



ICB

Institute for Computer Science and
Business Information Systems

Seminararbeit

Entwicklung von Anwendungen für mobile Geräte

im Rahmen des Seminars
Mobile Informationssysteme
Wintersemester 2005/2006

Eingereicht am Lehrstuhl für
Wirtschaftsinformatik und Unternehmensmodellierung
Prof. Dr. Ulrich Frank
der Universität Duisburg-Essen

Betreuer: **Prof. Dr. Ulrich Frank**

Bastian Könings

Email: bastian@koenings.net
Matrikelnr.: 1393761

Datum: 2006-01-20

Inhaltsverzeichnis

1	Einleitung	5
2	Mobile Geräte	6
2.1	Abgrenzung	6
2.2	Anwender	6
3	Charakteristika der Softwareentwicklung für mobile Geräte	8
3.1	Restriktionen - Zugeständnisse an hohe Mobilität.....	8
3.1.1	User Interface.....	8
3.1.2	Kommunikation	9
3.1.3	Hardware-Ressourcen	10
3.1.4	Energieversorgung	11
3.1.5	Restriktionen im Blick zu erwartender Entwicklungen.....	11
3.2	Anforderungen	12
3.2.1	Integration	12
3.2.2	Awareness und Adaption.....	13
3.2.3	Skalierbarkeit	13
3.2.4	Trennung von Logik und Design.....	14
3.2.5	Synchronisation.....	14
3.2.6	Standardisierung	15
3.2.7	Sicherheit.....	15
3.2.8	Approximation an Nutzerbedürfnisse	15
3.3	Entwicklungsclassen.....	16
3.4	Anwendbarkeit moderner Softwareentwicklungsprinzipien	16
4	Entwurf eines Vorgehensmodells.....	17
4.1	Allgemeingültige Aspekte im Vergleich zu stationären Systemen....	17
4.2	Anforderungsanalyse.....	18
4.2.1	Intensive Einbeziehung der Endanwender	18
4.2.2	Machbarkeit.....	19
4.2.2.1	Integration im Spannungsfeld der Mobilität	19
4.2.2.2	Mobile Anwendungen für kritische Prozesse	19
4.3	Entwurf.....	20
4.3.1	Architekturauswahl anhand des Kommunikationsbedarfs	20
4.3.1.1	Mobile Standalone - Lösungen.....	20
4.3.1.2	Anwendungen mit synchronisierten Daten.....	21
4.3.1.3	Dienstvermittlung in mobilen Umgebungen	21
4.3.1.4	Anwendungen mit integrierter Unternehmensanbindung ..	21
4.3.2	Modellierung auf hohem Abstraktionsniveau	22
4.4	Implementierung, Test und Wartung.....	23
5	Resümee und Ausblick.....	24
	Anhang.....	27

Abkürzungsverzeichnis

EAI	Enterprise Application Integration
EDGE	Enhanced Data Rates for GSM Evolution
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communication
GUI	Graphical User Interface
HSCSD	High Speed Circuit Switched Data
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JINI	Java Intelligent Network Infrastructure
LAN	Local Area Network
PC	Personal Computer
PDA	Personal Digital Assistant
QoS	Quality of Service
RUP	Rational Unified Process
SMS	Short Message System
SyncML	Synchronization Markup Language
UIML	Universal Interface Markup Language
UMTS	Universal Mobile Telecommunication System
UPnP	Universal Plug and Play
VM	Virtual Machine
VUI	Voice User Interface
WLAN	Wireless Local Area Network
XML	Extensible Markup Language
XP	Extreme Programming
XSL	Extensible Stylesheet Language

Glossar

Action language.....	8	One-To-One Synchronisation	13
Anforderungsanalyse	17	Pattern	19
Awareness	12	Personal Digital Assistant	5
Business-To-Business.....	14	Pervasive Computing	5
Business-To-Customer	14	Pflichtenheft.....	19
Dezentralisierung.....	7	Presentation language	8
Diversifikation.....	7	Pro Activness.....	14
Elektronisches Papier	10	Process Execution Language for	
Embedded Systems.....	5	Webservices.....	18
Entwurf.....	19	Redundanz	13
Extensible Stylesheet Language	20	Simplizität.....	7
Extreme Programming.....	16	Smart Phone.....	5
Framework.....	15	Sollkonzept	17
Geschäftsprozess	12	Standardisierung	14
Global Positioning System	9	Synchronization Markup Language	20
Global System for Mobile Communication		Test	22
.....	9	Thin Client.....	21
Graphical User Interface.....	8	Touch Screen	8
Hypertext Markup Language	13	Universal Mobile Telecommunications	
Implementierung.....	22	System	9
Integration	11	Universal Plug und Play.....	20
Java 2 Enterprise Edition.....	21	User Interface.....	7
Java 2 Micro Edition.....	19	User Interface Markup Language.....	20
Java Intelligent Network Infrastructure .	20	Virtual Machine	15
Konnektivität	7	Voice User Interface	10
Laptop.....	5	VoiceXML.....	10
Laser-Tastatur.....	10	Vorgehensmodelle	16
Location-Awareness	12	Wartung	22
Many-To-Many Synchronisation.....	13	Webservice	18
Mobile Geräte.....	5	Wireless Access Protocol.....	13
Mobiltelefone	5	Workflow	12
Modell	21	Workflow-Management-System.....	18
Moor'sche Gesetz.....	11	XForms	20
One-To-Many Synchronisation	13		

1 Einleitung

„Our Future is Pervasive“ [HMNS03] lautet es vielfach in der Literatur und in Kongressbänden. Eine Zukunft, die vor allem durch Mobilität gekennzeichnet ist und die für die Unternehmen zahlreiche Einsatzmöglichkeiten mobiler Geräte insbesondere im Außen- und Kundendienstbereich aufzeigt. Die Rede ist von Systemen, die problemlosen Zugriff auf alle Unternehmensinfrastrukturen suggerieren, die Mitarbeiter beim Dienst am Kunden vor Ort direkt produktiv unterstützen, ein leichtes Customer Relationship Management ermöglichen und vor allem Geschäftsprozesse über die Grenzen des Unternehmens hinaus tragen. Zu früh wurde eine Aufbruchstimmung geschürt, welche auf Grund zu hoher Erwartungen und folgender Enttäuschungen bis heute in eine Negativstimmung umgebrochen ist. Zum Thema der mobilen Anwendungen titelte die Wirtschaftsinformatik: „Hammer sucht Nagel“ [RSF05], und brachte die aktuelle Situation mobiler Anwendungen auf den Punkt. Dennoch ist nicht zu leugnen, dass die Durchdringung des Alltags mit mobilen Geräten die Zukunft maßgeblich beeinflussen wird, was eine Betrachtung der Einflussfaktoren auf die Machbarkeit von sinnvollen mobilen Anwendungen lohnenswert macht. Diese Seminararbeit wird mobile Anwendungen aus einer softwaretechnischen Sicht betrachten, wobei zunächst die Restriktionen untersucht werden, die sich mit solcher Software verbinden. Weiterhin werden Anforderungen an mobile Anwendungen aufgestellt, die sich den Szenarien und Versprechungen aktueller Literatur entnehmen lassen, um deren Realisierbarkeit, aber auch, um deren Notwendigkeit für den Erfolg mobiler Anwendungen zu prüfen. Diese Erkenntnisse werden dann in ein exemplarisches Vorgehensmodell einfließen, das versucht, die komplexe Materie in die verschiedenen Phasen des Softwareentwicklungsprozesses einzuordnen. Dabei soll auch betrachtet werden, inwieweit herkömmliche Paradigmen und Konzepte der Softwareentwicklung auf mobile Anwendungen übertragbar sind.

2 Mobile Geräte

Unter einem mobilen Gerät wird ein komfortabel und handlich tragbarer, miniaturisierter Computer verstanden, der verschiedene Funktionalitäten in einer mobilen, also nicht stationären, Umgebung anbietet. Ergänzend ist ein mobiles Gerät nach [FrJu02] im Stande auf irgendeinem Wege zu kommunizieren, entweder online oder offline in einer Synchronisierung mit einem anderen System. Alleine im Jahr 2003 wurden weltweit 520 Millionen Mobilfunkgeräte[Gart04] verkauft, was die starke Marktdurchdringung der mobilen Geräte exemplarisch verdeutlicht. Heute lautet die Vision, mobile Geräte gänzlich in den Alltag der Menschen zu integrieren. Ein Prozess, der in der Literatur als „Pervasive Computing“ bezeichnet wird. Die Geräte wurden und werden dabei mit immer größeren Funktionalitäten und Anwendungen angereichert, so z.B. als Substitut für Adressbuch und Kalender.

