilweise um ein Mehrfaches der ursprünglichen Schätzung, oder die sogar ganz und gar scheitern.

Einer der Gründe liegt darin, dass die Kommunikation zwischen IT-Nutzern und IT-Spezialisten zu unpräzise ist. Je unschärfer die Kommunikation, desto höher ist die Wahrscheinlichkeit, dass sich die Vertragspartner früher oder später mit unterschiedlichen Erwartungen gegenüber stehen. Projekterfolg und Kundenzufriedenheit sind also unmittelbar verknüpft mit der Qualität der Kommunikation auf der Basis verständlicher Dokumente.

### Handlungsbedarf der Software-Hersteller

Große Projekte werden erfahrenen Projektleitern anvertraut – und erfahrene Projektleiter stützen sich meist auf bekannte Vorgehensweisen ab. Zugleich herrscht ein hoher Zeit- und Kostendruck seitens der Auftraggeber. Ob sich an der Art Softwareprojekte abzuwickeln etwas ändert, hängt also davon ab, ob das eigene Management ausreichende (Zeit-) Ressourcen für die Ausbildung und Erprobung neuer Methoden gewährt.

Andererseits herrscht speziell bei eingefleischten Software-Entwicklern noch allzu oft ein „Kurzschluss zwischen Hirn und Tastatur“. Ein am Bildschirm ausgeführtes Programm ist dann das erste und einzige Arbeitsergebnis im Projekt, über welches diskutiert werden kann. Die Anwendung von Methoden und formalen Darstellungen in früheren Projektphasen hemmt nach Ansicht vieler Entwickler sowohl die Kreativität als auch die Effizienz. Während das bei einfachen Aufgabenstellungen vielleicht sogar zutrifft, ist es bei großen Softwareprojekten erwiesenermaßen falsch. Erst die Möglichkeit mit anderen Experten die Lösung im Entstehungsprozess detailliert zu diskutieren stellt sicher, dass keine Anforderungen vergessen, dass optimale Alternativen gewählt werden und dass schließlich alle Module zusammen passen. Auch eine schrittweise Verbesserung im Lebenszyklus einer Software ist nur dann möglich, wenn einzelne Entwicklungsschritte nachvollziehbar dargestellt werden: nur was sichtbar ist, ist verbesserungsfähig. Wenn also elegante Lösungen bereits auf den ersten Blick als solche erkennbar sind und wenn dieses Vor-

gehen erst einmal mit besonderen Erfolgen belohnt wird, wird auch erwähnter Kurzschluss einem gemeinsamen Erarbeiten der besten Lösung weichen.

### **Handlungsbedarf der Ausbildungsinstitutionen**

Vor einigen Jahren bauten wir ein Java-Entwicklungszentrum auf und suchten ein starkes Team junger Software-Entwickler. Ein wesentliches Kriterium für die Einstellung waren die Erfahrungen im Zusammenhang mit Software-Modellierung und Design-Methoden. Es war eine große Überraschung, dass die Informatik-Absolventen der Universitäten nur am Rande davon gehört, die Fachhochschul-Absolventen immerhin schon einen Kurs über das Thema belegt und Absolventen der Berufsakademien zur Hälfte bereits in Projekten gearbeitet hatten, in welchen jene Methoden zum Einsatz kamen. Wir hätten das Gegenteil erwartet! Diese Beobachtung hatte sicherlich keine statistische Relevanz, war aber doch auffallend. Kurz: wenn die ingenieurmäßige Beherrschung großer Softwaresysteme nicht zu einem zentralen Lerninhalt gemacht wird, wird die Software-Krise, die bereits vor 30 Jahren ausgerufen wurde, nicht überwunden. Geeignete Methoden sind ausführlich beschrieben, sie sind mittlerweile praxiserprobt und es gibt sogar weitgehende Einigkeit über die zu verwendenden Notationen.

Berlin, den 31.Mai.2003

***Dr.-Ing. Oskar von Dungen***

*Vorstand R&D  
Beta Systems Software AG  
[www.betasystems.com](http://www.betasystems.com)*



## Position Statement der Robert Bosch GmbH

Gibt es Kommunikationsprobleme speziell im Bereich der Entwicklung von Software-Systemen für Ottomotoren bei Bosch? Probleme sind in der Entwicklung mitunter an der Tagesordnung, doch sind diese in bedeutendem Maße einer mangelnden und nicht angemessenen Kommunikation zuzuordnen?

Zur Klärung dieser und ähnlicher Fragen gilt es im Vorfeld das Augenmerk auf mögliche Ursachen und Gefahren mangelnder Kommunikation im Bereich der Softwareentwicklung zu richten und alle sonstigen Gründe, wie z.B. sprachliche, kulturelle und soziale Barrieren, außer Betracht zu lassen.

Ein Ansatz kann über die Betrachtung eines größeren zeitlichen Abschnitts und dabei insbesondere der Anfänge der Softwareentwicklung in der Automobilindustrie gefunden werden. Die Ursprünge dieser Entwicklung sind vor mehr als zwei Jahrzehnten bei Bosch selbst zu suchen und haben somit im eigenen Hause eine entsprechend lange Historie. Dennoch stellen sie, gemessen an der Zeitspanne der Entwicklung des Ottomotors, einen sehr kurzen Zeitraum dar. Waren es zu Beginn nahezu ausschließlich Ein-Mann-Projekte, die Softwareentwicklung für Steuerung und Regelung der Verbrennungsvorgänge des Motors betrieben haben, befinden wir uns heute in der Situation, dass wesentliche Funktionen eines Fahrzeugs nicht selten durch eine Vielzahl von Entwicklern und in immer verstärkterem Maße im weltweiten Entwicklungsverbund entwickelt werden.

Betrachtet man eben diese Anfänge, stellte sich die Frage der Kommunikation so gut wie gar nicht; mit wem hätten diese Ein-Mann-Projekte im großen Maße kommunizieren sollen?

Hinzu kommt, dass Software mehr als Anhängsel zu einer Hardware verstanden wurde, und nicht als eigenständiges Produkt, welches einer entsprechenden Projektorganisation bedurft hätte.

Doch eine rasante Verbreitung und Anwendung der Technik – noch vor der eigentlichen Etablierung – haben Komplexität und Einsatz von Ingenieuren in hohem Maße erfordert. Die Entwicklung geeigneter Methoden hat dabei in keinster Weise Schritt halten können und führte unter anderem auch dazu, den stetig anwachsenden Informationsbedarf zu vernachlässigen.

Erst in jüngster Vergangenheit sind diese Defizite stärker in den Vordergrund und ins Bewusstsein gerückt und haben adäquate Lösungen erfordert.

