



Universität Hamburg

Wolf-Gideon Bleek

# **Software-Infrastruktur**

Von analytischer  
Perspektive  
zu konstruktiver  
Orientierung

Hamburg University Press



Software-Infrastruktur

Von analytischer Perspektive zu konstruktiver Orientierung

Wolf-Gideon Bleek



Universität Hamburg



# Software-Infrastruktur

Von analytischer Perspektive zu konstruktiver Orientierung

Wolf-Gideon Bleek

Hamburg University Press ~ Hamburg

Dissertation zur Erlangung der Würde eines Doktors der Naturwissenschaften  
(Dr. rer. nat.) am Fachbereich Informatik der Universität Hamburg.

Vorgelegt von Wolf-Gideon Bleek aus Hamburg.

Betreuerin: Prof. Dr. Christiane Floyd

Gutachter: Prof. Dr.-Ing. Reinhard Keil-Slawik,  
Prof. Dr. Horst Oberquelle

Tag der Einreichung: 9. Dezember 2002

Tag der Disputation: 9. April 2003

#### Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen  
Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über  
<http://dnb.ddb.de> abrufbar.

ISBN 3-9808985-5-5

© Hamburg University Press, Hamburg 2004  
<http://hup.rrz.uni-hamburg.de>

Rechtsträger: Universität Hamburg

Produktion: Elbe-Werkstätten GmbH, Hamburg  
<http://www.ew-gmbh.de>

*Meinen Eltern*





# Inhaltsverzeichnis

|   |      |
|---|------|
| <b>Abbildungsverzeichnis</b>                                    | xiii |
| <b>Tabellenverzeichnis</b>                                      | xv   |
| <b>Zusammenfassung</b>  | xvii |
| <b>Abstract</b>   | xix  |
| <b>Geleitwort von Christiane Floyd</b>                          | xxi  |
| <b>Vorwort</b>  | xxv  |
| <b>1 Einleitung</b>   | 1    |
| 1.1 Motivation . . . . .  | 2    |
| 1.2 Einordnung . . . . .  | 4    |
| 1.3 Das Anliegen dieser Arbeit . . . . .                        | 7    |
| 1.4 Weiterer Aufbau der Arbeit . . . . .                        | 9    |
| 1.5 Allgemeine Bemerkungen . . . . .                            | 10   |
| <b>I Verständnis von Infrastruktur</b>                          | 13   |
| <b>2 Allgemeine Betrachtungen über große technische Systeme</b> | 15   |
| 2.1 Beispiele für große technische Systeme . . . . .            | 15   |
| 2.1.1 Das deutsche Eisenbahnsystem . . . . .                    | 16   |
| 2.1.2 Die deutschen Telefonnetze . . . . .                      | 18   |
| 2.1.3 Das amerikanische Luftfahrtsystem . . . . .               | 21   |
| 2.2 Große technische Systeme . . . . .                          | 24   |
| 2.2.1 Schlüsselbegriffe großer technischer Systeme . . .        | 27   |
| 2.2.2 Entstehungsprozesse großer technischer Systeme .          | 29   |
| 2.3 Zusammenfassung und Diskussion . . . . .                    | 31   |

|           |  |           |
|-----------|--|-----------|
| <b>3</b>  | <b>Analytische Auseinandersetzung mit Infrastruktur</b>                    | <b>35</b> |
| 3.1       | Die Verwendung des Wortes <i>Infrastruktur</i> . . . . .                   | 36        |
| 3.2       | Perspektiven auf Informations-Infrastrukturen . . . . .                    | 41        |
| 3.3       | Analytische Konzepte . . . . .   | 46        |
| 3.3.1     | Das Web-Modell . . . . .   | 48        |
| 3.3.2     | Die Infrastruktur-Kriterien von Star und Ruhleder . . . . .                | 52        |
| 3.3.3     | Die Akteur-Netzwerk-Theorie . . . . .                                      | 55        |
| 3.4       | Synthese der Analysekonzepte . . . . .                                     | 59        |
| <br>      |  |           |
| <b>II</b> | <b>Kritische Reflexion der Projekterfahrungen</b>                          | <b>61</b> |
| <br>      |  |           |
| <b>4</b>  | <b>Erfahrungen aus Infrastrukturprojekten</b>                              | <b>63</b> |
| 4.1       | Empirische Grundlagen . . . . .  | 63        |
| 4.1.1     | Das empirische Material . . . . .  | 64        |
| 4.1.2     | Methodischer Ansatz zur Auswertung . . . . .                               | 66        |
| 4.1.3     | Weiteres Vorgehen zur Auswertung . . . . .                                 | 68        |
| 4.2       | Das Infrastrukturprojekt <i>Internationale Frauenuniversität</i> . . . . . | 69        |
| 4.2.1     | Unerklärliches Verhalten: Der Server wird gebootet . . . . .               | 70        |
| 4.2.2     | Langsames Netz: Unnötiger Dateitransfer . . . . .                          | 72        |
| 4.2.3     | E-Mail-Aussperrung: Aktivierung der Firewall . . . . .                     | 75        |
| 4.2.4     | Chinesisches Windows nötig: Kulturelle Unterschiede . . . . .              | 77        |
| 4.3       | Das Infrastrukturprojekt <i>hamburg.de</i> . . . . .                       | 79        |
| 4.3.1     | Missbrauch von E-Mail-Adressen: Offener Versand . . . . .                  | 80        |
| 4.3.2     | Nachträgliche Einführung von Webmail: Wechselnde Anforderungen . . . . .   | 83        |
| 4.3.3     | Webserver zu langsam: Cache funktioniert nicht . . . . .                   | 84        |
| 4.3.4     | Preismodell konterkariert Implementierung: Mandantenfähigkeit . . . . .    | 86        |
| 4.4       | Die Lernplattform <i>CommSy</i> als Infrastrukturbestandteil . . . . .     | 87        |
| 4.4.1     | Reibungsverluste: Überschrittene Organisationsgrenzen . . . . .            | 89        |
| 4.4.2     | Langsame Diskussionsforen: Ineffiziente Datensammlung . . . . .            | 91        |
| 4.4.3     | Gegenläufige Entwicklungen: Zwei Projekte . . . . .                        | 92        |

|            |   |            |
|------------|---|------------|
| 4.4.4      | Neuer Kooperationspartner: Aufbau eines öffentlichen Servers . . . . .        | 94         |
| 4.4.5      | Update einspielen: Benutzer informieren . . . . .                             | 96         |
| 4.5        | Eigenschaften und Phänomene zusammengefasst . . . . .                         | 97         |
| <b>5</b>   | <b>Eigenschaften von Software-Infrastrukturen</b>                             | <b>103</b> |
| 5.1        | Ein Bewertungsrahmen für die Verdichtung und Einordnung der Empirie . . . . . | 103        |
| 5.2        | Software-Infrastruktur als großes technisches System . . .                    | 104        |
| 5.2.1      | Erfindung, Entwicklung und Innovation . . . . .                               | 104        |
| 5.2.2      | Übertragung . . . . .   | 105        |
| 5.2.3      | Wachstum, Wettstreit und Konsolidierung . . . . .                             | 106        |
| 5.2.4      | Beharrungsvermögen und radikale Änderungen . .                                | 107        |
| 5.2.5      | Konsequenzen für Software-Infrastrukturen . . . . .                           | 108        |
| 5.3        | Software-Infrastruktur als Artefakt . . . . .                                 | 109        |
| 5.3.1      | Standardisiertheit . . . . .  | 110        |
| 5.3.2      | Kopplungsfähigkeit . . . . .  | 112        |
| 5.3.3      | Geschichtlichkeit . . . . .   | 114        |
| 5.3.4      | Konsequenzen für Software-Infrastrukturen . . . . .                           | 116        |
| 5.4        | Software-Infrastruktur-Akteure . . . . .                                      | 117        |
| 5.4.1      | Akteure und Politik . . . . .   | 117        |
| 5.4.2      | Grenzüberschreitung zwischen Organisationen . .                               | 119        |
| 5.4.3      | Innensicht und Außensicht . . . . .   | 120        |
| 5.4.4      | Konsequenzen für Software-Infrastrukturen . . . . .                           | 123        |
| 5.5        | Software-Infrastruktur-Entwicklung als Prozess . . . . .                      | 123        |
| 5.5.1      | Ein vernetztes Prozessverständnis . . . . .                                   | 124        |
| 5.5.2      | Interferenzen . . . . .   | 127        |
| 5.6        | Zusammenfassung . . . . .   | 130        |
| <b>III</b> | <b>Infrastruktur als Orientierung für die Software-Entwicklung</b>            | <b>133</b> |
| <b>6</b>   | <b>Software-Infrastruktur-Entwicklung</b>                                     | <b>135</b> |
| 6.1        | Software-Infrastruktur – eine Einordnung . . . . .                            | 136        |
| 6.2        | Methodische Einbettung in die Softwaretechnik . . . . .                       | 138        |
| 6.2.1      | Evolutionäre Software-Entwicklung . . . . .                                   | 140        |