2.1 Abgrenzung

In [FrJu02] werden Mobiltelefone, Personal Digital Assistants (PDA), Smart Phones und Embedded Systems als mobile Geräte abgegrenzt. Eine Einordnung der Geräte in Gruppen ist dabei mit steigender Kapazitäts- und Ressourcenausstattung recht komplex. Mobiltelefone haben ursprünglich ausschließlich der Kommunikation gedient. PDAs, und heute auch das Äquivalent Pocket PCs der Firma Microsoft, werden „vorwiegend von Berufstätigen unterwegs für das persönliche Informationsmanagement“ [HaNe01] genutzt. Dies beinhaltet die Funktionen eines mobilen Büros, aber auch die Anbindung an das Unternehmen durch spezielle Programme, durch die sich ein solches Gerät erweitern lässt. Smart Phones sind ein Resultat der schwindenden Abgrenzung der genannten Produktgruppen und deuten die Entwicklung zu nur noch einer mobilen Gerätegruppe an. Embedded Systems beschreiben mobile Systeme, die speziell für eine Funktionalität programmiert und gefertigt wurden. Man könnte an dieser Stelle noch weitere Geräte (MP3 Player, Spielkonsolen etc.) nennen. Diese Seminararbeit wird sich jedoch allein mit Anwendungen im unternehmerischen Umfeld auseinandersetzen. Laptops liegen nicht im Definitionsbereich mobiler Geräte, da diese mittlerweile ähnliche Rechenleistungen wie stationäre Personalcomputer (PC) erreicht haben und der Handlichkeitscharakter beim Transport nicht gegeben ist.

2.2 Anwender

Der Anwender eines mobilen Geräts, angelehnt an [BFar05], unterscheidet sich von einem stationären Benutzer durch die ständige Bewegung von einem Ort zum anderen und die Tatsache, dass dieser nicht auf den eigentlichen Prozess

auf dem mobilen Gerät fokussiert ist. Aufgaben, Vorhaben und Standorte des mobilen Anwenders ändern sich dabei spontan, und dieser hat die Anforderung an das Gerät jederzeit und an jedem Ort möglichst schnell auf Informationen zugreifen zu können. Mobile Anwender werden in der Literatur auch als Nomaden [BFar05] bezeichnet, die das Grundbedürfnis der Menschen nach freier Standortwahl ausleben.

3 Charakteristika der Softwareentwicklung für mobile Geräte

Die Softwareentwicklung für mobile Geräte zeichnet sich durch besondere Herausforderungen an den Entwickler aus, die sich in weiten Teilen von der herkömmlichen Softwareentwicklung abheben. Den mobilen Anwendungen für das Pervasive Computing können nach [BFar05] vier fundamentale Paradigmen zugeordnet werden. Diese lauten Dezentralisierung, Diversifikation, Konnektivität und Simplizität. Dezentralisierung beschreibt die hohe Verteilung und die damit verbundene Verbreitung mobiler Geräte über eine weite Unternehmenslandschaft. Der Punkt Diversifikation erklärt weiterhin, dass im mobilen Kontext Anwendungen und Geräte stärker auf die jeweiligen Nutzeranforderungen zugeschnitten werden, sodass unterschiedliche Tätigkeiten auch eine entsprechend hohe Anzahl an spezialisierten Lösungen mit sich führen. Das Paradigma Konnektivität meint, dass sich Pervasive Computing durch eine hohe Verbindungsdichte auszeichnet. Ein Gerät gelangt an jeder Stelle in jedes Netzwerk und kann mit anderen mobilen Geräten in seiner Nähe kommunizieren. Letztlich wird Simplizität angeführt, welche nicht besagt, dass mobile Anwendungen leicht zu programmieren seien, sondern dass komplexe Sachverhalte für den User transparent hinter einer benutzerfreundlichen Bedienmaske versteckt werden müssen.

3.1 Restriktionen - Zugeständnisse an hohe Mobilität

Die folgend vorgestellten Restriktionen entstammen, soweit nicht anders genannt, [FrJu02], [BFar05] und [MüWi02] und versuchen die Literatur in diesem Bereich zusammenzufassen. Restriktionen sind insbesondere dort relevant, wo die technische Entwicklung keine wirkliche Perspektive auf Besserung oder den Wegfall einer Restriktion verspricht. Alle Faktoren, die hier aufgeführt werden, wirken direkt oder indirekt auf die Benutzerakzeptanz eines Systems, welche, wie wir später noch sehen werden, ein zentraler Faktor für die Softwareentwicklung eines mobilen Gerätes ist.

3.1.1 User Interface

Unter einem User Interface versteht man die Schnittstellen, die dem Benutzer einer Anwendung zur Verfügung stehen, um mit dieser zu kommunizieren. Der Entwicklung und Gestaltung dieser Schnittstellen wird in einem Softwareentwicklungsprozess eine hohe Priorität beigemessen, da hier sowohl der Komfort als auch die Einfachheit im Umgang mit einem Gerät bestimmt, und somit direkt Einfluss auf die Benutzerakzeptanz genommen wird. [MyRo92]

beschreibt sogar: „ [...] in today's applications, an average of 48% of the code is devoted to the user interface [...] “. Bei näherer Betrachtung wird in der Literatur, als Beispiel sei hier [Coll95] genannt, das User Interface in eine Präsentations- (*presentation language*) und eine Interaktionssprache (*action language*) differenziert. Die Präsentationsprache befasst sich mit der Aufbereitung der Daten einer Anwendung hin zu einer für den Benutzer meist adäquat lesbaren Form auf einem Bildschirm. Bei mobilen Geräten unterliegt gerade diese populäre und als sinnvoll erwiesene Darstellungsform der erheblichen Restriktion kleiner Bildschirme. Der Wunsch möglichst großer Flächen für die Informationsdarstellung läuft dem Ziel der Portabilität entgegen. Die Heterogenität bei den Displaygrößen ist dabei sehr hoch und variiert unterhalb der 640x480 Pixel erheblich. Wichtig für den Benutzer eines Systems ist ein intuitiv bedienbares und optisch ansprechendes GUI¹. Die Frage lautet in diesem Kontext, wie es gelingen kann, eine Präsentation der Daten auf einer so kleinen Fläche ansprechend darzustellen. Aktuell kann dies nur so beantwortet werden, dass ein Layout sich auf die notwendigsten Elemente beschränken sollte (Simplizität).

Die Interaktionssprache beschreibt alle Komponenten des User Interfaces, die für eine Interaktion mit der Anwendung gedacht sind. Dies beinhaltet nach [FrJu02] die Manipulation der Visualisierung, die Erstellung, das Editieren und das Löschen der Anwendungsdaten. Typische Elemente der *action language* sind Textfelder, Checkboxes, Menüs und Listen. Doch auch hier ergeben sich wieder Diskrepanzen zwischen stationären und mobilen Systemen in der Art, wie diese Elemente bedient werden. Die klassischen und wichtigen Eingabegeräte Tastatur und Maus sind auf den mobilen Bereich nicht in voller Größe übertragbar, sodass an dieser Stelle nach Ersatz gesucht werden muss. Mobile Geräte werden z.B. über Touch Screens, virtuelle Tastaturen auf dem Screen und im Fall der Mobiltelefonen über wenige Bedienelemente und die Zeicheneingaben über den Ziffernblock bedient. Alle Verfahren sind weitaus weniger effektiv und kosten den Benutzer erheblich mehr Zeit als herkömmliche Eingabegeräte. Der Eingabekomfort steht dabei meist in Konkurrenz zu der Displaygröße, da hier eine große Eingabefläche eine Reduzierung der Displaygröße nach sich zieht.

3.1.2 Kommunikation

Die meisten Anwendungen unserer Zeit zeichnen sich durch hohen Kommunikationsbedarf aus. In der stationären Welt sind wir an eine permanente und vor allem schnelle und günstige Online- oder Netzwerkanbindung meist in Form einer Kabelanbindung gewöhnt. Im mobilen Bereich ist eine Kabelanbindung im Sinne der Mobilität keine Lösung, sodass die Technologie der Wahl zur Anbindung mobiler Geräte an andere Dienstleister im Netzwerk kabellos sein muss. Mit der Zeit haben sich mehrere mobile Datendienste zur Realisierung dieser Anforderung entwickelt. Der heutige de facto Standard

¹ GUI steht für Graphical User Interface und liefert die bildhafte Darstellung der für den Benutzer erforderlichen Informationen (vgl. auch [HaNe01]).