Einen wichtigen Beitrag konnte die systematische Einführung des Projektmanagements und der Prozessentwicklung für die SW-Entwicklung gespiegelt am Reifegradmodell CMM (SEI der Carnegie Mellon University, Pittsburgh) leisten. Demnach darf sich Kommunikation nicht als nebensächliche Erscheinung manifestieren, sich als ein zufällig im Flur abspielender Prozess darstellen. Es darf ebenso nicht als Selbstläufer angesehen werden, welches nicht strukturiert und organisiert werden muss. Vielmehr muss Kommunikation als aktive Führungsaufgabe verstanden werden, die Menschen in einer definierten Projektorganisation zusammenbringt und mit Instrumentarien wie Projektmeeting, Konzeptdurchsprachen, Reviewsitzungen, Entwicklungsplanungsrunden u.v.m. versorgt, um nicht zuletzt dadurch die notwendigen Kommunikationskanäle für die Softwareentwicklung mit allen Bereichen der Entwicklungsorganisation zu schaffen.

Nicht zu unterschätzen ist dabei der Beitrag, den das Reifegradmodell durch die Schaffung einer „common language“ – einer einheitlichen Sprache und Begrifflichkeit – leistet, mittels der innerhalb der Softwareentwicklung, aber auch z.B. mit Management und Leitung eine definierte Kommunikation erfolgen kann.

Mit den beschriebenen Lösungsansätze ist man jedoch nicht am Ende, sondern eher am Anfang des Weges. Künftig müssen die Ansätze verstärkt in den Hochschulen Eingang finden und neben einer breiten Diskussion auch das Bewusstsein für die Problemstellung schärfen.

Weiterhin sind Ansätze zu finden, die die mit der steigenden Komplexität der Software-Systeme einhergehende Informationsflut kanalisiert und somit gezielte und nicht zufällige Informationsvernetzung schafft – damit *Kommunikation* auch künftig möglich ist und nicht ineffizient oder als Belastung empfunden wird.

*Taskin Ege*

*GS-ED/ESI-E*

*Robert Bosch GmbH, Stuttgart*



Berliner Str. 62 • 14467 Potsdam

Phone: +49 (0)331 8170688

Fax: +49 (0)331 8170698

http: [www.intervista.ag](http://www.intervista.ag)

Email: [info@intervista.at](mailto:info@intervista.at)

## Positions-Papier

Zur Initiative

### „Softwaresysteme als

### Kommunikationsthema“

#### International Branches

INTERVISTA GmbH (Austria)

Vogelsangweg 2

A-3270 Scheibbs

INTERVISTA LTD (Georgia)

105a, Tsereteli Avi

GE-380019 Tbilisi

INTERVISTA ist ein Integrations- und Reengineering- Expert. Es werden unternehmensweite und unternehmensspezifische Softwarelösungen geplant, erstellt und gewartet. Diese auf zugekauften oder eigenentwickelten Standardkooomponenten basierenden Lösungen werden Corporate-Software genannt. Denn diese speziell auf die beauftragenden Unternehmen zugeschnitte Gesamtsysteme schaffen den erwünschten IT-Vorsprung zu Standardlösungen und sichern so den Wettbewerbsvorsprung der Kunden. Den Lösungen gemeinsam ist:

- Ausschöpfung hoher Rationalisierungspotentiale
- Unternehmensweite Lösungen auf Basis von Client/Server- oder Internetsystemen
- Große Datenbankanwendung

INTERVISTA betreut den gesamten Lifecycle der Software, es müssen also Designer, Programmierer und Wartungspersonal gleichermaßen einem transparenten System gegenüberstehen. Die Transparenz zu vermitteln, ist zur Zeit unser größtes Problem, denn mit Architekturbeschreibung und Klassendiagramm sind meist die wirklich „verdeutlichenden“ Möglichkeiten erschöpft. „Ein Bild sagt mehr als tausend Worte“, wo nur Text vermittelt wird, ist die Kommunikation mühsam. In der Zukunft muß und wird im Softwarebereich wesentlich mehr mit Graphic gearbeitet werden. Es ist völlig unmöglich, mit Texten in einem Design-Team zu arbeiten. Das Verständnis aller, in nachfolgenden Phasen tätigen Personen wird ebenso erschwert.

Bei Systemdefekten wird am schmerzlichsten klar, daß das Thema Komplexität mit ansteigender Größe zum Show-Stopper wird. Es hat den Anschein, daß die Software-Systeme heute zwar größer sind, aber die Probleme das selbe Ausmaß behalten.

Es müssen immer größere Strukturierungseinheiten gefunden werden, um dem Problem der „großen Zahl“ Herr zu werden. Eine Übersicht über eine Vielzahl von Modulen, Komponenten, Objekten und Subsystemen kann nur mit Hilfe eines graphischen Systems gewonnen werden. Damit wäre ein integratives Mittel zum Verständnis aller beteiligten Personen gegeben.

HR B 16071 P • Amtsgericht Potsdam

Vorstand: Matthias Stauch, Prokura: Dr. Andreas Föhrenbach

Aufsichtsrat: Reinhard Irsigler (Vorsitzender), Klaus-Otto Schmidt, Wolfgang Kapst

Bankverbindung: Dresdner Bank • Konto Nr. 900 69 04 00 • BLZ 100 800 00

## Kommunikation mit Software im Kontext IBM – SAP Umfeld

SAP unterstützt mit seinen Produkten u.a. die gesamte IBM eServer Palette mit den zugehörigen Betriebssystemen inklusive Linux auf Intel Servern und zSeries. Desweiteren wird DB2 als strategisches Datenbanksystem mit den SAP Lösungen angeboten. Die Zusammenarbeit erstreckt sich aber auch auf den Bereich der Kundenprojektberatung und Implementierung.

Im Softwarebereich werden hauptsächlich IBM Produkte als Zusatz- bzw. Integrationskomponenten positioniert und in vielen Fällen auch von SAP zertifiziert. Beispiele sind CommonStore, MQ Series Link für R/3 und mehrere Tivoli Produkte. Ankündigungen aus der jüngsten Vergangenheit beinhalten auch eine Zusammenarbeit im Bereich IBM Websphere und SAP NetWeaver.

Die heute bei vielen Kunden eingesetzten Anwendungen umfassen eine Vielzahl von sogenannten „Standard“-Anwendungen, wie ERP, CRM und SCM verschiedener Hersteller, eine Menge von „Spezial“-Anwendungen, sowie eine Vielzahl von „alten“ und selbst entwickelten Anwendungen. Daraus lassen sich die folgenden charakteristischen Merkmale feststellen:

- Weitestgehende Homogenität innerhalb der einzelnen Produkte/Anwendungen eines Anbieters, aber Heterogenität der Anwendungen, Middle Ware und Infrastruktur in der Gesamtarchitektur
- Viele Anwendungsinseln mit eigenen Schnittstellen
- Steigende Anzahl der Systeme und Anwendungen

Diese Ausgangsposition sowie die Anforderungen der heutigen Zeit, wie Firmenübergreifende Anwendungen, Flexibilität, Nutzung neuester Technologien und Integration der Geschäftsprozesse und Systeme ergeben eine Menge von aktuellen Problemen:

- Die Komplexität der Anwendungen steigt
- Die Komplexität der gesamten Systemlandschaft ist kaum noch beherrschbar
- Jedes einzelne System, HW, SW und Anwendung hat einen immer höheren Automatisierungsgrad
- Die „echten“ Probleme tauchen erst beim Kunden auf, jeder Hersteller konzentriert sich auf seine Produkte
- Die Middleware bietet immer mehr Funktionalität und hilft dem Anwendungsprogrammierer sich auf die Anwendungslogik zu konzentrieren
- Die Anforderungen bezüglich Skills für Programmierer, System-Integratoren, Operator und andere Personen sind sehr hoch
- Es gibt kaum Architekten, die eine Systemlandschaft noch überblicken bzw. verstehen können
- Immer engere Verzahnung zwischen Anwendung und Middleware-„Funktionalität“

Aus den oben genannten Problemen lassen sich eine Reihe von Konsequenzen ableiten. Um erfolgreich komplexe Systemlandschaften zu betreiben, um neue Anwendungen zu entwickeln und um existierende Anwendungen und Prozesse zu integrieren, müssen veränderte Personalprofile geschaffen werden. Neben den klassischen Rollen der Architekten und Spezialisten werden Moderatoren bzw. Translatoren benötigt, die die Verbindung und Integration der einzelnen Komponenten herstellen können. Die Komplexität der verschiedenen Systeme bedingt einen immer höheren Spezialisierungsgrad, die Integration einer immer grösser werdenden Systemlandschaft ein immer breiter werdendes Gesamtwissen. Die Vielzahl der Systemkomponenten HW, SW, Middleware, Anwendung und Integration wird durch immer komplexere Software unterstützt. Diese Softwareprodukte bieten immer bessere Werkzeuge und Möglichkeiten zur Anwendungsentwicklung und -betreuung. Damit solche Produkte effizient eingesetzt werden können, müssen im Vorfeld immer umfangreichere Schulungsmaßnahmen vorgenommen werden. Die neuen Produkte ermöglichen einen erheblichen Produktivitätszuwachs, haben aber eine relativ hohe Einstiegschürde im Vergleich zur Vergangenheit.

Heterogenität auf mehreren Ebenen, Komplexität der verschiedenen Komponenten und Schnittstellen mit den gesteigerten Anforderungen an Integration können nur dann bewältigt werden, wenn Produkte eingesetzt werden, die die heute wichtigsten Standards unterstützen. Bei Eigenentwicklungen müssen solche Standards genauso berücksichtigt werden. Integration von neuen und alten Anwendungen kann nur auf Basis von Standards erfolgreich sein. IBM propagiert bereits seit längerer Zeit die Vision offener, Standard-basierende Systeme und ist in vielen Gremien aktives Mitglied. J2EE, Eclipse und Web Services sind dafür Beispiele aus jüngster Zeit. SAP hat in den letzten Jahren auch besonders Standard-basierende Lösungen propagiert. Eine weitere Trend ist die Verlagerung „allgemeiner Dienste und Funktionen“ in die Middleware z.B. Kommunikation, Interoperability.

Die Standards erleichtern die „Kommunikation“ der SW miteinander. Eine ebenso große Herausforderung ist die Kommunikation zwischen den Anwendungs- und Systemarchitekten, den Implementierern, den Betreibern und den Geschäftsprozeß-Verantwortlichen. Hier werden je nach Aufgaben- /Anwendungsgebieten die gleiche Terminologien für unterschiedliche Sachverhalte benutzt. Daher benötigen wir in Projekten und Betrieb „Moderatoren“, die in enger Zusammenarbeit mit Spezialisten und Architekten die Fäden zusammenführen, „übersetzen“ und beim Design und der Umsetzung die Koordination übernehmen.

***Rainer Hofmann***

*Director IBM SAP Alliance  
IBM Deutschland GmbH*

The logo for Aviso GmbH features the company name in a bold, lowercase sans-serif font. To the right of the text is a stylized green graphic consisting of several concentric, overlapping loops that resemble a spiral or a stylized 'A'.

## Position Statement

### Das Unternehmen in der Softwareentwicklung

Die aviso GmbH als Systemhaus ist im Bereich der Softwareentwicklung vorrangig in der Anpassung von Schnittstellen und in der Entwicklung webbasierter Systeme und Security-Applikationen tätig. Dabei bedienen wir uns der Entwicklungsframeworks ECLIPSE und der Engineering-Tools von Rational und Visio zur Projektierung und Planung.

Komplexität haben unsere Projekte dadurch, dass sie in den Unternehmen unserer Kunden unternehmens- und abteilungsübergreifend Prozesse abzubilden und sicherheitssensitiv über öffentliche und private Netze zur Verfügung gestellt werden.

### Zum Problem Komplexität und Kommunikation

In unserem Arbeitsumfeld zeigt sich die steigende Komplexität in der Entwicklung bzw. im Betrieb von software-intensiven Systemen durch die erhöhten Anforderungen aus den unternehmensübergreifenden Prozessabbildungen und aus dem Bereich der Integration des Security-Frameworks bei Einbindung von Multiplikatoren, Lieferanten, Kunden und Partnern in der Supply-Chain. Dabei sind die einzelnen Nutzer mit den Lösungen innerhalb ihrer Prozesskette so abzuholen, dass keine umfassenden Einschnitte in den Geschäftsbetrieb notwendig werden. Gleichzeitig verlagert sich die Integration neuer Applikationen server- oder webbasiert auf produktions- und kommunikationsschwache Arbeitszeitabschnitte.

Für uns ist es notwendig, jedes Herangehen auf die zukünftige Entwicklung unserer Kunden hinsichtlich der Skalierbarkeit, Wirtschaftlichkeit, Redundanz und Sicherheit abzustellen. Dazu bedarf es umfassender Projektierungs-, Simulations- und Testmethoden, deren Entwicklung wir im Rahmen der Schulung verfolgen und an deren Erprobung wir nach unseren Möglichkeiten als gewerbliches Unternehmen teilnehmen.

Insofern reichen uns die bisherigen Vorgehensweisen für die Zukunft nicht vollständig aus und regen uns an, Partner für innovative Entwicklungsansätze zu suchen.

In unserem Unternehmen und Netzwerk verbundener Unternehmen tauschen Projektmanager, systemnahe Integratoren, Kaufleute und Softwareentwickler ständig Wissen über Softwaresysteme aus. Dieser Austausch wird auch über Fachverbände und mit ausländischen Partnern geführt.

Erschwerend wirkt derzeit der Stand des Wissens der einzelnen beteiligten Personen vor allem in verbundenen Projekten und in Projekten mit externen Ressourcen. Prozessorientierte Softwareentwicklung auf der Basis ingenieurtechnischen Herangehens und komplexer Projektierung wird noch nicht oft praktiziert. Somit sind in erster Linie Kommunikationsschwierigkeiten durch die Gewinnung der Ressourcen zu überwinden, die Projekte entsprechend verteuern.

Meiner Meinung nach sehe ich vor allem eine starke Abhängigkeit der Kommunikationsprobleme von den Randbedingungen in den jeweiligen Projekten und in der Betroffenheit des Aspektes eines Softwaresystems, in welchem das notwendige Wissen benötigt wird.

Kommunikationsprobleme sind immer vorrangige Verhinderungspotentiale. Vor allem in der

Projektplanung und Kostenfindung sowie der Budgetverteidigung und in der rechtzeitigen Systemerweiterung sind diese besonders hinderlich.