|           |  |     |
|-----------|--|-----|
| 6.2.2     | <i>Design for Opportunities und Continuous Engineering</i> . . . . .           | 140 |
| 6.2.3     | Agile Software-Entwicklungsprozesse . . . . .                                  | 141 |
| 6.3       | Software aus einer Infrastruktur-Perspektive . . . . .                         | 142 |
| 6.3.1     | Komplexität von Software-Infrastruktur . . . . .                               | 142 |
| 6.3.2     | Verschiedene Praxisgemeinschaften und ihre Infrastruktur-Wahrnehmung . . . . . | 144 |
| 6.3.3     | Arenen im Software-Entwicklungsprozess . . . . .                               | 147 |
| 6.4       | Die lokale und die globale Ebene bei der Infrastruktur-Entwicklung . . . . .   | 150 |
| 6.4.1     | Lokale Abgeschlossenheit herstellen . . . . .                                  | 153 |
| 6.4.2     | Globalen Bezug sicherstellen . . . . .   | 156 |
| 6.5       | Ein Entwicklungszyklus für Software-Infrastruktur . . . . .                    | 157 |
| 6.5.1     | Etablierung des Entwicklungsvorhabens . . . . .                                | 161 |
| 6.5.2     | Herstellung in der Entwickler-Arena . . . . .                                  | 163 |
| 6.5.3     | Systemversion . . . . .  | 164 |
| 6.5.4     | Einsatz in der Anwender-Arena . . . . .  | 165 |
| 6.5.5     | Kommunikation im Entwicklungsprozess . . . . .                                 | 166 |
| 6.6       | Interferenz-Management . . . . .   | 167 |
| 6.7       | Software-Entwicklung als Infrastruktur-Entwicklung . . . . .                   | 170 |
| <b>7</b>  | <b>Super infrastruttura</b> . . . . .  | 173 |
| 7.1       | Zusammenfassung . . . . .  | 174 |
| 7.2       | Kritische Bewertung des Erreichten . . . . .                                   | 176 |
| 7.3       | Infrastrukturen für Infrastrukturen . . . . .                                  | 179 |
| 7.4       | Offene Fragen – Ein Blick nach vorne . . . . .                                 | 180 |
| <b>IV</b> | <b>Anhang</b> . . . . .  | 183 |
| <b>A</b>  | <b>Empirieprojekt Internationale Frauenuniversität</b> . . . . .               | 185 |
| A.1       | Die Büro-Infrastruktur . . . . .   | 188 |
| A.2       | Errichten einer Infrastruktur für die Lehre . . . . .                          | 191 |
| A.2.1     | Örtliche Verteilung . . . . .  | 192 |
| A.2.2     | Sprache . . . . .  | 192 |
| A.2.3     | Technik . . . . .  | 194 |
| A.2.4     | Aufbau des Hamburger ifu-Servers . . . . .                                     | 199 |

|          |   |     |
|----------|---|-----|
| A.2.5    | Aufbau des Webservers . . . . .                                     | 201 |
| A.2.6    | E-Mail-Server und Server für Benutzerkonten . . .                   | 202 |
| A.2.7    | Zusammenspiel von Directory-Server und<br>lokalen Servern . . . . . | 205 |
| A.2.8    | Lernumgebung <i>CommSy</i> . . . . .                                | 206 |
| A.3      | Infrastruktur-Perspektiven . . . . .                                | 206 |
| A.3.1    | Sprache und Kultur . . . . .  | 208 |
| A.4      | Zusammenfassende Betrachtung . . . . .                              | 210 |
| A.4.1    | Rückwirkungen . . . . .   | 211 |
| A.4.2    | Wahrgenommene Problemfelder . . . . .                               | 212 |
| A.4.3    | Infrastrukturelle Herausforderungen . . . . .                       | 214 |
| <b>B</b> | <b>Empirieprojekt <i>hamburg.de</i></b>                             | 219 |
| B.1      | Organisatorische Rahmenbedingungen . . . . .                        | 222 |
| B.2      | Anforderungen an das Portal . . . . .                               | 226 |
| B.2.1    | Bestimmung des minimalen Kernsystems . . . . .                      | 229 |
| B.2.2    | Bestandteil Anmeldeprozess . . . . .                                | 231 |
| B.2.3    | Bestandteil Free-E-Mail-Plattform . . . . .                         | 234 |
| B.2.4    | Bestandteil Free-Homepage-Plattform . . . . .                       | 238 |
| B.2.5    | Bestandteil Kontenverwaltung . . . . .                              | 240 |
| B.2.6    | Allgemeine Probleme bei der Produktauswahl . . .                    | 242 |
| B.2.7    | Softwaretechnische Probleme . . . . .                               | 243 |
| B.3      | Verwendete Software-Komponenten . . . . .                           | 244 |
| B.3.1    | Content-Management-System und Webserver . . .                       | 244 |
| B.3.2    | Shopsystem . . . . .  | 245 |
| B.3.3    | Integration von Werbeformen . . . . .                               | 246 |
| B.4      | Zusammenfassende Betrachtung . . . . .                              | 247 |
| B.4.1    | Rückwirkungen . . . . .   | 247 |
| B.4.2    | Wahrgenommene Problemfelder . . . . .                               | 248 |
| B.4.3    | Infrastrukturelle Herausforderungen . . . . .                       | 248 |
| <b>C</b> | <b>Empirieprojekt <i>Community System</i></b>                       | 251 |
| C.1      | Funktionsumfang und Basisdefinition . . . . .                       | 253 |
| C.2      | Entwicklung des <i>CommSy</i> -Projektes . . . . .                  | 258 |
| C.2.1    | KnowNet . . . . .   | 258 |
| C.2.2    | Projektseminar im Wintersemester 1999/2000 . . .                    | 259 |

|                             |   |            |
|-----------------------------|---|------------|
| C.2.3                       | CommSy- <i>CommSy</i> – ein <i>CommSy</i> für die<br>Entwickler . . . . .                             | 262        |
| C.2.4                       | <i>Internationale Frauenuniversität</i> – der <i>CommSy</i> -<br>Einsatz bei der <i>ifu</i> . . . . . | 262        |
| C.2.5                       | <i>CommSy@uni.de</i> – ein öffentlicher <i>CommSy</i> -Server   | 263        |
| C.2.6                       | WissPro . . . . .   | 266        |
| C.3                         | Technische Infrastruktur . . . . .  | 267        |
| C.4                         | Zusammenfassende Betrachtung . . . . .  | 268        |
| C.4.1                       | Rückwirkungen . . . . .   | 269        |
| C.4.2                       | Wahrgenommene Problemfelder . . . . .   | 269        |
| C.4.3                       | Infrastrukturelle Herausforderungen . . . . .   | 270        |
| <b>Literaturverzeichnis</b> |   | <b>273</b> |

# Abbildungsverzeichnis

|    |  |     |
|----|--|-----|
| 1  | Kommunikation über Infrastruktur . . . . .   | 126 |
| 2  | Software-Entwicklungsprozess: Beispielhafte<br>Zusammensetzung der Interessengruppen-Arena . . . . .           | 147 |
| 3  | Software-Entwicklungsprozess: Beispielhafte<br>Zusammensetzung der Entwickler-Arena . . . . .                  | 148 |
| 4  | Software-Entwicklungsprozess: Beispielhafte<br>Zusammensetzung der Anwender-Arena . . . . .                    | 150 |
| 5  | Die Hülle eines Infrastruktur-Entwicklungsvorhabens mit<br>relevanten Grenzbereichen und Delegierten . . . . . | 155 |
| 6  | Schematische Darstellung des STEPS-Prozesses . . . . .   | 159 |
| 7  | Der Software-Entwicklungsprozess eines<br>Entwicklungsvorhabens . . . . .                                      | 160 |
| 8  | Exemplarische Entwicklungsvorhaben an einer Infrastruktur  | 172 |
| 9  | Ein Ausschnitt des Zeitverlaufs der <i>Internationalen<br/>Frauenuniversität</i> . . . . .                     | 188 |
| 10 | Lageplan der akquirierten Räumlichkeiten am<br>Hauptcampus mit technischer Ausstattung . . . . .               | 197 |
| 11 | Struktur der Verzeichnisse auf dem Fileserver . . . . .  | 200 |
| 12 | Verwendete Dienste nach Standorten . . . . .   | 205 |
| 13 | Informationsdokument Anmeldung . . . . .   | 216 |
| 14 | Letter of Consent für die geführten Interviews . . . . .   | 217 |
| 15 | Der Zeitverlauf des <i>hamburg.de</i> -Projektes . . . . .   | 221 |
| 16 | Hardware-Architektur von <i>hamburg.de</i> für das Webportal .   | 243 |
| 17 | Hardware-Architektur von <i>hamburg.de</i> für die Free-E-Mail   | 244 |
| 18 | Der Zeitverlauf des <i>CommSy</i> -Projektes . . . . .   | 252 |
| 19 | Die Einstiegsseite des <i>CommSy</i> . . . . .   | 256 |
| 20 | Software-Architektur des <i>CommSys</i> . . . . .  | 268 |