Global System for Mobile Communication (GSM), und die darauf aufbauenden Standards werden in Zukunft, aller Voraussicht nach, durch das Universal Mobile Telecommunications System (UMTS) ersetzt. Eine Übersicht der Verbindungsstandards und Details zur Technik sind Abbildung 1 zu entnehmen. Das zentrale Problem ist jedoch nicht die Frage,

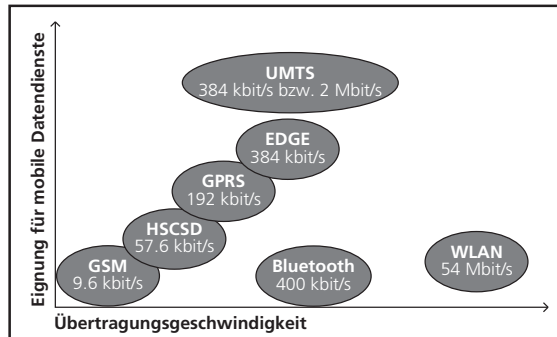


Abbildung 1 : Übersicht mobiler Datendienste [Vare04]

welche Technologie oder welcher Mix sich letzten Endes durchsetzen wird, sondern vielmehr die Tatsache, dass keine der Technologien eine 100% Netzabdeckung und garantierte Bandbreiten im Sinne von Quality of Service (QoS²) zur Verfügung stellen kann. Insbesondere die Technologie UMTS konnte im Punkt Geschwindigkeit bisher nicht überzeugen. So wurde zu Beginn mit Bandbreiten von bis zu 2 MBit geworben. Heute werden maximal 384 kbit/s erreicht, und auch diese Bandbreite kann nicht garantiert werden. Besonders gering wird die Übertragungsgeschwindigkeit, wenn sich viele Personen in einer Netzzelle befinden, da UMTS die Bandbreite teilt und somit die Übertragung mit der Anzahl der Dienstnehmer sinkt. Auf Grund mangelnder Genauigkeit der bisher genannten Netze zur Positionsbestimmung benötigt man für ortsbasierte Funktionalitäten weiterhin ein satellitengestütztes Positionsbestimmungssystem wie das Global Positioning System (GPS).

3.1.3 Hardware-Ressourcen

In der Anwendungsentwicklung wurde in den letzten Jahren der Blick auf Ressourcen wie Speicher und Prozessorleistung immer mehr vernachlässigt. Die derzeitigen stationären Rechnersysteme bieten Ausstattungen mit bis zu 3 Ghz Prozessorleistung. Auch Arbeits- und Festplattenspeicher sind mit großen Kapazitäten und zu erschwinglichen Preisen auf dem Markt erhältlich. Dem gegenüber stehen die mobilen Geräte, welche maximale Prozessorleistungen um die 600 Mhz und einen maximalen Arbeitsspeicher von 64 - 128 MB mit sich führen. Bei Handymodellen liegen die Leistungen noch deutlich darunter. Dies wirft die Problematik auf, dass sich mobile Anwendungen an diese Restriktionen geringerer Ressourcen anpassen müssen.

² Quality of Service (QoS) sichert eine minimale Verbindungsqualität zu, die sich durch Latenz und Varianz bestimmt (vgl. auch [HaNe01]).

3.1.4 Energieversorgung

Aufwendige Berechnungen, kabellose Kommunikation, Display und Eingabegeräte verbrauchen Energie. Insbesondere hohe Prozessorlast und Techniken wie UMTS etc. verbrauchen Strommengen, die eine Batterie, nach aktuellem Stand der Technik, nur begrenzt zur Verfügung stellen kann. Die Batterie ist zudem üblicherweise das größte Einzelteil eines mobilen Gerätes und nimmt nicht nur wichtigen Platz für Bedienelemente, sondern steigert ebenso das Gewicht. Ziel der Softwareentwicklung muss es also auch sein, Algorithmen so effizient zu schreiben, dass diese wenig Prozessorlast erzeugen und ebenso sollte durch die Software eine unnötige Kommunikation vermieden werden.

3.1.5 Restriktionen im Blick zu erwartender Entwicklungen

Vorgestellte Einschränkungen können im Kontext der zeitlich-technologischen Entwicklung mit unterschiedlicher Priorität für die zukünftige Softwareentwicklung eingestuft werden. Insbesondere bei User Interfaces finden derzeit Entwicklungen statt, die sich an der Aufhebung der Defizite in diesem Bereich versuchen. Ein viel versprechender Lösungsansatz ist die Interaktion mit den Geräten per Spracherkennung, bzw.

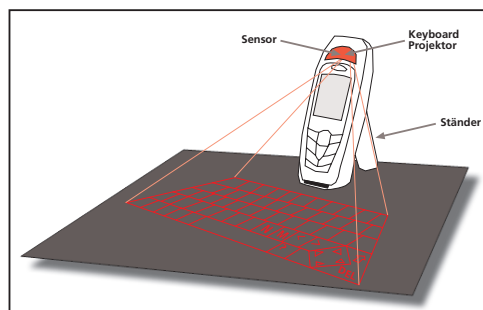


Abbildung 2: Laser-Tastatur [Vkb05]

Sprachgenerierung über sog. Voice User Interfaces (VUI). Vorstellbar wäre, dass ein mobiles Gerät Texte als Teil der *presentation language* vorliest und Befehle und Aktionen in der *action language* primär über Sprachbefehle erfolgen. Dies würde elementare Probleme der User Interfaces lösen, insbesondere dann, wenn ein solcher Ansatz mit herkömmlichem Rechnersystem integriert und somit der Bruch zwischen stationären und mobilen Systemen an dieser Stelle entfallen würde. Die Technik baut auf dem Ansatz auf, dass Mensch-Computer-Interaktionen als Datenflussdiagramme dargestellt werden können. Aktuell befinden sich diese Verfahren noch in einer rudimentären Entwicklungsphase. Ein bekannter Ansatz auf diesem Gebiet ist VoiceXML³ (vgl. zu diesem Thema auch [BFar05]). Eine greifbarere Technik ist die sog. Laser-Tastatur, wovon bereits erste Serienmodelle existieren. Ein Laser ist im Stande, eine Tastatur auf gerade Oberflächen zu projizieren, was die Mitführung einer Tastatur im mobilen Einsatz möglich machen würde. Aktuell noch als externes Modul sollen diese Bauteile in Zukunft auch als Bestandteil eines Mobiltelefons realisiert werden können (siehe Abbildung 2). Ebenfalls in der Entwicklung befindet sich das flexible und knickbare Display, das als elektronisches Papier bekannt ist. Dieses besitzt das Potenzial, herkömmliche Displays auf Grund von besserer Schärfe und

³ VoiceXML ist ein auf XML basierender Standard, um auf Inhalte des Internets über Sprache oder Telefon zuzugreifen (vgl. auch [HMNS03]).

geringerem Stromverbrauch zu ersetzen, aber auch große Displayflächen im mobilen Einsatz zu ermöglichen (vgl. auch [ScKa02]). Prinzipiell ist davon auszugehen, dass eine technologische Entwicklung in Zukunft auch die Restriktionen im Ressourcenbereich lockern wird. Nimmt man das Moor'sche Gesetz⁴ als Maßstab, so wird sich die Rechenleistung weiterhin erhöhen, auch bei mobilen Geräten. Zentral wird die Herausforderung sein, effektivere Batterien für diese Geräte zu entwickeln, da sowohl Kommunikation als auch Rechenleistung, daran gekoppelt sind. Als Fazit kann festgehalten werden, dass die technische Entwicklung in diesem Bereich starke Milderungen der Restriktionen in Aussicht stellt, diese vermutlich aber alsbaldig nicht außer Kraft setzen wird. Kleine Displays sind ein Bestandteil der Mobilität und auch eine permanente Konnektivität wird, wie vorher dargestellt, in naher Zukunft nicht erreichbar sein.

3.2 Anforderungen

Anwendungsszenarien zeigen auf, dass sich in einer mobil durchdrungenen Umgebung neue Anforderungen an effiziente Software ergeben. Hier seien als Beispiel die kontextbezogene Anpassungsfähigkeit an einen Benutzer, gute Integrationsstrukturen zur Unternehmensarchitektur und die Überwindung von Heterogenität genannt. In diesem Abschnitt werden die sich von der herkömmlichen Softwareentwicklung abhebenden oder besonders zu betonenden Anforderungen betrachtet.

3.2.1 Integration

In der Literatur werden 4 zentrale Ebenen der Integration im Bereich der Softwareentwicklung unterschieden. Diese Ebenen lauten in folgender Reihenfolge Daten-, Funktions-, Objekt- und Prozessebene (vgl. auch [HaNe01] und [DLPR02]). Dabei steigt die Wertigkeit der Integration mit jeder Ebene, und die vorigen Integrationsebenen sind jeweils Voraussetzung für die aktuell betrachtete Integrationsstufe. Der höchste Grad der Integration wird also auf der Prozessebene erreicht, wo neben Datentypen und Funktionsbeschreibungen auch Ereignisse durch ein gemeinsames semantisches Referenzmodell bestimmt werden. Die vornehmliche Zielsetzung der Integration im mobilen Bereich ist der „Austausch von Daten zwischen Mobilgerät und Zentralrechner, zwischen Arbeitsplatzrechner und Mobilgerät und auch zwischen verschiedenen Mobilgeräten“ [MüWi02]. Jedoch reicht eine Datenintegration nicht aus, da nicht nur Daten ausgetauscht, sondern auch Dienste von mobilen Geräten genutzt bzw. angeboten werden sollen. Dies setzt weiterhin die Anforderung einer Funktions- bzw. Objektintegration voraus.

⁴ Das Moor'sche Gesetz besagt, dass sich die Anzahl der Transistoren pro Chip etwa alle 18 Monate verdoppelt (vgl. auch [StHa02]).