Der Zusammenhang zwischen unbefriedigender Qualität und Produktivität auf der einen Seite und Mängeln in der Kommunikation auf der anderen Seite besteht, da durch Qualitätssicherungsverfahren auch eine höhere Transparenz zu verzeichnen ist. Diese fordert unmissverständlich Änderungen (Change Management) in der Methoden und den Prozessen beim Projektteam und beim Kunden, die liebgeordnete Bequemlichkeiten angreifen.

### **Welchen Handlungsbedarf sehe ich**

Die Ausbildung von Fachkräften ist auf ein umfangreiches frühzeitig beginnendes Auswahlverfahren aufzubauen und mit Assessmentmethoden während des Ausbildungsverlaufes zu untersetzen. Individualisierte Ausbildungsverfahren in Kleingruppen und Teams sollten die soziale Kompetenz der Auszubildenden entwickeln helfen.

Eine frühzeitige Projektbeteiligung sollte die Kommunikationsfähigkeit und das kreative Querdenken fördern.

In der Forschung sind Wege zu finden, schnell und ohne große bürokratische Hemmnisse Unternehmens- und Bildungsstätten-übergreifend Entwicklungsteams zu bilden und die Ressourcen im Rahmen von Kompetenznetzwerken zu verbinden. Für die vor allem klein- und mittelständischen Unternehmen sind die Existenzrahmenbedingungen politisch zu verbessern. Ein besserer Zugang zu den vorhandenen Innovationsförderprogrammen ist zu ermöglichen.

Die Organisationsstruktur innerhalb der Unternehmen hat direkte Auswirkungen auf die Einschränkung von Kommunikationsproblemen und die rechtzeitige Projektvorbereitung und Durchführung.

Die Informatikausbildung an den Universitäten und Hochschulen vermag ich nicht einzuschätzen, die Ergebnisse in Form ausreichend vorbereiteter Absolventen für den unmittelbaren Einsatz in komplexen Softwareentwicklungsprojekten zeigt uns in unserem Erfahrungsraum zu wenige qualifizierte und ergebnisorientiert denkende Bewerber.

Das lässt auf eine dringende Orientierung der Ausbildung auf praxisnahe, ingenieurtechnische Verfahren und einen zeitnahen Schulterchluss mit den gewerblichen Unternehmen schließen. Ergebnis- und Marktorientierung, kaufmännisches Grundwissen, Fähigkeiten und Fertigkeiten und eine entsprechende Übersicht sind bisher bei den von uns konsultierten Bewerbern nicht ausreichend vorhanden. Hier sehen wir großen Handlungsbedarf für die Ausbildung.

Die durch die IHK ausgeschriebene Fachausbildung als Fachinformatiker und IT-Systemkaufmann führt durch eine fehlende Qualitätskontrolle nicht zu den gewünschten Ergebnissen. Hier hat vor allem die durch die Arbeitsverwaltungen geförderte Fortbildung negative Marktauswirkungen hinsichtlich der angebotenen Qualität der Softwareentwicklung und der Verhinderung des Aufbaus komplexer Strukturen. Besonders wirkt sich bei diesen Marktanbietern fehlendes Wissen und gering entwickelte soziale Kompetenz aus.

**Jürgen Weinreich**

*IT-Consultant*

*aviso GmbH Werder*

## Softwaresysteme als Kommunikationsthema

Die IZB SOFT ist eines der führenden IT-Unternehmen der Sparkassen-Finanzgruppe und steht für höchste Qualität und Effizienz in der Informationstechnologie. Im Dienste unserer Kunden – der bayerischen Sparkassen – entwickeln, implementieren und betreiben wir branchenspezifische IT-Anwendungen und -Systeme. Im Kern handelt es sich um ein komplexes und umfassendes Anwendungssystem, das alle Anforderungen der bayerischen Sparkassen zur Bewältigung des Kundengeschäftes abbildet. Dabei können wir auf Kompetenzen zurückgreifen, die wir in unserer 30-jährigen Tätigkeit im Bereich Datenverarbeitung für das Bank- und Kreditwesen erworben haben. Dieses Wissen ist das Kapital in Softwareunternehmen.

Die kontinuierliche Weiterentwicklung und Pflege unseres umfassenden Systems erfordert effiziente Strukturen und Prozesse, für die bestmögliche Unterstützung unserer Kunden. Aufgrund der starken Abhängigkeiten bei der Softwareentwicklung in einem integrierten System ist mit steigender Komplexität und wachsendem Funktionsumfang auch ein überproportional steigender Kommunikationsaufwand zwischen den einzelnen Entwicklungsteams festzustellen. Dieser Umstand tritt besonders dann in den Vordergrund, wenn ein bestehendes System grundlegend umgebaut oder erneuert wird. Deshalb wird nicht nur die technische Systemlandschaft ständig den allgemein hohen Anforderungen eines Bankensystems angepasst, auch die Prozesse werden laufend auf den Prüfstand gestellt und optimiert, falls dies erforderlich ist.

In einem Softwareentwicklungsprojekt arbeiten die unterschiedlichsten Spezialisten und Wissensträger zusammen. Die zusätzliche Integration unterschiedlicher Führungsebenen ist obligatorisch. Die Herausforderung ist dabei, die unterschiedlichen Wissensträger so einzubinden, dass jeder Beteiligte in einem Vorhaben das gleiche Verständnis hat und seinen optimalen Beitrag zum Gesamterfolg leistet.

Eine Strategie, die Konsolidierung der Datenbestände und Wiederverwendbarkeit von Modulen voraussetzt, stellt hohe Anforderungen an die Abstimmung zwischen den einzelnen Projekten. Oftmals wird dies für den laufenden Betrieb unterschätzt. Jede Änderung im System kann bei Unkenntnis von Abhängigkeiten zu fatalen Störungen führen. Mangelhafte Kommunikation führt somit relativ schnell zu sinkender Qualität in der angebotenen Leistung.

Daher heißt die Herausforderung für die Entwicklung und den Betrieb eines komplexen und umfassenden Gesamtsystems: die Kommunikation auf allen Ebenen sicherzustellen. Hierzu gibt es zahlreiche technische Hilfsmittel, die jedoch noch keine ideale Lösung bieten. Auch können sie immer nur so gut sein, wie der Input in diese Tools. Weiter kommt erschwerend hinzu, dass die Menschen in den Arbeitsprozessen inzwischen mit einer wahren Informationsflut konfrontiert werden. Dies geschieht in bester Absicht. Informationen sollen nicht zurückgehalten werden und die Kommunikationsmedien gestalten es inzwischen sehr einfach, Wissen weiterzugeben. Die Herausforderung dabei ist, das Wissen am richtigen Ort bereitzustellen. Häufig sind das Problem nicht fehlende Informationen, sondern die Identifikation der richtigen Daten.

Es reicht nicht, wenn Wissen nur in den Köpfen einzelner vorhanden ist. Um es optimal zur Verfügung zu stellen, ist es wichtig, dass Informationen klar strukturiert aufbereitet sind. Neben der textlichen Dokumentation ist die grafische Visualisierung besonders wichtig. Der Mensch kann Bilder ungleich schneller und länger speichern, als Zahlen oder Text.