# Tabellenverzeichnis

|    |   |     |
|----|---|-----|
| 1  | Die Kerneigenschaften von Software-Infrastrukturen . . . .  | 101 |
| 2  | Die Projekte des Projektbereichs Information . . . . .  | 186 |
| 3  | Dienste des Fileservers . . . . .   | 199 |
| 4  | Spezielle Einstellungen des Netscape Communicators für<br>den Betrieb bei der <i>ifu</i> . . . . .                | 204 |
| 5  | Verwendete Software der <i>ifu</i> . . . . .  | 211 |
| 6  | Ausgewählte Standardprodukte für die Errichtung des<br>Webportals . . . . .                                       | 228 |
| 7  | Ergänzungsprodukte für das Webportal . . . . .  | 229 |
| 8  | Probleme beim Einsatz eines E-Mail-Standardproduktes .  | 231 |
| 9  | Fachliche Arbeitspakete . . . . .   | 264 |
| 10 | Technische Arbeitspakete . . . . .  | 265 |
| 11 | Technische Infrastruktur für das <i>Community System</i> . . . .  | 271 |
| 12 | Liste relevanter Projekte, die an der Universität Hamburg<br>mit Software-Unterstützung gehalten wurden . . . . . | 272 |



# Zusammenfassung

Ursprünglich in militärischen und wirtschaftlichen Zusammenhängen versteht man unter Infrastruktur den „Unterbau einer Organisation“; zunehmend wird der Begriff bei Informationssystemen verwendet, dort aber unterschiedlich gebraucht. In dieser Arbeit geht es um Software-Infrastruktur. Damit wird auf das entscheidende Merkmal von Infrastruktur Bezug genommen: dass sie als selbstverständliche Voraussetzung für alltägliche Arbeit gilt. Software-Anwendungen stehen in Beziehung zu anderer Software und bestehen selbst aus Software unterschiedlichen Ursprungs. Software ist vernetzt, wird in wechselnden Arbeitszusammenhängen eingesetzt, wirkt auf unterschiedliche Kontexte und steht unter dem jeweils begrenzten Einfluss von Benutzern, Betreibern, Entwicklern, Organisationen, Praxisgemeinschaften usw. In der Software-Entwicklung geht es heutzutage selten um die Entwicklung neuer Produkte, sondern vorwiegend um Weiterentwicklung, Parametrisierung und Konfiguration. Software-Infrastrukturentwicklung muss also mit bestehender Software umgehen und in komplexen Zusammenhängen unter Berücksichtigung einer stark angestiegenen Anzahl von Interessen und Möglichkeiten zur Wechselwirkung durchgeführt werden. Aus einer softwaretechnischen Perspektive stellt sich hier die Frage, wie Entwicklungsvorhaben abgegrenzt werden können, mit welchen theoretischen Konzepten Infrastruktur-Zusammenhänge angemessen erschlossen werden können und wie sich dies für ein konkretes Software-Entwicklungsvorhaben operationalisieren lässt.

Als theoretische Grundlagen aus der Literatur werden die Forschungen zu großen technischen Systemen, das Web-Modell, die Akteur-Netzwerk-Theorie und Infrastruktur-Kriterien zur spezifischen Theoriebildung herangezogen. Empirisches Material wird mit Methoden der qualitativen Forschung, Aktionsforschung und gegenstandsbegründeter Theorie ausgewertet. Drei Fallstudien (die *Internationale Frauenuniversität*, das Projekt *hamburg.de* und das Projekt *CommSy*), an denen der Autor als Entwickler maßgeblich beteiligt war, dienen als Grundlage für die Erarbeitung von

Infrastruktur-Eigenschaften, einem Verständnis von Software-Infrastrukturen und Ergänzungen für einen spezifischen Software-Entwicklungsprozess. Ausgehend von der verdichteten Erfahrung aus Projekten und vor dem Hintergrund der eingeführten Forschung wird ein existierender Methodenrahmen zu evolutionärer partizipativer Software-Entwicklung an die herausgearbeiteten Bedürfnisse angepasst.

Direktes Ergebnis aus den empirischen Fällen sind Eigenschaften der untersuchten Software-Infrastrukturen: Entwicklungsfähigkeit, Reproduzierbarkeit, Konfigurierbarkeit, Alltagstauglichkeit, Standardisiertheit und Aufbaubarkeit. Sie dienen zum Verständnis von Software-Infrastrukturen und als Ausgangspunkt für eine konstruktive Orientierung. Um eine Vielzahl unterschiedlicher Akteure koordiniert einzubeziehen und mit ihren Interessen umzugehen, reichen Rollen und Praxisgemeinschaften nicht aus. Dazu wird ein Arenenkonzept, in das sich Akteure einordnen, und ein Verständnis von parallel asynchron ablaufenden Entwicklungsvorhaben aufgebaut. Damit können verschiedene Bedürfnisse im Software-Entwicklungsprozess anhand unterschiedlicher Perspektiven, hier Außensicht und Innensicht genannt, abgeleitet werden.

Als ein besonderes Problem bei der Infrastruktur-Entwicklung wurden Interferenzen erkannt, die unerwünschte Wechselwirkungen zwischen unabhängigen asynchron stattfindenden Entwicklungsvorhaben bezeichnen. Das vorgeschlagene Interferenz-Management adressiert dies mittels Herstellen und Aufrechterhalten eines kontinuierlichen Überblicks, der Pflege von Kommunikation mit Beteiligten, das Identifizieren, Reduzieren, Behandeln, Ausgrenzen und Lindern von Interferenzen.

Der Kernpunkt des erweiterten evolutionären Software-Entwicklungsprozesses ist die kontinuierliche Erschließung der relevanten Bestandteile und Akteure der Infrastruktur vor dem Hintergrund des eigenen Entwicklungsvorhabens unter Berücksichtigung der verschiedenen Interessen. Dazu wird wiederholt zwischen lokaler Abgeschlossenheit, die Sicherheit im eigenen Entwicklungsvorhaben garantiert, und globalem Bezug, der die Verankerung der Entwicklung in der Infrastruktur-Gesamtheit sicherstellt, gewechselt.

# Abstract

The term infrastructure was originally used in military and economic contexts, where it meant the “basic structures and facilities necessary for an organization”; it is now being used increasingly for information systems in a variety of meanings. This text focuses on software infrastructure and makes the essential point that infrastructures are taken for granted as necessary conditions for everyday tasks. Software applications relate to other software and are themselves composed of software of different origins. Software is networked; it is used in changing work contexts, affects other contexts, and is to a limited extent controlled by users, providers, developers, organizations, communities of practice, and so on. Nowadays, software development rarely involves the development of new products from scratch, but the further development, customization, or configuration of existing products. Developing software infrastructures therefore needs to take into account existing software and handle a rising number of interests and possibilities of interference in complex contexts. What would a suitable development process look like? How does one handle interferences from concurrent development projects? From a software engineering perspective, the question arises how a line can be drawn between development projects. Which theoretical concepts can help to identify infrastructural interrelationships, and how can these be adapted to concrete development projects?

Theoretical foundations that are taken into account to construct a specific theory are research on large technical systems, the web model, actor-network-theory, and infrastructural criteria. Empirical data are examined with methods of qualitative research, action research and grounded theory. Three case studies (the International Women’s University, the hamburg.de project, and the CommSy project), in all of which the author actively participated as a developer, form the basis for developing characteristics of infrastructures in general, an understanding of software infrastructures, and provide supplementary insights for a specific software development pro-

cess. Taking the condensed experiences from the projects and the results of the above-mentioned research as a starting point, an existing framework for evolutionary participatory software development is taken and adjusted to the specific needs identified.

The following characteristics of the examined software infrastructures are derived directly from the case-studies: they must be capable of being further developed, configured, and reproduced; they further need to provide everyday viability, to be standardized and extendable. The characteristics help understand software infrastructures and provide a starting point for a constructive approach. Where a large number of different actors and their various interests have to be coordinated, the concepts of role and community of practice do not suffice. To deal with these shortcomings, the concept of arenas and an understanding of parallel asynchronous development activities are introduced. Actors arrange themselves in arenas, which are defined by a common point of reference and synchronous actions. This helps to identify disparate needs within the software development process from different perspectives, which we shall call the view from within and the view from without.