In den letzten Jahren hat sich die Entwicklung auf die Betrachtung von Geschäftsprozessen⁵ im Unternehmen konzentriert, insbesondere im Bereich der Integration von Unternehmensorganisation und Prozessen mit den Applikationen. In [Goer03] schreibt der Autor: "Integration ist eine der wichtigsten Voraussetzungen, damit Unternehmen ihre wirtschaftlichen Ziele erreichen.". Dies unterstreicht die Anforderung der Integration auf Prozessebene, um das Konzept der Geschäftsprozesse als Workflow⁶ in die Softwareentwicklung zu übersetzen. Die Anforderung besteht also darin, eine Integration für mobile Anwendungen auf möglichst hohem Niveau zu erreichen, um eine wirtschaftliche Interoperabilität zwischen verschiedenen Gerätegruppen zu ermöglichen.

3.2.2 Awareness und Adaption

Stationäre Umgebungen erlauben eine Abstraktion von Änderungen der Umgebungsvariablen, da diese nie oder selten variieren. Ein mobiles Gerät jedoch wechselt häufig seinen Standort, was eine ebenso häufige Änderung von Kontext und Umfeld mit sich führt, insbesondere auch während der Benutzung. Eine Einbeziehung variierender Umgebungsparameter wird als Awareness (Bewusstheit), die Reaktion auf solche Änderungen als Adaption bezeichnet (Vgl. hierzu [MüWi02] sowie dort angegebene Quellen). Generell entsteht so ein neuer Anspruch an Informationen. Die Bewusstheit bezieht sich auf mehrere Aspekte. Hier soll exemplarisch die Orts- bzw. Location-Awareness angeführt werden. Die Software eines mobilen Gerätes sollte sich seiner Position „bewusst“ sein und dem Benutzer geg. ortsabhängige Services aber vor allem ortsrelevant gefilterte Informationen zur Verfügung stellen. Selbiger Anspruch auf Anpassung gilt auch für Kontext-, Situations- Netzwerk- und Energie-Awareness.

3.2.3 Skalierbarkeit

Dezentralisierung und hoher Kommunikationsbedarf sind ein Auszeichnungsmerkmal der mobilen Anwendungen. Für Anbindungen an Systeme z.B. einer Unternehmung ist also meist der Zugriff auf einen zentralisierten Server nötig. Auf Grund der aktuellen und zukünftigen weiten Verbreitung der mobilen Geräte als Alltagsgegenstände sollte eine oft nachträglich höhere Nutzlast der Systeme in Form der Skalierbarkeit in ein System eingeplant werden. Bei Durchsetzung einer Software- bzw. Gerätegruppe muss in dieser Umgebung mit deutlich höheren Gerätezahlen als im stationären Netzwerkbereich gerechnet werden.

⁵ „Ein Geschäftsprozess ist eine wiederkehrende Abfolge von Aktivitäten, die mehr oder weniger rigiden Regelungsmustern genügt. Er ist zielgerichtet und steht in einem direkten Zusammenhang mit der marktgerichteten Leistungserstellung eines Unternehmens“ [FrLa03]

⁶ Ein Workflows ist eine Abstraktion eines Geschäftsprozesses, welche primär auf den Fluss von digitalen Dokumenten und Objekten gerichtet ist (vgl. auch [FrLa03]).

3.2.4 Trennung von Logik und Design

Wie bereits beschrieben, zeichnet sich der Markt der mobilen Geräte auch durch starke Diversifikation, also durch hohe Heterogenität der verschiedenen Gerätetypen aus. Software hat hier den Anspruch auf möglichst vielen Geräten performant und optisch ansprechend ablaufen zu können. Insbesondere an die Allgemeingültigkeit des Designs, des GUI, werden höhere Anforderungen gestellt, als in der herkömmlichen Softwaretechnik, da auf eine große Anzahl von Displaygrößen eingegangen werden muss. Aber auch die Darstellungssprachen auf den Geräten variieren insbesondere im Browser-Bereich. Beispielsweise nutzen Geräte, neben zahlreichen anderen Sprachen, bei kleinen Displays üblicherweise WAP⁷, wohingegen neuere Geräte im Stande sind HTML⁸ zu interpretieren. Um nicht auf die Besonderheiten eines Gerätes mit einer jeweils neuen Programmlogik eingehen zu müssen, lautet eine weitere Anforderung an die Programmierung mobiler Anwendungen, das Layout von der Programmlogik, soweit es geht, zu trennen. Ebenso müssen Wege gefunden werden, ein Layout flexibel für viele Gerätetypen generieren zu lassen.

3.2.5 Synchronisation

Aus der nicht garantierbaren Konnektivität, welche bei den Restriktionen erläutert wurde, resultiert die Notwendigkeit, Daten auf mobilen Geräten bei Nichtverfügbarkeit einer Netzanbindung zu replizieren. Dieser Prozess widerspricht dabei den Konzepten der herkömmlichen Softwareentwicklung, da eine solche Zwischenspeicherung generell Redundanzen, also Mehrfachspeicherung gleicher Daten, erzeugt. Redundanzen gefährden die Integrität von Daten, da diese, abweichend von einer zentralen Speicherung, unkontrolliert an mehreren Stellen verändert werden können. Die Rückspeicherung dieser Daten benötigt in der Software effiziente Synchronisationsverfahren, die dafür Sorge tragen, dass die Integrität gewährleistet wird. Weiterhin muss durch die Anwendung dafür Sorge getragen werden, dass lokal gespeicherte Replica aktuell gehalten werden. An dieser Stelle wird in [HMNS03] die Synchronisation zwischen zwei Einzelgeräten (One-To-One), vielen verschiedenen Geräten zu einem Server (One-To-Many) und der Synchronisation zwischen beliebig vielen Geräten ohne zentralen Server (Many-To-Many) unterschiedenen. Ideal wäre im mobilen Rahmen die Many-to-Many Synchronisation, da hier eine Aktualität der Daten, unabhängig von einem Server, besser garantiert werden könnte. Auf Grund der damit verbundenen hohen Komplexität und einem Mangel an Lösungsverfahren dafür, wird in der Praxis jedoch durchgehend auf die One-To-Many Synchronisation gesetzt, die zwar ebenfalls sehr komplex, aber realisierbar ist.

⁷ Wireless Access Protocol (WAP) ist ein auf GSM basierender Standard, der sich auf die Anforderungen von mobilen Geräten konzentriert. WAP enthält die HTML ähnelnde Sprache WML zur Beschreibung von Informationen (vgl. auch [HMNS03]).

⁸ HyperText Markup Language (HTML) ist eine Sprache zur Beschreibung von Dokumenten im Internet

3.2.6 Standardisierung

Ein Ziel der Softwareentwicklung für mobile Geräte sollte es sein, Standards zu erfüllen und auch eine weiterführende Standardisierung zu unterstützen. Die Vision: Ein mobiles Gerät betritt einen fremden Raum, und kann dort alle zur Verfügung stehenden Dienste (Drucker, Beamer, etc.) nutzen. Die Standardisierung ist Voraussetzung und Bestandteil des Integrationsprozesses der mobilen Geräte und eines solchen Szenarios der Interoperabilität, da eine Verbindung mit unterschiedlichen Protokollen, Daten und Funktionen sonst nicht hergestellt werden könnte. Insbesondere bei einer Integration von mehreren Unternehmen (Business-To-Business) und zwischen Unternehmen und Kunden (Business-To-Customer) ist die Nutzung gemeinsamer Standards essenziell, insbesondere auch bei der Nutzung gemeinsamer Geschäftsdokumente und der „Integration von Anwendungssystemen entlang von unternehmensübergreifenden Wertketten“, [Fran01].

3.2.7 Sicherheit

Wie andere Gegenstände des Alltags, so z.B. die Geldbörse, kann ein mobiles Gerät außerhalb des Firmengeländes sehr viel leichter dem Diebstahl unterliegen, als stationäre Systeme innerhalb einer Firma. Auch bei einer kabellosen Verbindung besteht ein wesentlich höheres Risiko abgehört zu werden, als bei einem durch feste Leitungen verbundenen Kommunikationsweg. Somit muss die Software an dieser Stelle entsprechende Sicherheitsmaßnahmen, wie Verschlüsselung oder Fingerabdrucktechnologien, für empfindliche Anwendungen mit einbeziehen.

3.2.8 Approximation an Nutzerbedürfnisse

Die diffizilste Anforderung stellt nach [Lueg02] die Anpassung der mobilen Anwendungen an die kontextbasierten und nutzergewünschten Situationen dar. In kaum einem Anwendungsszenario im Bereich des Pervasive Computing wird nicht auf die Darstellung der Möglichkeit verzichtet, dass mobile Anwendungen die Fähigkeit besitzen, sich in weiten Teilen auf das Verhalten von Menschen einzustellen. Der Definition des mobilen Anwenders zu Folge, darf das Verhalten eben dieser als komplex bezeichnet werden. Der Autor in [Lueg02] bezeichnet ein anpassendes Verhalten der Systeme als „Pro Activeness“ und schreibt, dass diese Funktionalität als Science Fiction angesehen werden muss. Eine Anpassung, schon um simple Dinge abzubilden, würde eine Intelligenz voraussetzen, die der des Menschen ähnlich wäre. Weiterhin müsste dem System eine große Menge an Informationen über die umgebende Welt zur Verfügung stehen, die sich auch weiterhin stetig ändert und eine Abbildung daher noch schwieriger macht. Ein Lösungsansatz, der eine derartig künstliche Intelligenz und somit eine Approximation an das menschliche Verhalten zur Verfügung stellen kann, ist derzeit nicht in Sicht.