Dieser Umstand sollte in der Ausbildung von Informatikern stärker berücksichtigt werden. Ein exzellenter Entwickler muss neben der Codierung auch die entsprechende Dokumentation beherrschen, um sein Werk anderen zur Verfügung zu stellen. Softwareentwicklung ist mehr denn je zum Teamwork geworden.

*Michael Weigoldt*

*Geschäftsführer der IZB SOFT (Sparkassen- und Giroverband)*

## **Position Statement der Deutsche Telekom**

### **Unternehmensprofil bezüglich Software:**

Die Deutsche Telekom, einer der weltweit größten Diensteanbieter der Telekommunikationsbranche, gliedert sich in vier Divisionen, von denen die größte, T-Com, für die Leistungen im Zusammenhang mit dem Festnetz verantwortlich zeichnet.

Innerhalb der T-Com verantwortet unser Betrieb "IP" (Informationsmanagement und Prozesse) die Gestaltung der Prozess- und IV-Landschaft und hat dabei auch die Aufgabe, alle IV-Lösungen der T-Com bereitzustellen und zu betreiben.

### **Das Problem Komplexität und Kommunikation:**

Die wachsende Zahl an IV-Anwendungen, die miteinander vernetzt werden, führt zu einer steigenden Komplexität und stellt damit immer größere Herausforderungen, z. B. an ein Multiprojektmanagement.

Alle Tätigkeiten zur (Weiter-) Entwicklung einzelner IV-Systeme und der gesamten IV-Landschaft werden entlang eines verbindlich vorgeschriebenen Unternehmensprozesses erledigt. Wir unterstützen diesen Prozess durch IV-Anwendungen, die diesen Prozess selbst und die Kommunikation aller am Prozess beteiligten Kräfte unterstützen und verbessern, so dass z. B. ein reibungsloser Informationsfluss, eine enge Abstimmung zwischen allen Beteiligten oder ein Monitoring aller laufenden Maßnahmen gewährleistet werden.

Entlang des bereits erwähnten Prozesses gibt es eine Vielzahl von Personen, die in verschiedenen Rollen mit den unterschiedlichsten Aufgaben betraut sind und Informationen über die im Betrieb befindlichen IV-Systeme bzw. die geplanten oder bereits eingeleiteten Maßnahmen zu deren Weiterentwicklung benötigen. Dies sind im Zusammenhang mit dem Managen der Anforderungen z. B. die Key-Accounter, die unsere Ansprechpartner für unsere internen Auftraggeber darstellen und die fachlichen Anforderungen aufnehmen oder die Business Engineers, die diese Anforderungen für die Projekte aufbereiten. Im Bereich der Realisierung u. a. Projektleiter und -mitarbeiter, aus übergeordneter Sicht z. B. Programmmanager oder Architekten (zu deren Aufgaben die Gestaltung der Soll-IV-Landschaft und die Migrationsplanung gehört).

Die Kommunikation innerhalb dieses Personenkreises wird dadurch zu einer besonderen Herausforderung, dass es sich um einen sehr großen und heterogenen Personenkreis an einer Vielzahl von Standorten handelt und neben den technischen Wechselwirkungen zwischen den IV-Systemen auch deren Inhalte vielfältige Wechselwirkungen mit den Inhalten anderer IV-Systeme haben.

Die Kommunikationsprobleme würden erschwert, wenn das Wissen nur mündlich weitergegeben wird und Vereinbarungen nur mündlich erfolgen würden. Wir legen deshalb großen Wert auf eine schriftliche Dokumentation mit klaren Strukturen, z. B. bei der Darstellung des Inhaltes oder der Art und Weise der Ablage.

Die größten Kommunikationsprobleme treten in der Regel in zwei Bereichen auf: dort, wo der Übergang aus der eher fachlichen Sicht (z. B. Vertrieb) in die iv-technische Sicht stattfindet und dort, wo unterschiedliche „Softwarewelten“ mit unterschiedlichen Sprachen, Begriffen, Datenmodellen usw. miteinander in Berührung kommen.

Kommunikationsprobleme führen zu Problemen bei der Realisierung der IV-Systeme und können dann zu Budget- und Terminproblemen führen, weil z. B. in der Planungsphase übersehen wurde, dass es eine Schnittstelle zu einem anderen IV-System gibt und dies erst im Laufe der Entwicklung erkannt wurde, weil Anforderungen nicht eindeutig beschrieben wurden usw.

## **Handlungsbedarfe**

Da es immer wichtiger wird, Gesamtzusammenhänge zu verstehen und zu gestalten, sollten sich die Hochschulabsolventen aus unserer Sicht - neben den eher technischen Aspekten eines IV-Projektes oder den Methoden zur Leitung eines Projektes verstärkt mit dem Thema Multiprojektmanagement (i. w. die einzelprojektübergreifende Koordination aller selektierten Projekte eines definierten Bezugsbereiches hinsichtlich ihrer Terminpläne, Ressourcen, Schnittstellen und inhaltlichen Veränderungen) beschäftigen, den Nutzen des Projektes für das Unternehmen auch aus betriebswirtschaftlicher Sicht verstehen und darstellen können (z. B. im Rahmen eines Business Case, im Gespräch mit dem Controlling usw.) und eine Architekturplanung (Zusammenhänge zwischen mehreren IV-Systemen) erstellen können.

***Jürgen Brunn***

*T-Com Zentrale, Deutsche Telekom*

## Position Statement der Deutschen Post ITSolutions

Die Deutsche Post ITSolutions ist zuständig für das Design, die Erstellung und die Implementierung von IT-Lösungen für die Deutsche Post AG. Die Anwendungen decken das volle betriebswirtschaftliche Spektrum des Unternehmens ab: Von klassischen BWL-Applikationen für das Finanz- und Rechnungswesen über HR-Applikationen, Track und Trace Anwendungen bis zur PPS-Software für Brief-Verteilzentren.

Historisch bedingt und durch Unternehmensakquisitionen/zukauf wird auf unterschiedlichen Entwicklungsplattformen gearbeitet. Dadurch ergibt sich eine heterogene Entwicklungslandschaft. Integration und Standardisierung von Business Applications unter dem Aspekt der Kostensenkung ist vordringliches Ziel.

Zukünftig ist der Einsatz von Standardsoftware geplant, bzw. Standards für XML- und Web-Services. Voraussetzung ist eine weitgehend einheitliche Infrastruktur.

E-business-Applications, Altsysteme, Paketlösungen (ERP) müssen sich annähern, da Application Server vermehrt Kernservices liefern. Gleichzeitig ist eine Technologieerweiterung notwendig:

- Anwendungsarchitektur (auf Basis von Standards inkl. Web-Services), die die Abbildung der Geschäftslogik unterstützt
- Konfigurations- und Customizing-Support
- Zulassen von interner und externer Integration durch Adapter/Konnektoren (EAI-Technologie).

Auch die Prozesse müssen Standards unterliegen (Beispiel Requirement-Engineering)

Somit müssen sich die Spezialisten unterschiedlicher Technologien untereinander sowie mit SW-Architekten, System-Architekten und Projektmanagern abstimmen können.