Interferences constitute a particular problem when developing infrastructures; they are negative reciprocal effects between independent asynchronous development activities. The proposed interference management addresses this by constructing and maintaining a continuous overview, taking care of the communication among the participants, and by identifying, reducing, treating, banishing and alleviating interferences.

The core feature of the extended evolutionary software development process is the continuous development of the relevant sections of and actors in the infrastructure. This has to happen in the light of one's own development activities and taking into account the various disparate interests. To ensure this, continual switching has to take place from local closure – which guarantees the security of the development activity – to global coverage, which anchors the development process within the infrastructure as a whole.

# Geleitwort

In den letzten Jahren ist die Nutzung von komplexen und vernetzten Software-Werkzeugen als Grundlage für individuelles und gemeinschaftliches Arbeiten, persönliche Kommunikation und Freizeitgestaltung weitgehend selbstverständlich geworden. Fast überall stehen technische Arbeitsumgebungen zur Verfügung, deren Dienstleistungen bedarfsgerecht angepasst und flexibel kombiniert werden können.

Wie bei vielen anderen technologischen Entwicklungen ist hier die Praxis der Theorie- und Methodenentwicklung in der Informatik voraus. Während alle Welt von informationstechnischer Infrastruktur spricht und hochwertige sozialwissenschaftliche Analysen zu diesem Phänomen bereits vorliegen, findet es in der Softwaretechnik noch wenig Beachtung. In dieser Arbeit befasst sich Wolf-Gideon Bleek mit IT-Infrastruktur auf Software-Ebene und verfolgt ein softwaretechnisches Interesse. Ausgehend von Erfahrungen in drei Projekten, die er wesentlich mitgestaltet hat, geht es ihm um ein vertieftes Verständnis von Software als Infrastruktur sowie um tragfähige methodische Vorgehensweisen.

Die Softwaretechnik wollte schon immer Ordnung ins Chaos bringen. Zuerst war sie gegen die Spaghettiprogrammierung selbst ernannter Künstler angetreten, um sie zu geordneter Zusammenarbeit bei der Herstellung eines Softwareproduktes zu bewegen. Hier ging es um Beherrschung der Gesamtsituation im Top-Down-Verfahren, was als Ausgangspunkt wohl definierte Probleme mit eindeutigen Lösungen voraussetzt. Dies widersprach jedoch der Realität sich wandelnder fachlicher Anforderungen und laufender technischer Innovation. Produkte wurden aufgrund von Erfahrung beim Einsatz in Versionen weiterentwickelt, dazu wurde die iterative oder evolutionäre Systementwicklung vorgeschlagen. Doch bald genügte auch das nicht, denn einzelne Produkte wurden in technische Umgebungen eingebettet, ganze Produktlandschaften integriert, Produkte auf Bibliotheken und Rahmenwerke abgestützt – und das bei raschem Technologiewechsel auf allen Ebenen. Die Notwendigkeit, mit veralteter Software umzugehen,

konkurrierte mit dem Anspruch, stets auf neuesten Stand zu sein, alle Tools zu vernetzen, sie von überall zu benutzen und Softwarekomponenten stets verfügbar zu haben. Re-Engineering war angesagt, um Alt-Software in Organisationen entweder entsorgen oder kapseln und veredeln zu können.

Software als Infrastruktur anzusehen, bedeutet einen Perspektivwechsel in mehrerer Hinsicht. Jetzt geht es nicht mehr um die Produkte selbst, sondern um ihre Wechselwirkung: die Abhängigkeiten und die Kopplungsfähigkeit von Softwarekomponenten. Auch reicht der Blick auf die Technik nicht aus: Zum einen wird diese von spezialisierten Fachkräften als Infrastruktur verfügbar gemacht. Zum anderen wird sie zur Infrastruktur nur im Nutzungskontext, wo sie häufig nicht scharf umrissenen Personengruppen, so genannten *communities of practice*, als selbstverständliche Voraussetzung für ihre Arbeit dient. Dabei unterscheiden und überschneiden sich die technischen Umgebungen, in denen die einzelnen Beteiligten arbeiten, so dass auch die Ränder der Infrastruktur unscharf werden.

Wie lassen sich Software-Infrastrukturen verstehen? Dazu werden hier drei Sichten ausgearbeitet:

- Als großes technisches System erscheint Software-Infrastruktur nicht nur in der Gegenwart, sondern in der geschichtlichen Dimension ihrer Entstehung. Während zu Beginn viele Freiheitsgrade existieren, entwickeln Systeme mit der Zeit ein Beharrungsvermögen, das sie auf eine Entwicklungslinie festlegt. So werden wir gewahr, dass zu unterschiedlichen Zeitpunkten auch verschiedene Eingriffsmöglichkeiten bestehen.
- Die Artefakt-Sicht stellt die Arrangements aus Softwareprodukten, die zunehmende Standardisierung sowie die Probleme von Verkopplung und Vernetzung in den Vordergrund. Gerade diese technische Problematik macht zugleich ihre kulturelle Bedingtheit deutlich, da Standards in Gemeinschaften entstehen, die eine jeweils individuell entwickelte Nutzungskultur pflegen. Bei der Durchsetzung von Standards werden Kulturen gestärkt oder gefährdet.
- Schließlich macht die Akteurs-Sicht deutlich, wie eine Vielzahl unterschiedlicher und teilweise voneinander unabhängiger Einzelpersonen, Gemeinschaften und Organisationen mit verschiedenen Interessenlagen auf Software-Infrastruktur Einfluss nehmen. Zwischen ver-



schiedenen Akteuren zu vermitteln, ist eine offenkundige Voraussetzung, um ihre Einfluss-Sphären voneinander abzugrenzen und Reibungsverluste zu vermeiden.

Welche Denkansätze hat die Softwaretechnik, um mit dieser Herausforderung umzugehen? Herr Bleek bietet dazu ein vernetztes Prozessverständnis an. Infrastrukturentwicklung vollzieht sich in vielen, häufig unvorhersagbaren Schritten, die von verschiedenen Akteuren ausgelöst werden, unterschiedliche Produkte betreffen, meist lokal durchgeführt werden, aber weitergehende Auswirkungen haben. Dabei werden einzelne Artefakte in verschiedenen Teilprozessen verfügbar gemacht, integriert und geändert. Diese Prozesse sind häufig unabhängig organisiert und daher nicht abgestimmt. So kann es zu Interferenzen kommen, bei denen ein Teilprozess den anderen stört.

Wie können Infrastrukturen zuverlässig in geordneten Verfahren verändert werden? Offenkundig ist der ursprüngliche softwaretechnische Anspruch eines Top-Down-Vorgehens aussichtslos, aber auch ein einfaches zyklisches Vorgehen genügt nicht. Wer im Chaos der Infrastrukturentwicklung bestehen will, muss vielmehr in der Lage sein, die Auswirkungen von lokalen Änderungen im Kontext abzuschätzen und Einzelvorhaben zuverlässig gegeneinander abzuschirmen. Dies führt zu einem Verständnis von ineinander verschränkten Entwicklungszyklen, bei denen Akteure in verschiedenen Arenen zusammen- oder gegeneinander arbeiten. Um handlungsfähig zu sein, gilt es zum einen, die lokale Abgeschlossenheit für ein Teilvorhaben herzustellen, aber zugleich den globalen Bezug sicherzustellen. So erweist sich das Management von Interferenzen als Angelpunkt einer gesicherten Vorgehensweise.

Die von Herrn Bleek vorgeschlagene Vorgehensweise fußt auf Erfahrungen in Projekten, in denen er Infrastrukturentwicklung als drängendes Problem hautnah erlebt hat. Um diese Erfahrungen für eine empirische Untersuchung zu verwerten und daraus aussagekräftige Ergebnisse zu gewinnen, hat er die gegenstandsbegründete Theorie als geeignete wissenschaftliche Methodik sinnvoll adaptiert. Ausgewählte Vorfälle aus den Projekten wurden anhand von Belegen rekonstruiert und dann mittels der von ihm gewählten analytischen Begriffe reflektiert. Die Ergebnisse der Arbeit sind richtungweisend, um die Entwicklung von Software-Infrastrukturen adäquat durchführen, dabei die Wünsche der Benutzer und Benutzerinnen

aufgreifen und zugleich technische Neuerungen, Anforderungen des Rechnerbetriebs und der Sicherheit berücksichtigen zu können.