3.3 Entwicklungsklassen

In Softwareprojekten für mobile Geräte lassen sich 3 Entwicklungsklassen unterscheiden. Die erste Klasse beschreibt Software, die ausschließlich für mobile Geräte entwickelt wird. Dies bedeutet im Allgemeinen Einzelanwendungen ohne hohen Kommunikationsaufwand zu anderen Anwendungen und maximal einer Synchronisation mit einer Workstation. Die zweite Klasse enthält Anwendungen, die auf einer bereits existierenden stationären Softwarearchitektur aufbauen und in diese integriert werden sollen. Anwendungen, die sowohl für mobile, als auch die für stationäre Geräte der Metapher nach „auf der grünen Wiese“ völlig neu und ohne Altlasten erstellt werden, gehören zu der dritten Klasse.

3.4 Anwendbarkeit moderner Softwareentwicklungsprinzipien

Als Errungenschaft der letzten Jahre setzt die Softwareentwicklung nach [FrJu02] zunehmend auf Konzepte, die auf der Idee der Wiederverwendung basieren, wie z.B. der plattformunabhängige Programmiercode. Nach dem Prinzip: „Write once, run anywhere“ [HMNS03], nutzen neuere Programmiersprachen wie Java oder Smalltalk eine sog. Virtual Machine (VM), die für die meisten Rechnersysteme zur Verfügung gestellt wird, um dort Anwendungen plattformunabhängig auszuführen. Diese VM erzeugt aber einen Overhead, der die knappen Ressourcen eines mobilen Gerätes zusätzlich belastet. Ein weiterhin populäres Konzept sind Frameworks, welche ein durch verschiedene Methoden anpassbares, wieder verwendbares Design zur Verfügung stellen, das zumeist aus vorgefertigten Softwarekomponenten besteht (vgl. auch [HaNe01]). Auch bei diesem Konzept wird zusätzlicher Overhead erzeugt, da durch den generischen Charakter der Frameworks keine spezifischen, sondern allgemeine Lösungen angeboten werden. Häufig stellt ein Framework Funktionalitäten bereit, die von einer darauf resultierenden Anwendung nicht benötigt werden. Die nicht benötigten Teile belegen aber dennoch Ressourcen. Generell lässt sich sagen, dass Sprachen die näher an der spezifischen Maschinensprache arbeiten, wie z.B. Assembler⁹, sich optimierter, und somit mit höherer Performanz an die Ressourcenanforderungen der mobilen Umgebung anpassen lassen. Wiederverwendung als Solche ist natürlich auch für mobile Umgebungen ein wichtiges Ziel. Moderne Prinzipien, wie oben aufgeführt, bedürfen meist einer Anpassung an den mobilen Kontext, wobei eine Annäherung mit steigenden Ressourcen der mobilen Geräte möglich erscheint.

⁹ „Ein [...] Assembler ist ein Übersetzungsprogramm, das ein in der Assemblersprache abgefasstes Quellprogramm in eine Maschinensprache umwandelt“ [HaNe01]

4 Entwurf eines Vorgehensmodells

Vorgehensmodelle sind eine fest etablierte Komponente der Softwareentwicklung und erscheinen in der Literatur in zahlreicher Ausprägung (vgl. [StHa02] oder [HaNe01]). Typischerweise werden diese Modelle in Phasen eingeteilt, die ein Entwicklungsprozess durchlaufen soll. Generell haben die Modelle den Anspruch, sich die Problematik und die Anforderungen einer Entwicklung bewusst zu machen. Durch einen besonders hervorgehobenen Kommunikationscharakter, soll die Grundlage dafür geschaffen werden, diese innerhalb einer Gruppe diskutieren zu können. Grob lassen sich alle Modelle in die nacheinander folgenden Phasen Anforderungsanalyse, Entwurf, Implementierung, Test und Wartung einteilen. Bei diesen Ansätzen handelt es sich meist um sehr komplexe Gebilde, die in manchen Programmiererkreisen als bürokratisch und zu kompliziert gelten, da diese ausgiebige Dokumentation und aufwändige Planung erfordern. In den letzten Jahren hat sich der Prozess des Extreme Programmings (XP) verbreitet, der nach dem Prinzip der Einfachheit auf Dokumentation und starke Modellierung verzichtet. Durch direkte Miteinbeziehung des Kunden und zahlreiche andere Methoden, siehe [Jeff01], stellt XP den Anspruch, sich flexibler auf stetig ändernde Kundenwünsche einstellen zu können. Ein Ansatz, der in kombinatorischer Weise mit herkömmlichen Entwicklungsprinzipien für mobile Geräte insbesondere im Interface Design denkbar wäre.

4.1 Allgemeingültige Aspekte im Vergleich zu stationären Systemen

Obschon Ansätze wie Extreme Programming existieren, gilt, wie auch bei der herkömmlichen Softwareentwicklung, dass eine gute und strukturierte Planung eines größeren Projektes Voraussetzung für den Projekterfolg ist. Laut [Stan01] sind im Jahr 2000 auf Grund mangelnder Planung die Hälfte der Softwareprojekte nicht im Planungshorizont abgeschlossen worden, und erzeugten höhere Kosten oder erfüllten nicht die Anforderungen

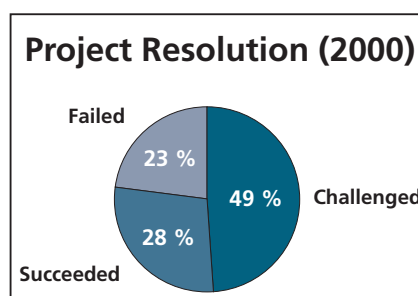


Abbildung 3 : Projektauflösung [Stan01]

(siehe Abbildung 3). 23 % der Projekte sind gar gescheitert. Letztlich sollten mobile und herkömmliche Softwareentwicklung gleiche Ziele verfolgen, wobei der Aspekt einer guten Planung, auf Grund der meist hohen Komplexität mobiler Umgebungen, in der Anforderungsanalyse und dem Entwurf der zentrale Faktor im Softwareentwicklungsprozess sein sollte. Dabei muss natürlich der Nutzen der

Planung die Kosten rechtfertigen können. Oft kann ein solcher Prozess auch nicht unabhängig voneinander betrachtet werden, da eine parallele Entwicklung für beide Systeme in einem Projekt denkbar ist. Gerade wegen der engen Verbundenheit mit stationären Systemen, sollten sich mobile Entwicklungsprozesse mit modernen Konzepten wie Objektorientierung oder Softwarekomponenten¹⁰ auseinander setzen, um, bei Eignung für das jeweilige Problem, eine bessere Integration der Entwicklungsprozesse zu ermöglichen. Bei der Miteinbeziehung von mobilen Geräten in den Entwicklungsprozess sind jedoch zusätzliche Faktoren zu beachten, die im Folgenden erläutert werden.

4.2 Anforderungsanalyse

Die Phase der Anforderungsanalyse steht am Anfang des Softwareentwicklungsprozesses und legt den wichtigsten Grundstein für das System. Die Klärung dessen, was genau der Kunde erwartet, ist essenziell und bedingt die Wertschätzung dieser Phase für das entstehende Softwareprodukt. Nach [RSF05] unterlagen die mobilen Anwendungen einem Hype, der vor allem die Diskussion, nicht aber die tatsächliche Marktentwicklung maßgeblich beeinflusste. Dies war ein Resultat zu hoher Erwartungen in die Möglichkeiten mobiler Systeme. Ziel ist also die Erstellung eines sog. Sollkonzeptes unter der Bewusstmachung und somit Vermeidung einer Gefahr von zu hohen Anforderungen. Ebenfalls von Bedeutung, bei einer bereits existierenden Softwarearchitektur, ist vor dem Sollkonzept die Erhebung des Istzustandes, um mögliche Integrationspotenziale und später einen Vergleich ermitteln zu können.

4.2.1 Intensive Einbeziehung der Endanwender

Von der Anforderungsanalyse ausgehend, sich dann über alle Phasen der Softwareentwicklung erstreckend, sollte der Endanwender in diesen Prozess mit eingebunden werden, da mobile Anwendungen mehr als stationäre Systeme einer teils mangelnden Benutzerakzeptanz unterliegen. Dies ist zum einen Teil auf die Restriktionen der User Interfaces zurückzuführen, die ein mobiles Gerät beschränken, aber auch einer komplexen soziologischen Problematik zuzuschreiben. Diese Problematik beinhaltet die Frage, ob ein Benutzer eine mobile Anwendung unterwegs, in einem sozialen Kontext, mit gesellschaftlichen Verpflichtungen und Zwängen duldet (vgl. hierzu [HFHH05]). Vielfach besteht „Unsicherheit bezüglich der Anforderungen und Bedarfe mobiler Kunden und Mitarbeiter“ [RSF05], was natürlich die klare Erstellung eines Sollkonzeptes schwierig gestaltet. In bestimmten Komponenten oder Bereichen, wie z.B. der GUI-Implementierung, sollte deshalb z.B. auch der Einsatz von XP erwogen werden, da sich so im Zeitverlauf ändernde Kundenbedürfnisse besser kompensieren ließen.