„Kommunikationsforen“ wie VTR, Collaboration-Suites, KM-Tools werden immer notwendiger, also wichtiger.

Kern ist und bleibt das Abbilden von integrierten Geschäftsprozessen in Technologien und Architekturen, die zukünftig zeitnah und flexibel Anforderungsänderungen berücksichtigen können.

**Andreas Kersten**

*Geschäftsführung  
Deutsche Post ITSolutions*

## **Kommunikation als Notwendigkeit im Betrieb von komplexen Software-Systemen**

Wie stellt man Betrieb und Weiterentwicklung eines Verbundes von Software-Systemen sicher? Überwachung von Hardware, verfügbaren Ressourcen und einzelnen Software-Prozessen ist heutzutage durch etablierte Methoden und Werkzeuge möglich. Problematisch wird es jedoch dann, wenn nicht das Verhalten eines einzelnen Systems, sondern das Zusammenspiel mehrerer Systeme (im folgenden System-Verbund genannt) eine entscheidende Rolle spielt.

Zur Modellierung einzelner Systeme existieren vielfältige Notationen und Methoden wie z.B. UML. Diese lassen sich jedoch nicht ohne weiteres auf System-Verbünde anwenden.

Um Betrieb und Integration von verteilten Systemen in einem Verbund zu gestalten, bedarf es einer Möglichkeit über den System-Verbund und dessen Verhalten in einer einfachen und strukturierten Weise zu kommunizieren. Für die effektive Kommunikation fehlt bisher jedoch eine standardisierte Vorgehensweise und Beschreibungs-Sprache zur Abbildung und Modellierung des Verhaltens und der Abhängigkeiten.

Beispiele aus dem Umfeld der Lufthansa sollen dies verdeutlichen:

Gerade in der Luftfahrtindustrie zeigen sich die Auswirkungen des genannten Problems in einem hohen Maße. Alleine im System-Verbund der Lufthansa interagieren weit über 100 verschiedene Software-Systeme, welche in verschiedenen Betriebssystem-Umgebungen, verschiedenen Programmiersprachen und verschiedenen Jahrzehnten in unterschiedlichen Architektur-Ansätzen erstellt wurden. Im Zeitalter der Globalisierung werden allerdings die Verbindungen zwischen Software-Systemen weltweit immer wichtiger und gleichzeitig auch unüberschaubarer.

Die Änderung von einzelnen Systemen des Verbundes im laufenden Betrieb darf keinesfalls zu einer Störung des Gesamt-Systems und damit des operativen Ablaufs der Flugdurchführung der beteiligten Airlines führen.

Die IT-Industrie steht damit an Anfang dieses Jahrtausends vor einer großen Herausforderung. Der Betrieb von komplexen System-Verbänden erfordert eine effiziente Kommunikation über die Abhängigkeiten und Auswirkungen von Änderungen innerhalb und außerhalb des einzelnen Systems. Wie kann dieser Bedarf erfüllt werden um in Zukunft nicht durch die weiterhin steigende Komplexität begrenzt zu werden?

***Thomas Kullmann***

*Lufthansa Systems Airline Services GmbH*

## Softwaresysteme als Kommunikationsthema



### SPM Unternehmensprofil bzgl. Software

SPM Technologies ist ein Professional Services Anbieter im IT Bereich. Der Schwerpunkt der durch SPM durchgeführten Projekte liegt in der Konzeption und Umsetzung von Integrationsprojekten in komplexen IT Landschaften bei Großkunden. Hierbei decken SPM Experten den gesamten Lebenszyklus eines Systems von Anforderungserhebung bis Betrieb ab.

### Komplexität und Kommunikation

#### Kundenkommunikation

Bei einem typischen Projekt gibt es viele Kommunikationsschnittstellen zwischen Entscheidern und Technikern auf Kundenseiten mit den Experten von SPM. Oftmals geht es zunächst darum, Strukturen in Organisation und IT-Systemen beim Kunden zu verstehen. Gerade im Großkundenbereich zeigt sich immer wieder, dass die Erfassung vor allem der Zusammenhänge der einzelnen Systeme aufgrund von internen Kommunikationsbarrieren schwierig ist. Ursache dafür ist häufig die eher strukturelle Verantwortungs-Separation beim Kunden, die den Blick auf die globalen Systemzusammenhänge verschleiert. IT Bereiche wirken wie autonome „Fürstentümer“, die nur ein begrenztes Interesse an der Erarbeitung eines Gesamtkonzeptes haben. Schnittstellen zwischen Systemen werden auf einer eher technischen Art betrachtet. Die Konzentration auf den technischen Kanal des Datentransports verhindert jedoch ein Verständnis darüber, was die Daten fachlich repräsentieren. Dieser Umstand stellt den Dienstleister vor die komplexe Aufgabe, vor allem die fachlich integrativen Zusammenhänge eines Gesamtsystems gewissermaßen bottom-up kommunikativ erschließen zu müssen. Eine solche Sicht über komplexe Gesamtsysteme und der Abhängigkeiten der einzelnen Komponenten ist bei Kunden oftmals nicht vorhanden, da dies durch eine starre Aufteilung in begrenzte Verantwortlichkeiten (Tunnelblick) erschwert wird.

Die oben geschilderten Kommunikationsprobleme wirken sich vor allem dann negativ aus, wenn es um Änderungen in einem bestehenden System geht. Da die fachliche Bedeutung bestimmter Schnittstellen und ihrer Datenstrukturen ungeklärt ist, ist eine Änderung immer mit einem sehr hohen Risiko für das Gesamtsystem verbunden. Abhängigkeiten sind oft unentdeckt, undokumentiert, implizit oder mittlerweile vergessen.

Eine große Hilfe bei der Erschließung von Systemen sind alle Arten von Schnittstellenbeschreibungen zwischen Komponenten. Auch eine grobe Dokumentation der im Projektverlauf gefällten Entwurfsentscheidungen erleichtert das Verständnis über historisch gewachsene Systeme. Diese müssen erfahrungsgemäß aufgrund technischer und organisatorischer Rahmenbedingungen Kompromisse in Bezug auf die Eleganz der technischen Lösung eingehen, die im nachhinein immer schwer nachzuvollziehen sind.

#### Projektkommunikation

SPM hat die Erfahrung gemacht, dass die Kommunikation innerhalb eines Projektteams dann am besten ist, wenn die einzelnen technischen und fachlichen Aspekte eines Systems von Experten des jeweiligen Gebiets bearbeitet werden. Diese müssen über einen definierten Prozess hinweg Artefakte miteinander austauschen, die die Fachsprache des Empfängers sprechen. Durch solche prozessuale Reglementierungen steigt der Wert der beschreibenden Artefakte, da diese einen klareren Fokus bekommen. Zudem kann die Informationsaufbereitung und -weitergabe durch die Verwendung von Artefakt-Schablonen beschleunigt werden.