Faszinierend ist, wie sich die Ordnungs- und Managementkonzepte der Softwaretechnik wandeln und dabei aktuelle Entwicklungen in der angewandten Erkenntnistheorie nachvollziehen. Vom außenstehenden Beobachter, der durch Formalisierung einen vorhersagbaren Konstruktionsprozess beherrscht, wurde der Projektleiter bald zum teilnehmenden Moderator in einem komplexen evolutionären Kommunikations- und Aushandlungsprozess. Nun wird er gar zum *bricoleur*, einem Bastler – eine interessante Parallele zur postmodernen Charakterisierung von Erkenntnisprozessen. Der ordnende softwaretechnische Anspruch beschränkt sich darauf, vor dem Hintergrund einer immer nur teilweise bekannten Geschichte und Vernetzung lokale Abgeschlossenheit von Änderungsprozessen zu erreichen, um sie von globalen Wechselwirkungen abschirmen zu können. Obwohl die Gesamtheit der Veränderungen und Wechselwirkungen das Verständnis jedes einzelnen Akteurs notwendig übersteigt, gelingt es so, lokal sinnvoll zu handeln und Entscheidungen aufgrund von Erfahrung revidieren zu können.

Der Prozess, in dem ich diese Arbeit betreuen konnte, war gekennzeichnet durch ungewöhnlich intensive Kommunikation, in der sich die begriffliche Basis, die Struktur des Textes und der wissenschaftliche Ertrag allmählich konkretisiert haben. Ich fühle mich durch diesen Prozess bereichert und freue mich über die hohe Qualität des Ergebnisses. Möge es vielen Lesern und Leserinnen zu Einsichten verhelfen, die ihnen einen geordneten Umgang mit Software-Infrastruktur erleichtern.

Hamburg, im November 2003

Christiane Floyd

# Vorwort

Why have the wings unless you're ment to fly?  
And tell me please, why have a mind if not to question  
why? — Alan and Merilyn Bergman

A formular for a happy but short life: Always yield to  
temptation. — unbekannter Autor

Entstanden ist diese Arbeit während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter am Arbeitsbereich Softwaretechnik des Fachbereichs Informatik der Universität Hamburg und als Mitarbeiter für HITEC e.V. Während meiner Beteiligung in verschiedenen Projekten konnte ich die Grundlagen für das nun vorliegende Ergebnis sammeln.

Es wäre mir nicht möglich gewesen, große Teile der Arbeit zu leisten, wenn ich nicht von vielen Menschen unterstützt worden wäre. Mein Dank gilt zuerst Bernd Pape, dem ich mein Dissertationsthema zu verdanken habe und mit dem ich wunderschöne wissenschaftliche Auseinandersetzungen geführt habe. Ich möchte meinen Arbeitskollegen Ralf Klischewski, Martti Jeenicke und Henning Wolf für die vielen Diskussionen, Anregungen und kritischen Fragen danken. Ich habe gerade den wissenschaftlichen Streit als sehr wertvoll empfunden und immer wieder genossen. Burkhard Ohlmann danke ich, weil er mir während der gesamten Entstehung meiner Dissertation viel Verständnis entgegen gebracht und Unterstützung vermittelt hat. Ich möchte Holm Wegner, meinem „Bruder im Geiste“, ganz besonders für die Unterstützung, die Anregungen und das gegenseitige Korrekturlesen danken. Die ehrliche und rückhaltlose Kritik verbunden mit den vielen Diskussionen über Gott und die Welt hat mir viel gegeben. Auf der technischen Seite darf ich nicht vergessen, Reinhard Zierke zu danken, der mir bei jeder noch so ausgefallenen Frage in Bezug auf Textverarbeitung mit L<sup>A</sup>T<sub>E</sub>X geholfen hat.

Für die Projekte möchte ich mich ebenfalls bedanken. Mein Dank gilt den vielen Studierenden, Kollegen und Professoren im *CommSy*-Projekt. Danken möchte ich Andreas Günter für sein Engagement im *hamburg.de*-Projekt und dem Team. Bei Beate Orłowski bedanke ich mich für ihre Diplomarbeit und die Diskussionen um Infrastruktur. Dorina Gumm gilt mein Dank für die detailverliebten Korrekturen und die Diskussionen um die Möglichkeiten der Deutschen Sprache.

Besonders danken möchte ich meiner Doktormutter Christiane Floyd, die sich in jede meiner noch so verworrenen Ideen hineinversetzt und intensiv Kritik gegeben hat. Darüber hinaus hat sie mein konstruktivistisches Weltbild und meine Sichtweise auf Software grundlegend geprägt. Neben den vielen Möglichkeiten, die sie mir eröffnet hat, hat sie mich in eines der Empirieprojekte „hineingezogen“, sodass wir einen Großteil der Erfahrungen gemeinsam machen konnten. Das Vertrauen, dass sie in mich gesetzt hat, freut mich sehr. Mein Dank gilt Horst Oberquelle und Reinhard Keil-Slawik für die Bereitschaft zur Übernahme der Zweitgutachten und für die wertvolle Kritik.

Letztendlich gebührt mein herzlichster Dank meinen Eltern, denn sie haben mich unnachlässig in meinem Tun unterstützt und gefördert. Ohne die von ihnen gegebenen Freiheiten und ihr uneingeschränktes Vertrauen hätte ich so manches Ziel nicht erreichen können. Ihnen ist diese Arbeit gewidmet.

Abschließend möchte ich mich bei allen bedanken, die mich in den letzten Monaten ertragen haben und hoffe, dass ich Ihnen eine Form der Unterstützung zurückgeben kann.

Hamburg, im Dezember 2003

Wolf-Gideon Bleek

# - 1 -

## Einleitung

Nel mezzo del cammin di nostra vita  
mi ritrovai per una selva oscura,  
ché la diritta via era smarrita.  
— Dante Alighieri

Dies ganze Buch ist eben Nichts als eine Lustbarkeit nach langer Entbehrung und Ohnmacht, das Frohlocken der wiederkehrenden Kraft, des neu erwachten Glaubens an ein Morgen und Übermorgen, des plötzlichen Gefühls und Vorgefühls von Zukunft, von nahen Abenteuern, von wieder offenen Meeren, von wieder erlaubten, wieder geglaubten Zielen. Und was lag nunmehr Alles hinter mir!  
— Friedrich Nietzsche, 1886, in der Vorrede zur „Fröhlichen Wissenschaft“

Softwaresysteme werden seit langem nicht mehr isoliert benutzt. Sie stehen in Beziehung zu anderer Software und bestehen selbst aus Software unterschiedlichen Ursprungs: Die Textverarbeitung steht in Bezug zum Fileserver, der E-Mail-Client zum Mailserver, der Browser zum Webserver. Ebenso integriert die Textverarbeitung eine Rechtschreibprüfung, der E-Mail-Client eine Verschlüsselungssoftware und der Browser stellt mithilfe von Plug-Ins Multimediainhalte dar. Diese Beziehungen hat die Softwaretechnik bisher über Schnittstellen, Protokolle und die damit verbundenen Standards ausgedrückt; alle drei werden in ihrem Einsatz kontinuierlich angepasst und weiterentwickelt. Auf der Ebene der Hardware und auf der Software-Ebene wurden diese angewendet und durch Strukturen und Technik ergänzt.

Über diese technischen und softwaretechnischen Beziehungen hinaus entwickeln sich beim Einsatz der Software auf der organisatorischen Ebene Zuständigkeiten, Dienstleistungen und Beziehungen und sind im jeweiligen Kontext in ständiger Entwicklung. Diese Vernetzung bringt es mit sich, dass Softwaresysteme, ihre organisatorische Einbettung und die mit

ihnen verbundene Hardwareinstallation zunehmend als Infrastruktur wahrgenommen werden. Die geänderte Wahrnehmung unterstreicht die wachsende Selbstverständlichkeit, mit der Hard- und Software verfügbar und bedienbar sind.

In diesem Zusammenhang stellt sich eine Vielzahl von Fragen. Neben dem eigentlichen Verständnis von Infrastruktur evoziert die Diskussion aus der Perspektive der Software-Entwicklung Fragen zum Charakter von Software-Infrastrukturen genauso wie nach systematischen Vorgehensweisen für deren (Weiter-)Entwicklung.

## 1.1 Motivation

Meine Motivation für das Schreiben dieser Arbeit liegt begründet in einer tiefen Unzufriedenheit in drei Gebieten. Zum Ersten ist das weit verbreitete Verständnis von Infrastruktur in der informatischen Diskussion geprägt von technischem Reduktionismus. Zum Zweiten waren meine Erfahrungen bei der Etablierung von Infrastrukturen in diversen Projekten mit Schwierigkeiten verbunden, ohne dass ich systematisch die Ursachen erkennen konnte. Und zum Dritten suche ich als Softwaretechniker nach einem Methodenrepertoire, das mir hilft, qualitativ hochwertige Software-Infrastrukturen zu konstruieren bzw. einen entsprechenden Entwicklungsprozess zu moderieren.