¹⁰ Softwarekomponenten stellen über eine wohldefinierte Schnittstelle Funktionen zur Verfügung, und sind wieder verwendbare und durch kompatible Komponenten ersetzbare Softwarefragmente (vgl. auch [HaNe01])

4.2.2 Machbarkeit

Der vorige Abschnitt hat die Mängel insbesondere bei Display, Kommunikation und der intelligenten Anpassung an den User erläutert. Vorlagen in der Literatur und im Anhang dieser Arbeit müssen daher auf ihre Machbarkeit hin untersucht werden.

4.2.2.1 Integration im Spannungsfeld der Mobilität

Eine Kommunikationsverbindung im Bereich mobiler Geräte bedeutet im Regelfall lose Kopplung zwischen den verteilten Softwareelementen. Die Elemente kommunizieren dabei nicht mit Referenzen auf den gemeinsamen Speicher, sondern über Schnittstellen. In diesem Kontext sind Schnittstellen meist auf einem recht niedrigen semantischen Niveau beschrieben, um performante und kompatible Kommunikation zu erzielen. Insbesondere Webservices (vgl. [BHMN04]) konnten sich hier als Technologie im mobilen Bereich etablieren. Wie zuvor gefordert, sollen nun aber die Geschäftsprozesse auf dynamischem Niveau integriert werden. Die populärste Möglichkeit dazu lautet, die Unternehmensprozesse auf Webservices abzubilden und so die Prozessintegration für z.B. Workflow-Management-Systeme (vgl. [HaNe01], S.445) herzustellen. Ein bekannter Ansatz hierfür ist Business Process Execution Language for Webservices (BPEL4WS), welcher neben der Funktionsbereitstellung durch Webservices auch den Kontrollfluss zwischen diesen Services regelt, die unternehmensintern, aber auch unternehmensextern genutzt werden können (vgl. [ACDG03]). Das Spannungsfeld stellt hier die Problematik, dass die verschiedenen Anwendungen hinter den Webservices kein gemeinsames, semantisches Referenzmodell besitzen und Webservices weiterhin generell von Datenhaltung als solcher abstrahieren. Zwar scheint auf den ersten Blick eine dynamische Integration hergestellt zu sein, aber es fehlt eine statische Integration der Daten. Wie erläutert, baut eine dynamische Integration auf der statischen auf, sodass am Beispiel dieses Ansatzes keine wirkliche Integration hergestellt wird. Unterschiedliche Datenhaltungen erzeugen in der Regel Redundanzen, was Gefahren für die Datenintegrität bedeutet. Die Techniken in diesem Bereich müssen also insbesondere im Sinne der Integration sorgfältig begutachtet werden.

4.2.2.2 Mobile Anwendungen für kritische Prozesse

Eine weitere Frage muss lauten, inwieweit sich kritische Prozesse über mobile Anwendungen verteilt ausführen lassen. Ein kritischer Prozess zeichnet sich auch durch die Notwendigkeit zeitnaher Verarbeitung aus. Würde nun auf dem Server ein kritischer Prozess als Teil des Workflows gestartet, an dem ein mobiles Gerät beteiligt ist, so kann es passieren, dass zum Ausführungszeitpunkt dieses Gerät nicht im Stande ist eine Verbindung zum Server herzustellen, oder dass dieses schlicht nicht eingeschaltet ist. Dies würde bedeuten, dass der Prozess solange blockiert ist, bis wieder eine Verbindung mit dem Gerät hergestellt wurde. Daher ist eine Nutzung der mobilen Geräte für kritische Prozesse aktuell nicht denkbar.

4.3 Entwurf

Der Entwurf greift als zweite Phase der Softwareentwicklung das Sollkonzept auf und verfeinert nun darauf basierend, detailliert die Systemeigenschaften für die folgende Implementierung. Ein sog. Pflichtenheft wird erstellt, das alle Funktionalitäten, die eine aus dem Planungsprozess resultierende Software erfüllen muss, detailliert dokumentiert. Dabei sollte sich der Entwurf, ganz im Sinne der Wiederverwendung, daran orientieren, ob aktuelle Anforderungen nicht bereits durch ein sog. Pattern¹¹ mustergültig implementiert wurden. Dies setzt natürlich die Prüfung zur Eignung der meist aus der herkömmlichen Softwaretechnik stammenden Pattern für mobile Anwendungen voraus.

4.3.1 Architekturauswahl anhand des Kommunikationsbedarfs

Eine der wichtigsten Entscheidungen die im Entwurfsstadium getroffen wird, ist die Auswahl der geeigneten Architektur und Standards für eine Softwarelösung. Jede nachträgliche Änderung an dieser Stelle verursacht hohe Kosten oder gar das Scheitern des Projektes. Daher sollte eine Auswahl hier sehr sorgfältig vorgenommen werden. Eine mittlerweile sehr etablierte Programmierumgebung ist die Java 2 Micro Edition (J2ME), welche eine speziell auf die Bedürfnisse der mobilen Geräte modifizierte Version der Java Standard Edition darstellt (vgl. hierzu [KrHa03]). Dies hat den Vorteil, dass durch die Technik der Virtual Machine eine Plattformunabhängigkeit erreicht wird, und diese Version keine Ressourcen mit nicht benötigten Funktionen belegt. Dennoch hält [Goet04] als Resultat seiner Softwareentwicklung fest, dass nach heutigem Stand der Technik noch situationsabhängig über die Programmierumgebung für ein mobiles Gerät entschieden werden muss. Insbesondere bei Anwendungen mit aufwändigeren Berechnungen sollte der Griff zu systemnäheren Sprachen als Java aktuell noch erwogen werden.

Ein wichtiger Aspekt bei der Auswahl der Architektur, wobei meist auf Drittanbieter und Standards zurückgegriffen wird, ist auch die Betrachtung des Investitionsschutzes. Bei vielen Technologien in dieser jungen Softwaredisziplin sind deren langfristige Durchsetzung auf dem Markt, und damit die Existenz von Support und Weiterentwicklung unklar. Ziel eines Architekturentwurfs muss auch immer die Realisierung langer Laufzeiten eines Systems sein.

Die folgenden Kategorien versuchen die Software zu gruppieren und geg. konkrete Techniken zur weiteren Recherche vorzustellen, wobei mit jeder Stufe der Kommunikationsaufwand steigt.

4.3.1.1 Mobile Standalone - Lösungen

Diese Softwarekategorie deckt kleinere Softwareprogramme mit fehlender Anbindung an andere Systeme ab. Darunter verstehen sich meist kleinere Tools und möglicherweise Spiele - Anwendungen zur Unterhaltung. Größere Konzepte

¹¹ Patterns (dt. Muster) sind bewährte Lösungen einer softwaretechnischen Problematik, die sich in der Praxis bewährt haben.

der Integration müssen dabei meist nicht beachtet werden. Jedoch sollte, um möglichst viele mobile Geräte mit einer solchen Anwendung bedienen zu können, der Einsatz von generischen GUIs angedacht werden. Die Erstellung eines User Interfaces für mehrere Geräte könnte z.B. durch die Einbeziehung von Sprachen wie Extensible Stylesheet Language (XSL), User Interface Markup Language (UIML) oder XForms realisiert werden (vgl. hierzu [BFar05]).

4.3.1.2 Anwendungen mit synchronisierten Daten

Typischerweise handelt es sich bei dieser Gruppe um Anwendungen, die sporadisch und in meist unregelmäßigen Abständen Daten, wie Adressen, Termine und persönliche Dateien, gegen ein stationäres System oder ein anderes mobiles System auf einer semantisch geringen Datenintegrationsebene abgleichen. Die Typen der Synchronisation wurden dabei bereits vorgestellt. Ein wichtiger Standard stellt in dieser Kategorie die plattform- und netzwerkunabhängige Synchronization Markup Language (SyncML) dar (vgl. [BFar05]).

4.3.1.3 Dienstvermittlung in mobilen Umgebungen

Die Anforderungen dieser Softwaregruppe verlangen, dass mobile und stationäre Geräte untereinander auf einer funktional integrierten Ebene Dienste in der Nähe nutzen und anbieten können. Vor allem aber sollen die Anwendungen zur gegenseitigen Verbindungsaufnahme keinerlei Administration benötigen. Derzeit finden sich jedoch noch keine Architekturlösungen, die eine solche Funktionalität uneingeschränkt zur Verfügung stellen, wobei die Ansätze Universal Plug and Play (UPnP) von Microsoft und Java Intelligent Network Infrastructure (Jini) von Sun (vgl. hierzu [HMNS03]) die größten Aussichten auf Erfolg haben. Zentrale Voraussetzung ist hier vor allem eine möglichst weltweite Standardisierung eines Dienstes, um möglichst weiträumig diesen Service zur Verfügung stellen zu können.