Über ein abstraktes Gebilde wie ein IT-System, kann nur mittels eines „Bauplans“ kommuniziert werden, der in schriftlicher oder elektronischer Form vorliegt. Die Sprachmittel, die diesem Plan

zu Grunde liegen, sollten jedoch nicht zu stark reglementiert sein. So reichen oft sehr einfache Mittel aus (z.B. benannte Blöcke mit attributierten Verbindungen), um eine ergebnisorientierte Kommunikation zu unterstützen. Formale Regelungen dieser Mittel wirken dabei eher hinderlich. Ohne jegliche bildliche Darstellung eines Systems ist die Erlangung einer gemeinsamen Vorstellung der einzelnen Projektmitglieder nahezu unmöglich. Textuelle Beschreibung lassen zu viel Interpretationsspielraum und sollten daher eher unterstützender Natur sein. Es sollte in einem Projekt grundsätzlich ein Masterplan existieren, der immer wieder zur Unterstützung der Kommunikation zu Rate gezogen werden kann und als „Big Picture“ dient.

Auf Basis dieses Bauplans sollte eine Separation des Systems in Module erfolgen, deren Abhängigkeiten klar definiert sind. Dadurch wird es möglich, eine Minimierung an Kommunikationsschnittstellen der einzelnen Modulverantwortlichen zu erreichen.

### **Handlungsbedarf der Forschung**

In der Forschung wird oft der Fokus auf die vollständige Reglementierung der Prozess- und Sprachmittel zur Beschreibung von Systemen gelegt. Vernachlässigt wird dabei hingegen die Vermittlung von Fähigkeiten und Methoden zur Unterstützung des kreativen Akts des Software- und Systementwurfs.

*Dipl.-Inform. Guido Laures*

*spm technologies*

## Software–Systembeschreibungssprachen vor FMC

Oft halten Techniker, mit denen ich mich unterhalte, ihre Kommunikation über Software für effektiv. Sie glauben, ihre Unterteilung in Module und Klassen sei in einer Weise in sich selbst konsistent, die Missverständnisse ausschließe und die von ihnen postulierte Eindeutigkeit herbeiführe. Wenn ich mir dann das Ergebnis anschau, finde ich den unwiderlegbaren Beleg, dass eben all dies Behauptungen sind, die in die Irre führen.

Schauen zwei Techniker z.B. auf ein UML-Diagramm, so sehen sie eben nicht die Architektur des Systems. Sie sehen nicht das System. Sie sehen seine Bestandteile. In Wirklichkeit verhalten sie sich wie Auguren, die auf einen Haufen Knochen blicken und glauben daraus die Zukunft voraussagen zu können. Im Zweifel aber weiß keiner von Ihnen so genau, ob die Knochen nun von einem Hühnchen oder einem Hähnchen stammen.

„So what“ würden die Auguren fragen, welche Bedeutung hat das Geschlecht eines toten Tieres für die Zukunft, die daraus geweissagt werden soll. Ähnlich geht es unseren beiden Software-Ingenieuren vor ihrem UML-Diagramm. Und erst, wenn sie aus den Knochen einen Hahn zusammengesetzt haben, der den ganzen Tag kräht, dem Kunden aber nicht die gewünschten Eier legt, erkennen sie, welcher fataler Irrtum ihnen unterlaufen ist.

Auch die Weiterentwicklung zu UML 2.0 und die MDA (Model Driven Architecture) ändern an dem beklagten Umstand wenig. Wir haben mit beidem wiederum ausschließlich eine Sprache für Techniker, um Systemkomponenten zu beschreiben, die aber gleichzeitig unfähig ist, Systeme sichtbar zu machen.

Dass sich das Problem selbstverständlich verschärft, wenn wir versuchen, diese Sprache in der Kommunikation mit nichttechnischen Projektbeteiligten, also z.B. Kunden oder Anwendern zu verwenden, bedarf keiner besonderen Erwähnung. Dennoch zeigt die von der Standish-Group als Hauptgrund für das Scheitern von Projekten herausgearbeitete Ursache „mangelnde Benutzerakzeptanz“ auf erschreckende Weise, mit welcher dilettantischer Ignoranz eine ganze Branche die Kommunikation über eine ihrer wichtigsten Produktkategorien (Software) anfasst.

In unserem Beratungsgeschäft werden wir täglich beauftragt, uns Projekte, die gefährdet oder vom Scheitern bedroht sind, anzusehen und in der Art eines Feuerwehreinsatzes den lichterloh lodernen Brand zu löschen. Oft genug finden wir dabei, dass nicht technische Ursachen, nicht einmal Managementursachen zu den existierenden Problemen führen. Nein. Als Hauptursache finden wir immer wieder eine mangelnde Kommunikation zwischen allen Projektbeteiligten, besonders aber auch zwischen Technikern einerseits und Nichttechnikern andererseits über die Sache, das Ding (kurz das System), das erstellt werden soll, das erwartet wird und das als einziges den Nutzen bietet, der den erbrachten Aufwand rechtfertigt (ROI scheint ein Begriff zu sein, vor dessen nicht existenter Implementierung eine ganze Branche zu kapitulieren bereit ist).

Wirklich ärgerlich wird dieser Umstand nicht erst, wenn er einem branchenfremden erläutert werden muss. Bereits in der Kommunikation mit Budgetverantwortlichen oder Controlling-Beauftragten schlägt man nur resignierend die Hände über dem Kopf zusammen, holt entweder zu einem Tage dauernden Rundumschlag über die verheerenden Probleme der Softwarebranche aus oder wünscht sich einen Controller her, der diese Probleme bereits kennt und (Achtung: bitterer Zynismus:) keine „überflüssigen“ Nützlichkeitsfragen stellt.

Wenn wir uns dann noch die akademische Forschung anschauen, wird der Griff zum Strick, mit dem man sich am liebsten erschießen möchte, zum Kinderspiel. Gefällt sich doch die wissenschaftliche Elite darin, „Algorithmen und Datenstrukturen“ zu beschreiben, UML-Diagramme zu malen und zu behaupten, dies sei der „Stand der Technik“. Mit Verlaub, schon die Architekten der Römerzeit waren eher in der Lage aufzuzeichnen, ein Bild von Ihrem Produkt zu malen, als die gesamte akademische Informatikelite es noch heute in Bezug auf Software ist.

Das Lamento, Software sei nun mal ein abstraktes Produkt und nicht so leicht in konkrete Bilder zu fassen, das mir als Antwort auf diese Klage immer wider entgegen schlägt, kann ich wirklich nur mit einem mitleidigen Lächeln begleiten. Herrschaften, wenn Euer Geist nicht in der Lage ist, das Abstrakte in konkrete Bilder zu fassen, dann verschont uns doch wenigstens mit den Ergebnissen Eurer nichtsnutzigen Forschung – geschweige denn Anwesenheit. Stehlt uns nicht die Zeit mit Vorträgen über Eure Unfähigkeit, sondern schweigt stille bis ihr das Problem gelöst habt.

Die Frage, die jeder informatische Akademiker vor Beginn, ja vor Genehmigung seiner Forschungsarbeit beantworten müsste, lautet: „Bist Du Teil der Lösung oder Teil des Problems?“. Bis heute sind sie ausschließlich Teil des Problems.