Der erste Punkt, technischer Reduktionismus, drückt sich z. B. durch vielfache Versuche aus, eine *IT-Infrastruktur* als die Zusammenstellung von Hard- und Software, Geräte, Daten und IT-bezogenen Personals zu beschreiben (Weill und Broadbent 1998).

Diese Beschreibung deckt sich nur wenig mit meinen Erfahrungen. IT-Infrastrukturen habe ich als Geflechte unterschiedlichster Bestandteile kennen gelernt. Neben Geräten, Programmen und für die Wartung verantwortlichen Personen konnte ich eine überwältigende Vielzahl von weiteren beteiligten Personen, Beziehungen, Organisationen, Vereinbarungen usw. erkennen. Gerade durch die verbindende Fähigkeit von Infrastruktur besitzt sie viel mehr den Charakter eines aktiven Gebildes denn lebloser Technik zusammen mit Wartungspersonal. Außerdem drückt die o. g. Beschreibung in keiner Weise aus, dass Infrastrukturen historische Entwicklungen verkörpern, sich aufeinander beziehen, voneinander abhängig werden und kulturelle Praxis verkörpern.

Ich konnte z. B. bei Hanseth lesen, dass ich mit meiner auf den subjektiven Erfahrungen fußenden Meinung nicht alleine stehe und es sich lohnt, weiter in dieser Richtung zu forschen:

“Infrastructures are heterogeneous phenomena. [...] First, information infrastructures are more than ‘pure’ technology; they are rather socio-technical networks. [...] Secondly, infrastructures are connected and interrelated, constituting ecologies of infrastructures. One infrastructure is composed of ecologies of (sub)infrastructures by building one infrastructure as a layer on top of another; linking logical related networks; integrating independent components, making them interdependent.”

(Hanseth 2000, S. 58-59)

Der zweite Punkt, Schwierigkeiten bei der Etablierung, ist für einen Softwaretechniker besonders unbefriedigend. Entgegen der durch ein rationales Weltbild geprägten Annahme von Ursache und Wirkung verhielten sich die jeweiligen Unternehmungen weder deterministisch noch waren im Vorhinein Ursachen für die Hindernisse und Rückschläge zu erkennen. Es sei aber sogleich angemerkt, dass es mir nicht um das Ausmerzen aller Schwierigkeiten geht, sondern vielmehr um das geeignete situative Reagieren.

So verweisen z. B. Ciborra und Hanseth darauf, dass fehlende Steuerung in einer Infrastruktur-Entwicklung dazu führen kann, dass diese in verschiedene Richtungen wächst, die größtenteils außerhalb der Kontrolle der verschiedenen Beteiligten sind (Ciborra und Hanseth 1998, S. 265).

Der dritte Punkt, Suche nach einem Methodenrepertoire, spiegelt die Reaktion auf die Frustration durch den zweiten Punkt wider. Nur ein methodisches Vorgehen kann die Möglichkeit für einen Erfolg eröffnen genauso wie es Erklärungen liefert. Darüber hinaus wird erkennbar, dass es der Softwaretechnik bisher an geeigneten Methoden in diesem Kontext mangelt.

Summerton stellt – aus der Diskussion um große technische Systeme kommend – „meine“ Frage, die ich versuchen möchte, aus der Konstruktionsperspektive der Softwaretechnik zu beantworten:

“How can ‘lessons learned’ be used to influence the development of systems in directions that are desirable from the standpoint of the citizens, users and enlightened operators mentioned [...]?” (Summerton 1994b, S. 17)

## 1.2 Einordnung

Infrastrukturen spielen eine wichtige Rolle in unserem täglichen Leben. Die Verkehrs-Infrastruktur wird auf dem Weg zur Arbeit als selbstverständlich angenommen; dies betrifft sowohl Autobahnen wie U- und S-Bahnen, Busse und andere Formen des öffentlichen Nahverkehrs. Telekommunikations-Infrastruktur wird mittlerweile als eine ebensolche Selbstverständlichkeit betrachtet wie Strom- oder Wasserversorgung. Ein Beispiel für Infrastrukturen ist die Lebensmittelversorgung: Hier kann man beobachten, wie während der Nacht Supermärkte gefüllt werden, beliefert durch Lastwagen, die weite Strecken zurückgelegt haben. Waren werden über Großmärkte umgeschlagen, bevor sie geerntet wurden auf Auktionen gehandelt, und über komplexe Regularien, Subventionen und Gesetze in Wirtschaftsräumen von Bauern bis Industrieunternehmen gleichermaßen erzeugt wie veredelt.

Alle diese Dienstleistungen und Versorgungen können als Infrastruktur eines Landes betrachtet werden, die durch die moderne Zivilisation entstanden sind. Fast niemand nimmt diese Dienstleistungen und den dahinterstehenden Versorgungsapparat als etwas Besonderes wahr, solange diese regelmäßig verfügbar sind. Ein Großteil der Dienstleistungen ist so preiswert geworden, dass niemand deren Verwendung in Frage stellen würde.

Infrastrukturen werden in vielen Fachgebieten thematisiert. Nachdem der Begriff ursprünglich militärisch geprägt wurde ist er z. B. auf ökonomische Systeme, medizinische Versorgung, Bildungssysteme und zivile Verkehrssysteme ausgedehnt worden:

“To buy a car is, in a real sense, to buy into complex road, energy supply, parts distribution, maintenance, registration, insurance, police and legal systems.”  
(Hannay and McGinn 1980, S. 28)

In letzter Zeit konnte beobachtet werden, wie der Infrastruktur-Begriff in die Informatik übernommen wurde. Unter dem Titel „Informations-Infrastruktur“ existiert bereits ein unscharfes Verständnis. Hinzu kommen politisch motivierte Interpretationen der Infrastruktur-Konzepte, die auf die Informationswissenschaften wirken:

“Underlying the concept of ‘information infrastructures’, ‘information highway’, ‘information society’ etc. is the common concept of the need to develop and diffuse broadband communication technologies. This infrastructure based on recent technological developments will allow rapid transmis-



sion of large quantities of information, which integrate data, video, text and voice traffic at low cost.” (OECD 1996, S. 5)

Gleichzeitig wächst die Bedeutung von Informations-Infrastrukturen auf der technischen, ökonomischen und gesellschaftlichen Ebene:

“The context in which information infrastructures is discussed clearly emphasises the economic aspects, in addition to the pure technology aspects. National reports are addressing the potential impact these technological innovations have on modern society and the economic and social benefits. In this sense, information infrastructures refer to the ability of new technologies to transform the way we work, play, learn and live. Information infrastructures become a requisite for building the information society.”

(OECD 1996, S. 5)

Die von der Informatik eingenommene Perspektive in Hinblick auf Infrastruktur ist keine einheitliche. Neben den ausdifferenzierten fachlichen Perspektiven (Netztechnologie, Software-Entwicklung, Informationsverarbeitung, Wissensmanagement, Künstliche Intelligenz) innerhalb der Informatik finden sich an den Randgebieten der Disziplin bzw. den Übergängen und Zusammenhängen mit anderen Fachgebieten (Organisationswissenschaften, Wirtschaftswissenschaften, Sozialwissenschaften etc.) sich stark unterscheidende Perspektiven auf Infrastruktur. Ziel kann es deshalb nicht sein, nach einem einheitlichen und verbindlichen Verständnis von Infrastruktur zu suchen, sondern vielmehr eine bewusste Untersuchung existierender Perspektiven anzufertigen und eine eigene Perspektive vor dem Hintergrund der selbstgewählten Anforderungen zu beschreiben.

Relevant aus dem Bereich der softwaretechnischen Methoden und Vorgehensweisen sind Konzepte des *Continuous Software Engineering* (CSE). Weber stellt in einem Einführungsdokument zu dem hierzu laufenden Forschungsprojekt „IT Infrastrukturen 2005 – Informations- und Kommunikations-Infrastrukturen als evolutionäre Systeme“ die Situation wie folgt dar:

- „• Die in vielen Bereichen der Wirtschaft über zehn, zwanzig oder sogar dreißig Jahre Schritt für Schritt entstandenen Informations- und Kommunikations-Infrastrukturen bestehen aus hunderten von Software-Systemen und Datenbeständen, deren Erhalt und Weiterentwicklung immer größere Schnittstellenprobleme und Erhaltungsaufwände mit sich bringen. Die Überwindung dieser Probleme erfordert die komplette Renovierung, d. h. die Überführung möglichst großer Teile der Altsysteme

in möglichst weitgehend standardisierte (Branchen-) Architekturen mit standardisierten Schnittstellen, die Reduktion bzw. Elimination von Datenredundanzen und die Flexibilisierung durch Vorkehrungen für die weitere Änderungsfähigkeit der Informations- und Kommunikations-Infrastrukturen. [...]