4.3.1.4 Anwendungen mit integrierter Unternehmensanbindung

Die für Unternehmen mit Abstand interessanteste Softwaregruppe definiert Anwendungen, die integriert mit der Firmenarchitektur kommunizieren. Dabei muss geklärt werden, an welcher Stelle sich mobile Geräte in der Architektur befinden. Ihr größtes Potenzial nutzen die mobilen Geräte in der Generierung und im Abruf der Informationen (siehe

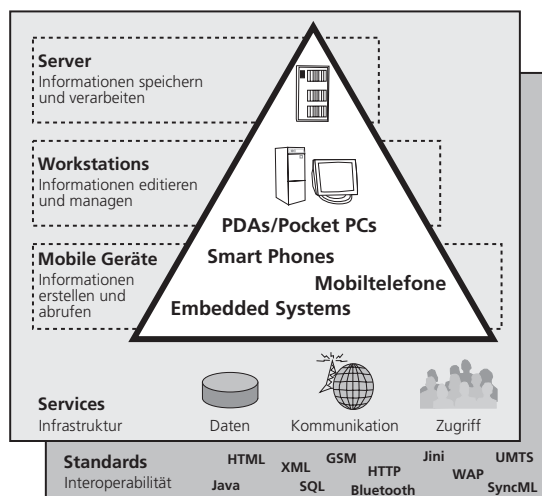


Abbildung 4 : Generischer Blick auf pervasive Informationstechnologie [HMNS03]

Abbildung 4), insbesondere im Außendienst, um so Medienbrüche z.B. durch vorherige Übertragung auf Papier zu vermeiden. Die Speicherung und Verwaltung der Daten erfolgt sinngemäß über stationäre Systeme. Das mobile Gerät interagiert in einer solchen Umgebung als Client, sodass die Frage lauten muss, wie viel Anwendungslogik auf dem Client implementiert werden soll.

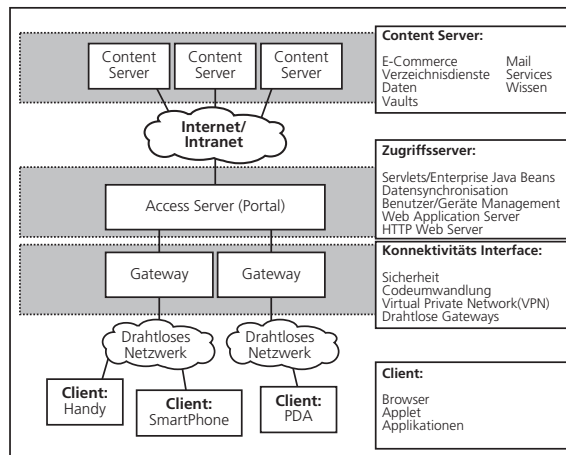


Abbildung 5 : Back-End Server Infrastruktur [HMNS03]

Idealtypischerweise würde das mobile Gerät einen sog. Thin Client darstellen, der die gesamte Programmlogik z.B. über einen Browser von einem Server abfragt. Das Problem hierbei ist wiederum die mangelnde Konnektivität, sodass ein Arbeiten während der Nichtanbindung an den Server nicht möglich wäre. Das Gleichgewicht zwischen Applikationen, die auf dem Client ausgelagert werden, und der Ausführung der Applikationen auf dem Server muss hier in einen sinnvollen Nutzenbezug gesetzt werden. Abbildung 5 zeigt eine klassische 3 Schichten Architektur, wobei die mobilen Geräte über Gateways¹² auf den Access Server zugreifen. Die Vorteile der Bearbeitung auf dem Server liegen auf der Hand. Es gäbe keine redundanten Datenspeicherungen und die Programmlogik läge auf dem Server, was somit auch die Ressourcen des Clients schonen würde. Weiterhin benötigten die mobilen Geräte nur einen Browser, der z.B. WAP oder HTML versteht, sodass auf die speziellen Anforderungen der User Interfaces wie eingeschränkte Bildschirmgröße leichter eingegangen werden könnte. Ein Beispiel für eine solche Application Server Architektur ist die Java 2 Enterprise Edition (J2EE) Spezifikation von Sun (vgl. [HMNS03]).

4.3.2 Modellierung auf hohem Abstraktionsniveau

Wie bereits in der Anforderungsanalyse festgelegt, soll sich auf Grund der Komplexität mobiler Anwendungen und der unklaren Anforderungen der Anwender eine ständige Kommunikation der Shareholder eines Projektes als roter Faden durch den Softwareentwicklungsprozess ziehen. Diese Kommunikation kann in der Regel nur durch Modelle geleistet werden. Modelle verstehen sich dabei als abstrahierte Abbilder der Realität. Da zumeist verschiedene Parteien an der Entwicklung einer Software beteiligt sind, bedarf es einer Abstraktion dieser Modelle auf hohem Niveau. Das bedeutet, dass man für

¹² Gateways stellen Funktionalitäten für bestimmte Gerätegruppen zur Verfügung, so wandeln diese z.B. Plain-Text in ein gültiges WAP-Format (vgl. auch [HMNS03]).

und mit den entsprechenden Zielgruppen Modelle entwerfen muss, die unrelevante Aspekte für diese Gruppe ausblenden, und somit diese Modelle verständlicher und übersichtlicher machen. Ein weiterer Vorteil der Modelle ist die Dokumentation, welche durch diese geleistet wird, wobei dies insbesondere nachträgliche Änderungen der Software erleichtert. Die Modelle mit den verschiedenen Abstraktionen sollten jedoch in jedem Falle untereinander integriert sein, um sicher zu stellen, dass verwendete Elemente in allen Modellen, so z.B. für stationäre und mobile Teilmodelle, die gleiche Bedeutung haben.

4.4 Implementierung, Test und Wartung

Die drei Phasen der Implementierung, Test und Wartung verlaufen in der Regel in ähnlichen Mustern wie die der herkömmlichen Softwareentwicklung, jedoch sollte der höhere Aufwand in diesen Phasen beachtet werden. Entscheidet man sich z.B. gegen eine plattformunabhängige Lösung und versucht dennoch mehrere mobile Gerätetypen nutzen zu können, so steigt der Aufwand mit jeder zusätzlichen Plattform, die bedient werden muss, erheblich. Auch der Test, also die Überprüfung der Anforderungen, die im Pflichtenheft formuliert wurden, muss auf einer größeren Zahl von Geräten und somit mit höherem Aufwand erfolgen. Weiterhin ergibt sich in der Phase der Wartung die Problematik, dass ein später entwickeltes Update auf Grund der hohen Anzahl der Geräte nicht so leicht installiert werden kann, wie bei stationären Systemen. Möglicherweise sind mobile Anwendungen längere Zeit von einem zentralen Server getrennt, sodass kritische Updates nicht zeitnah flächendeckend auf allen Geräten installiert werden können.

5 Resümee und Ausblick

In dieser Seminararbeit wurden die verschiedenen Aspekte der Entwicklung von Anwendungen für mobile Systeme analysiert. Dabei wurden vor allem Restriktionen, Anforderungen und die Machbarkeit einer mobilen Zukunft im Sinne der Softwareentwicklung des Pervasive Computing betrachtet. Resultierend muss nun festgehalten werden, dass der Hype und die Versprechungen, die mit dieser Vision verbunden sind, teilweise in absehbarer Zukunft aus softwaretechnischer Sicht schlicht nicht realisierbar sind. Dazu gehört vor allem eine ständig garantierte und schnelle Anbindung an das Unternehmen und die Awareness im Kontext mangelnder intelligenter Anpassungen an den mobilen Benutzer. Um über heute bereits akzeptierte Anwendungen hinaus zu gelangen, müssen sich die mobilen Anwendungen daran messen lassen, ob die Integration auf möglichst hohem Niveau zur Realisierung von mobilen Geschäftsprozessen gelingt. Gerade in diesem Bereich liegt das größte Potenzial der mobilen Anwendungen, da hier ökonomischer Nutzen durch mehr Effektivität, Flexibilität und Bearbeitungsschnelligkeit von Prozessen in einer Unternehmung gewonnen werden kann. Aber auch unterhalb dieser Ebene lassen sich produktive Einsatzmöglichkeiten finden, insbesondere dort, wo im Außeneinsatz viele Daten gesammelt werden müssen. Mitarbeiter können dann auf eine schriftliche Erfassung der Daten zugunsten direkter Eingabe in das mobile Gerät verzichten, was spätere Medienbrüche durch die netz- oder synchronisationsgestützte Übertragung in die Unternehmensperipherie vermeidet. Festzuhalten bleibt, dass die Anwendungsentwicklung für mobile Geräte zumeist mit erheblich höherem Aufwand im Vergleich zur herkömmlichen Softwareentwicklung verbunden ist. Dies resultiert aus der hohen Heterogenität der Geräte, welche unter der Beachtung von aufgestellten Restriktionen bedient werden muss. Ebenso muss ein zusätzlicher Aufwand bei der Überprüfung von in der herkömmlichen Softwaretechnik etablierten Verfahren bezüglich deren Eignung für mobile Anwendungen betrieben werden. Für den Softwareentwicklungsprozess ist es unverzichtbar, eine überlegte Planung voran zu stellen, um insbesondere die Anforderungen im Rahmen der Machbarkeit abzustecken und die Architektur der Software sinnvoll zu wählen. In Zukunft deutet sich die Milderung oder gar der teilweise Wegfall der Restriktionen an, was den mobilen Anwendungen dann eine solidere Plattform für eine breite Durchsetzung in der Unternehmenslandschaft zur Verfügung stellen sollte. Auch eine Weiterentwicklung von Integrationsarchitekturen wie Jini und die starken Bestrebungen Standards zu schaffen versprechen weitere Potenzialzugewinne mobiler Anwendungen. Die Aussage „The Future ist Pervasive“ gilt noch immer, wobei diese Zukunft nicht gestern war, sondern heute gerade erst begonnen hat.