***Michael W. Dietrich***

*modulo3 GmbH  
Karl-Rudolf-Straße 172  
D-40215 Düsseldorf*

## Kommunikation als Wertschöpfungsfaktor in der Softwareentwicklung

Liegt es daran, dass viele Softwareentwickler zu introvertiert sind? Oder daran, dass Software immer noch nicht die ihr gebührende Stellung als eigenständige naturwissenschaftliche Disziplin erlangt hat. Ohne hier eine Lanze für die 'Debattierwissenschaften' brechen zu wollen, sind Qualität und Quantität des professionellen Disputs in der Softwareentwicklung im Vergleich zu anderen wissenschaftlichen Berufen unterentwickelt.

Davon abgesehen, dass Softwareentwickler dadurch noch größere Probleme als andere Naturwissenschaftler in der Außendarstellung haben mit entsprechenden Konsequenzen bis hin zu ihrer Bezahlung. Durch diese Eigenschaft werden Effizienzverbesserungen bestehender Systeme nicht in dem notwendigen Maße erreicht und dadurch Kosten erhöht.

Noch einschneidender ist, dass Innovation verhindert wird, die sich ergäbe, wenn Anwendungs-, Middleware- und Betriebssystemsoftware und Hardwarekomponenten gerade aus der Software-sicht ganzheitlich betrachtet würden.

Drei Beispiele mögen dies verdeutlichen:

Wiederverwendbarkeit: die Funktionen eines Systems bestehend aus Hardware, Software und Eingabe- / Ausgabeeinheiten werden dem Systembenutzer weitgehend durch die Software vermittelt. Der Bedarf an Funktionalität, leichter Bedienbarkeit, schnellen Antwortzeiten und guter Qualität wächst exponentiell. Die erforderlichen Software 'Lines of Code' wachsen dadurch stärker als die Zahl der verfügbaren Programmierer.

Wiederverwendbarkeit von Code ist also unabdingbar. Selbstverständlich ist, dass im Code möglichst viele offene Standards zum Zuge kommen. Darüberhinaus muss der Code aber transparent sein, d. h., seine Inhalte müssen bei seiner Entstehung und nach seiner Fertigstellung gut kommuniziert werden. Open Source bildet hier einen wichtigen Erziehungsfaktor.

Codewartung: in den meisten Fällen sind Softwareentwicklungs- und -Wartungspersonal nicht identisch. Heute müssen aufwendige und kostenträchtige Hierarchien aufgebaut werden, um auf Fehler-, funktionale oder Antwortzeitdefekte beim Kunden zu reagieren. Das geschieht umso schneller, dadurch kostengünstiger und zu größerer Kundenzufriedenheit je besser der Code dokumentiert im Sinne von kommuniziert ist. Dasselbe, was sich langsam in der Industrie bei Gebrauchsanweisungen durchsetzt, ist auch bei der Softwaredokumentation notwendig: Softwareentwickler, Dokumentationsentwickler und Kunde sollten gemeinsam die Qualität der Dokumentation vor ihrer Freigabe gründlich bewertet haben.

Neue Einsatzgebiete: Und nicht zuletzt: vergessen wir nicht, dass nur 6% aller Mikroprozessoren in unseren Lap-, Desktops und Großrechnern stecken. 94% sind in Autos, Mobiltelefonen, Fernsehern, Haushaltsgeräten, Smartcards und bald sogar in Kleideretiketten.

Zusätzlich zu den traditionellen Softwareaspekten kommen hier ganz neue Erfordernisse an den Softwareentwickler auf: Geringer Energieverbrauch, bedachtsamer Umgang mit kostbaren Rechner- und Speicherressourcen, hundertprozentiges Erfüllen der Benutzerbedürfnisse und das alles bei höchster Qualität.

Wie kann das erreicht werden, ohne dass der Softwareentwickler, mit seinen Rechnerhardwarekollegen, den Geräteentwicklern und vor allem den Kunden spricht? Industrie, Ausbildung, vor allem aber die Softwareentwickler selbst stehen hier vor großen Herausforderungen.

**Jörg Michael Thielges**

*Dir. Software Solutions & Services  
IBM Deutschland Entwicklung GmbH*

## Software–Entwicklung in Communicating Communities

Viele Software-Ingenieure glauben noch immer, das Allheilmittel für das Software Engineering sei die bessere Gestaltung des Software-Prozesses – also der geplanten Abläufe des Software Engineering. Ich gebe zu, ich habe mich schon immer schwer damit getan, die idealisierten Lebenszyklusmodelle ernst zu nehmen, wie immer sie auch ausgesehen haben und von wem immer sie auch massiv gestützt wurden. Um so mehr hat mich die Vorstellung, den Software-Prozess als Programm zu begreifen, gestört. Die vielen weichen Faktoren der Software-Entwicklung halte ich seit langem für die wesentlich wichtigeren, als die in Software-Prozessen darstellbaren harten Faktoren. Ich muss gestehen, dass mich nach meiner ersten Skepsis vor 15 Jahren der Mut verlassen hat, dies auch öffentlich lautstark zur Kenntnis zu bringen. Das bedaure ich heute – unabhängig davon, ob ich damals gehört worden wäre oder nicht. Noch schlimmer, während eines großen EUREKA-Projektes, an dem ich damals federführend beteiligt war, ist ein erheblicher Aufwand in die Entwicklung des Software-Prozess-Managements geflossen. Nicht überraschend war, dass diese im Projekt erarbeiteten Resultate eine eher kurzfristige Bedeutung hatten.

Ich denke inzwischen, dass »Software-Prozesse« ein in die Sackgasse führendes Konzept sind.

Um nicht missverstanden zu werden, auch ich bin der Meinung, dass Software Engineering organisiert werden muss. Aber nach meinem Dafürhalten muss vor allen Dingen die Kommunikation zwischen den Beteiligten organisiert werden. Große Software-Projekte erfordern die Einbindung der Beteiligten in immer wieder neue Kommunikations-Teams, die Organisation der Kommunikation in den Teams und die Kommunikation zwischen den Teams. Software Engineering zu organisieren, entspricht für mich heute der Aufgabe, »Communicating Communities« zu organisieren.

In Communicating Communities wird Wissen in den Köpfen der Beteiligten erzeugt, Information auf Basis dieses Wissens kommuniziert, Information zu Wissen in Beziehung gesetzt etc. Um dann dieses Wissen im Software Engineering als Wert betrachten zu können und gezielt zugänglich zu machen, ist eine Darstellung dieses Wissens durch Informationen und das dazu notwendige Informationsmanagement sicherzustellen. Die Abläufe, d.h. die Ordnung mit den die Abläufe bestimmenden Aktivitäten, spielen nach meiner heutigen Einschätzung eine nur untergeordnete Rolle.

Leider wissen wir Software-Ingenieure nur annähernd, wie Kommunikationsprozesse ablaufen und wie sie zu organisieren sind. Hier wird – sicherlich mehr mit Hilfe der Wissenschaftsdisziplinen, die sich mit der menschlichen Kommunikation beschäftigen – ein neuer Anfang nötig sein.

***Prof. Dr. Herbert Weber***

*Leiter des Fraunhofer Instituts ISST Berlin*