- Die in Informations- und Kommunikations-Infrastrukturen zusammengeführten Software-Systeme verschiedener Hersteller werden zu neuen Releases, Varianten und Versionen weiterentwickelt und müssen in ihren neuen Ausprägungen in die Infrastruktur eingepaßt werden können, ohne daß diese Infrastruktur in ihrer Gesamtheit komplett renoviert werden muß.“ (Weber 1999, S. 6-7)

Weber stellt ferner fest, dass die bestehenden Methoden und Verfahren der Softwaretechnik nicht mehr angewendet werden können. Er benennt Anforderungen, die neue Methoden und Verfahren erfüllen sollen, um für die Weiterentwicklung von Informations- und Kommunikations-Infrastrukturen genutzt zu werden:

„Die für das Continuous Engineering von Infrastrukturen notwendig werdenden Techniken lassen sich wie folgt charakterisieren:

- Systeme müssen sich leicht weiterentwickeln lassen, um neuen Benutzer- oder Betreiberanforderungen ohne großen Aufwand gerecht zu werden.
- Existierende Systeme müssen zu neuen Systemen zusammengefügt werden können, um zunächst isolierte Lösungen zu neuen integrierten Lösungen weiterzuentwickeln.
- Der Austausch von Komponenten soll einfach möglich sein, so daß die Neuentwicklung eines Systems schrittweise erfolgen und die Migration von einem alten System zu seinem neuen Äquivalent kontrolliert stattfinden kann. [...]
- Die Wiederverwendung existierender Systeme und Komponenten zum Aufbau anderer Systeme soll ohne größeren Aufwand möglich sein. Durch die Veränderung einer Software-Infrastruktur in einigen Komponenten oder Teilsystemen sollen die verbleibenden Teile nicht beeinträchtigt werden.
- Die Architektur großer Systeme soll durch die Stabilität der Standard-Plattform möglichst invariant gehalten werden. So kann die Entwicklung und Weiterentwicklung immer durch einen Orientierungsrahmen kontrolliert durchgeführt werden, der durch die invarianten Plattformen definiert wird.“ (Weber 1999, S. 6-8)

Wenngleich man nicht mit allen von ihm konstatierten Notwendigkeiten übereinstimmen muss, so liefert dies dennoch eine adäquate Darstellung und Einführung in den Problemkomplex.

In der Softwaretechnik besitzen wir bereits Vorgehensmodelle, die uns helfen, Anforderungen in einem zuerst unklaren Kontext systematisch zu ermitteln. Sie erlauben es ferner, mit sich ändernden Anforderungen während der Entwicklung umzugehen. Diese evolutionären und partizipativen Vorgehensmodelle haben sich in der anwendungsorientierten Software-Entwicklung etabliert (Schuler und Namioka 1993). Ein Vertreter dieser Modellkategorie ist STEPS (Floyd et al. 1989). Dieser Methodenrahmen wird in dieser Arbeit als Ausgangspunkt für einen auf Infrastrukturen angepassten Prozess genommen.

### **1.3 Das Anliegen dieser Arbeit**

Die Softwaretechnik besitzt ein Repertoire an adaptiven bzw. evolutionären Methoden und Vorgehensmodellen, die insbesondere auf die vollständige Neuentwicklung von Softwareprodukten zugeschnitten sind. Durch die Verschiebung zur Infrastruktur sind die Methoden und Vorgehensmodelle den aktuellen Anforderungen an Software-Entwicklung nicht mehr gewachsen. Wenig Berücksichtigung finden insbesondere die Einbindung bereits existierender Software-Installationen, die konsequente Weiterentwicklung in Benutzung befindlicher Software, die Einbeziehung existierender Voraussetzungen, durch unterschiedliche Alltagssituationen hervorgerufener Effekte und die Entwicklung von Software für verschiedene Praxisgemeinschaften.

Weber stellt fest, dass bisher für notwendige kontinuierliche Änderungen und Anpassungen keine Vorkehrungen getroffen wurden (Weber 1999, S. 15). Infrastrukturen wurden nicht weiterentwicklungsfähig gebaut. Es ist aber auf allen Ebenen notwendig (OECD 1996, S. 5).

Vor dem Hintergrund dieser Problemstellung verfolge ich mit der vorliegenden Arbeit das Ziel, einen Beitrag zum Verständnis des Infrastruktur-Begriffs (1) zu leisten (siehe Kapitel 2 und 3), Herausforderungen aus empirischen Fällen abzuleiten (siehe Kapitel 4) und diese zu Software-Infrastruktur-Eigenschaften (2) zu verdichten (siehe Kapitel 5) und ergänzende Elemente für Software-Infrastruktur-Entwicklungsprozesse (3) bereitzustellen (siehe Kapitel 6):

1. Die Forschung zum Infrastruktur-Begriff hat in den letzten Jahren erhebliche Verbesserungen erzielt. Die Charakteristika von Infrastrukturen konnten auf der analytischen Ebene herausgearbeitet werden. Gleichzeitig sind verschiedene Forschungsstränge mit ähnlichen Fragestellungen verfolgt worden. Es scheint nun notwendig, hier ein konsolidiertes Verständnis zu entwickeln.
2. Erst in den letzten Jahren ist die Bedeutung von Infrastrukturen für die Software-Entwicklung erkennbar geworden. Es ist angebracht, mit einem vorsichtigen Vorgehen in diesen Bereich einzutreten. Die Übertragung auf die Softwaretechnik werde ich auf empirische Fälle abstützen. Sie helfen, eine Einordnung im Bereich der Software-Entwicklung mit besonderer Berücksichtigung von Software-Entwicklungsprozessen durchzuführen.
3. Wie die Erkenntnisse zu den ersten beiden Punkten im Verlauf der Arbeit noch zeigen werden, kann ein Umgang mit Infrastrukturen am besten durch angemessene ergänzende Prozesselemente für bereits existierende Software-Entwicklungsprozesse erreicht werden. Die verbreitete Verwendung von Software als Infrastruktur hat Defizite in bestehenden Software-Entwicklungsprozessen offenbart. Diese Lücke werde ich entsprechend schließen.

Mein Anliegen ist also, Infrastruktur als eine konstruktive Orientierung für die Software-Entwicklung zu etablieren. *Konstruktiv* deshalb, weil ich den Begriff „Infrastruktur“ aus dem analytischen Gebrauch für Software in die Konstruktion überführen werde. Mit *Orientierung* verbinde ich eine Haltung, die weder dogmatisch Gewonnenes verwirft, noch den Anspruch erhebt, hiermit die endgültige Lösung gefunden zu haben. Schließlich schränke ich meinen Fokus auf die *Software-Entwicklung* ein, in dem Wissen, nur in diesem Tätigkeitsfeld vor dem Hintergrund meiner Ausbildung und Erfahrungen Aussagen machen zu können.

Jede Zeit hat ihre alle Gedanken durchziehenden Muster, so auch diese. In Themen aus den verschiedensten Bereichen wie Gesellschaft, Politik, Wirtschaft, Technik und Forschung habe ich immer ein Muster wiedergefunden: „Werden widersprüchlicher Dinge zusammengestellt, so ergeben sie in der Kombination etwas durchaus Sinnvolles“. In der Philosophie wird die Diskussion um dieses Phänomen unter dem Etikett der Postmoderne behandelt. Ich möchte in den Raum stellen, dass die Parallele für die In-

formatik die Infrastruktur sein kann. An entsprechenden Stellen werde ich wieder darauf hinweisen.

## 1.4 Weiterer Aufbau der Arbeit

Der vorliegende Text spiegelt meine Erfahrungen aus drei Praxisprojekten wider. Die Projekte sind: die Bereitstellung der technischen Infrastruktur für die *Internationale Frauenuniversität*, die Etablierung einer technischen Infrastruktur im Rahmen einer Plattformentwicklung für das Stadtinformationssystem *hamburg.de* sowie die Entwicklung der Lernplattform *CommSy*. Im Projekt *Internationale Frauenuniversität* habe ich die Infrastruktur konzipiert und in den Gegebenheiten der Universität Hamburg aufgebaut, bei *hamburg.de* habe ich maßgeblich die Gestaltung der Infrastruktur beraten und für das *CommSy* habe ich initiale Teile der Entwicklung übernommen und mehrere Einsätze begleitet. In allen drei Projekten stellten sich elementare Fragen im Zusammenhang einer Schaffung von infrastrukturellen Bestandteilen. Diese Parallelen gaben Anlass für die Betrachtungen in dieser Arbeit.