Literaturverzeichnis

- [ACDG03] Andrews, T.; Curbera, F.; Dholakia, H.; Golland, Y.; Klein, J.; Leymann, F.; Liu, K.; Roller, D.; Smith, D.; Thatte, S.; Trickovic, I.; Weerawarana, S.: Business Process Execution Language for Web Services - Version 1.1. BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems, 2003
- [BFar05] B'Far, R.: Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML. Cambridge : Cambridge University Press, 2005
- [BHMN04] Booth, D.; Hass, H.; McCabe, F.; Newcomer, E.; Champion, M.; Ferris, C.; Orchard, D.: Web Services Architecture. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>, World Wide Web Consortium, 2004
- [Coll95] Collins, D.: Designing object oriented user interfaces. Benjamin/Cummings, Redwood City, 1995
- [DLPR02] Dangelmaier, W.; Lessing, H.; Pape, U.; Rüter, M.: Klassifikation von EAI-Systemen. In: HMD, Heft 225, dpunkt Verlag GmbH, 2002
- [Fran01] Frank, U.: Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze. In: Wirtschaftsinformatik, Band 43 / 3, Juni 2005, S. 283-293
- [FrJu02] Fraunholz, B; Jung, J.: Evaluation of Mobile Applications - Software-Technical and Human Aspects. In: Brown, A.; Remenyi, D. (eds): Proceedings of the Ninth European Conference on Information technology Evaluation. Université Paris-Dauphine, 15-16 Juli 2002
- [FrLa03] Frank, U.; van Laak, B.L.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 34, Koblenz 2003
- [Gart04] Gartner, Inc.: Worldwide Mobile Phone Sales Increased 21 Percent in 2003. http://www.gartner.com/press_releases/asset_62091_11.html, Letzter Abruf: 2006-01-02

- [Goer03] Görz, P.: Zusammenführen, was zusammen gehört : Die Integration verschiebt sich auf die Ebene der Geschäftsprozesse. <http://www.competence-site.de/eaisysteme.nst/ArtikelView>; Letzter Abruf: 2006-01-05
- [Goet04] Götz, J.: Entwicklung plattformunabhängiger Informationssysteme für mobile Geräte. Diplomarbeit, Universität Koblenz-Landau / Abt. Koblenz, FB Informatik, 2004
- [HaNe01] Hansen, H.R.; Neumann, G.: Wirtschaftsinformatik I : Grundlagen betrieblicher Informationsverarbeitung. 8. Aufl., Stuttgart : Lucius & Lucius, 2001
- [HFHH05] Hess, T., Figge, S., Hanekop, H., Hochstatter, I., Hogrefe, D., Kaspar, C.; Rauscher, B.; Richter, M.; Riedel, A.; Zibull, M.: Technische Möglichkeiten und Akzeptanz mobiler Anwendungen – Eine interdisziplinäre Betrachtung. In: Wirtschaftsinformatik, Band 47 / 1, Februar 2005, S. 6-16
- [HMNS03] Hansmann, U.; Merk, L.; Nicklous, M.S.; Stober, T.: Pervasive Computing, 2. Aufl., Berlin, Heidelberg : Springer, 2003
- [Jeff01] Jeffries, R.: What is Extreme Programming? <http://www.xprogramming.com/xpmag/whatisxp.htm>, Letzter Abruf: 2006-01-07
- [KrHa03] Kroll, M.; Haustein, S.: J2ME Developer's Guide : Java Anwendungen für mobile Geräte. München : Markt+Technik Verlag, 2003
- [Lueg02] Lueg, C.: On the Gap between Vision and Feasibility. In: Mattern, F., Naghshineh, M.: Pervasive Computing : First International Conference, Pervasive 2002. Springer, 2002
- [MüWi02] Müller-Wilken, S.: Mobile Geräte in verteilten Anwendungsumgebungen: Ein Integrationsansatz zwischen Abstraktion und Migration. Dissertation, Universität Hamburg, FB Informatik, 2002
- [MyRo92] Myers, B., Rosson, M.B.: Survey on User Interface Programming. In: Bauersfeld, P. (editor): Human Factors in Computing Systems (CHI'92 conference proceedings), ACM Press, New York, 1996, S.195 -202

- [RSF05] Rannenberg, K.; Schneider, I.; Figge, S.: Mobile Systeme und Anwendungen – Hammer sucht Nagel. In: Wirtschaftsinformatik, Band 47, Februar 2005, S. 1-2
- [ScKa02] Schryen, G.; Karla, J.: Elektronisches Papier - Displaytechnologie mit weitem Anwendungsspektrum. In: Wirtschaftsinformatik, Band 44, Dezember 2002, S. 567-574
- [Stan01] The Standish Group International, Inc.: Extreme Chaos. http://www.standishgroup.com/sample_research/, Letzter Abruf: 2006-01-06
- [StHa02] Stahlknecht, P.; Hasenkamp, U.: Einführung in die Wirtschaftsinformatik. 10. Aufl., Berlin, Heidelberg: Springer, 2002
- [Vare04] Varesi, A.: Was UMTS wirklich kann: Neue Anwendungen und Geschäftsmöglichkeiten. Erlangen: Publicis Corporate Publishing, 2004
- [Vkb05] VKB Inc.: technology. <http://www.vkb-tech.com/technology/default.php>, Letzter Abruf: 2006-01-04

Anwendungsszenario

Als Szenario soll hier der Mitarbeiter X des Außendienstes der Firma F heran gezogen werden. Die Firma F leistet weltweiten Support für diverse Industrieanlagen und wurde durch den Kunden U beauftragt eine defekte Anlage möglichst kurzfristig zu reparieren. Alle Mitarbeiter der Firma F sind mit mobilen Geräten ausgestattet, die ihre genaue Position auf der Welt an die Firma übermitteln. Das stationäre System S ermittelt den Mitarbeiter mit der benötigten Kompetenz und einer günstigen Nähe zum Kunden U und weist X einen Auftrag zu. Wurde ein Mitarbeiter beauftragt, kann sowohl Kunde U als auch Firma F den Status des Auftrags, und somit auch die Position von X auf der Welt, im Internet einsehen. Durch das mit System S integrierte mobile Gerät von Mitarbeiter X erlangt dieser alle kontextrelevanten Daten, wie z.B. detaillierte Kundendaten, die Historie der defekten Anlage des Kunden U und alle benötigten Schaltpläne übersichtlich auf seinen mobilen Bildschirm gelistet. Nach erfolgreicher Reparatur setzt der Vorstand des Kunden U ein Meeting an, in welchem ein Reparaturbericht vorgelegt werden soll. Mitarbeiter X sucht einen Platz in der Firma, um dort eine kurze Präsentation vorzubereiten. Sein Gerät erkennt automatisch den nächsten Drucker in der Nähe des aktuellen Standortes und druckt ein Handout für die Mitglieder des Meetings. In diesem sucht sich das mobile Gerät automatisch eine Schnittstelle zu dem Beamer im Raum und startet die Präsentation. Weiterhin schaltet sich das mobile System auf Grund des veränderten Kontextes auf lautlos, um die Präsentation nicht durch Geräusche zu stören. Die Mitglieder des Vorstandes sind ebenfalls mit mobilen Geräten ausgestattet, was eine Identifizierung der Personen durch eine automatische Verbindung (falls gestattet) der Geräte untereinander zulässt. Mitarbeiter X erhält auf seinem Bildschirm eine übersichtliche Darstellung der Personendaten und ist nun im Stande die ihm gegenüberstehenden Personen mit Namen anzusprechen. Außerdem werden die Daten an das System S übermittelt, um Kontaktadressen abzugleichen und geg. neu zu speichern, aber auch um Geburtstagsdaten für spätere automatisch unterstützte Kundenbeziehungspflege zu erhalten. Nach dem Meeting beendet Mitarbeiter X den Auftrag und bekommt vom System S seinen nächsten Auftrag zugewiesen. Die Navigationsdaten für die nächste Station werden von S direkt auf das mobile Gerät gespielt und eine Hotelbuchung wurde ebenfalls, nach Bestätigung durch Mitarbeiter X, durchgeführt.