Die Arbeit gliedert sich in drei große Abschnitte. Im ersten Abschnitt „Verständnis von Infrastruktur“ stelle ich einen Bezug zu existierenden Diskussionen her. Dafür gebe ich einen knappen Überblick über große technische Systeme (Kapitel 2). Daran schließt sich eine analytische Auseinandersetzung mit Infrastruktur an (Kapitel 3), in der es neben der begrifflichen Bestimmung um eine Rezeption existierender analytischer Konzepte geht. Der Abschnitt endet mit einer Methodenbestimmung für die Fallstudien, indem eine Synthese der vorgestellten Methoden angeboten wird.

Nach einem Überblick über die Erhebung des empirischen Materials werden im zweiten Abschnitt „Kritische Reflexion der Projekterfahrungen“ (Kapitel 4) jeweils ausgewählte Vorfälle aus den Empirieprojekten präsentiert (Kapitel 4.2, 4.3, 4.4). Die zeitliche und ausführliche Beschreibung der Projekte findet sich im Anhang. Anschließend werden die Eigenschaften informationstechnischer Infrastrukturen zusammen mit dem Phänomen Interferenz auf dieser Grundlage bestimmt und verdichtet (Kapitel 5).

Im dritten Abschnitt „Infrastruktur als Orientierung für die Software-Entwicklung“ stehen Software-Infrastrukturen im Mittelpunkt. Er liefert eine konstruktive Wendung für die Softwaretechnik, mit der eine neue Sichtweise bei der Software-Entwicklung zur Verfügung gestellt werden soll

(Kapitel 6). Hier wird das Prozessmodell mit seinen Elementen vorgestellt, welches für die Entwicklung von Infrastrukturen eingesetzt werden kann (Kapitel 6.5). Der Abschluss gibt die Möglichkeit, offene Fragen zu formulieren, die für weitere Forschung Anlass geben und im Rahmen dieser Arbeit keine tiefere Beachtung finden konnten (Kapitel 7). Für den Umgang mit Infrastrukturen wird eine geeignete Perspektive und Orientierung der Softwaretechnik vorgeschlagen.

Der Anhang hält die ausführliche Beschreibungen der Empirieprojekte und ausgewählte Materialien aus den Projekten bereit.

## 1.5 Allgemeine Bemerkungen

Den Text habe ich nach den Regeln der neuen deutschen Rechtschreibung (Dudenredaktion 2000) geschrieben. In der Informatik wird eine Vielzahl von englischen Worten, Begriffen und Akronymen verwendet. Ich wollte beim Schreiben dieses Textes wann immer möglich deutsche Wörter verwenden, solange diese identisch gebraucht werden können. Allein die Vielzahl an Zitaten aus englischen Texten ließ sich aus wissenschaftlichen Gründen nicht vermeiden. Dazu kommt, dass mehr als ein Drittel der beschriebenen Projekte zu einem Teil in einem englischsprachigen Kontext stattgefunden hat. Werden englische Textstellen oder Aussagen zitiert, so arbeite ich mit englischen Anführungszeichen (“resource”) ansonsten verwende ich deutsche („Ressource“) (Dudenredaktion 2000, S. 90).

Da es mir bei den Zitaten auf die Urheber als Personen, die für die Aussage stehen, ankommt, habe ich Quellen nicht mit in den in Informatik-Publikationen üblichen Abkürzungen, sondern mit vollständigen Personennamen angegeben. Quellenverweise in Zitaten wurden mit in das vorliegende Literaturverzeichnis übernommen, um die Zitate vollständig zu nennen und es den Lesern zu erleichtern, diese Quellen ebenfalls weiterzuverfolgen. Soweit möglich, wurden die in Zitaten verwendeten Quellenverweise ebenfalls recherchiert.

Was im Folgenden gesagt wird, gilt natürlich für Wissenschaftler und Wissenschaftlerinnen, so wie es für Leser und Leserinnen gilt. Da das Deutsche keinen neutralen Ausdruck hat, der beide Geschlechter umfasst, beschränke ich mich darauf, von *Benutzer*, *Entwickler* oder *Betreiber* zu sprechen. Ich folge dabei grammatikalischem Brauch und bringe keinerlei Diskriminierung wegen des Geschlechts zum Ausdruck. Man kann natürlich

fragen, warum ich dann nicht immer von *Entwicklerin* etc. spreche. Dies deshalb, weil ich aus eigener Erinnerung und Erfahrung arbeite und die Darstellung dadurch mehr Unmittelbarkeit vermitteln kann (frei nach Eco 1998).





I

## **Verständnis von Infrastruktur**



## Allgemeine Betrachtungen über große technische Systeme

It was on the corner of the street that he noticed the first sign of something peculiar – a cat reading a map. For a second, Mr. Dursley didn't realize what he had seen – then he jerked his head around to look again. There was a tabby cat standing on the corner of Privet Drive, but there wasn't a map in sight. What could he have been thinking of? — J. K. Rowling

Wie noch in Kapitel 3 gezeigt werden wird, ist der Begriff *Infrastruktur* vergleichsweise spät entstanden und hatte zuerst eine eingeschränkte (militärische) Bedeutung, bevor er für eine ganze Klasse von Systemen verwendet wurde. Als die historischen Vorgänger von Infrastrukturen können große technische Systeme gesehen werden, die in den Sozialwissenschaften bereits eingehend untersucht wurden.

Beispiele für große technische Systeme helfen deren Charakter zu verstehen. Das deutsche Eisenbahnsystem dient als Beispiel für Wachstum und Konsolidierung. Aus Anlass der deutschen Wiedervereinigung können die Rollen von Kultur und Homogenität bei der Zusammenführung zweier Telefonsysteme nachvollzogen werden. Das amerikanische Luftfahrtsystem offenbart die Spannungsfelder, in denen große technische Systeme weiterentwickelt werden.

Die Diskussion um große technische Systeme offenbart eine ausgeprägte Analogie der Phänomene zu den bei Infrastrukturen beobachteten. Eine Auseinandersetzung mit den etablierten Begriffen großer technischer Systeme liefert deshalb hilfreiche Bezeichnungen für entsprechende Phänomene bei Infrastrukturen.

### 2.1 Beispiele für große technische Systeme

Ich habe ausgewählte Beispiele zusammengestellt, um einen Eindruck von den Eigenschaften und Prozessen großer technischer Systeme zu vermit-

teln. Im Anschluss daran werde ich deren Charakteristika systematisch zusammentragen. Das deutsche Eisenbahnsystem wurde als Beispiel für einen Wachstumsprozess ausgewählt, die deutschen Telefonnetze als Beispiel für den Zusammenschluss von großen technischen Systemen und das amerikanische Luftfahrtsystem dient als Beispiel für ein virtuelles großes technisches System.

### 2.1.1 Das deutsche Eisenbahnsystem

Die historische Entwicklung von Eisenbahnsystemen kann als Beispiel für ein Muster von Wachstumsprozessen großer technischer Systeme angesehen werden. G. Wolfgang Heinze und Heinrich H. Kill zeigen vier wiederkehrende Entwicklungsstufen an der Entstehung des deutschen Eisenbahnnetzes (Heinze und Kill 1988). Sie unterscheiden zwischen *Erfindung und isolierter Einführung*, *bedarfsorientierter Errichtung*, *versorgungsorientierter Erweiterung* und *instandhaltungsorientierter Auswahl*:

- “1. invention and isolated introduction (localized linkage),
2. demand-oriented construction (integration) – fulfilling only the needs of existing business centers,
3. supply-oriented extension (intensification) – supplying even the remote parts of the country motivated by the belief in equal access rights,
4. maintenance-oriented ‘cut-back’ (selection) – accepting efficiency as a basic principle and taking into account that different systems might complement each other. (In the case of road and freeway traffic the last stage has not yet been fully developed.)” (Heinze und Kill 1988, S. 105)

Zu Beginn des deutschen Eisenbahnsystems (*Erfindung und isolierte Einführung*) wurden im Bund deutscher Länder einzelne private Eisenbahnen errichtet (1815–1840). Zu dieser Zeit bestand kein öffentliches Interesse an einem Eisenbahnsystem. Das Ergebnis waren vereinzelt, unzusammenhängende Eisenbahntrassen. Jeweils mussten separate Unternehmen zum Aufbau gegründet, Lizenzen beantragt und die Wirtschaftlichkeit erreicht werden. Der ständige Ausbau dieser Netze und die Schaffung von Verbindungen zwischen bestehenden Strecken erschloss immer größere Bereiche des Landes. Die Auswahl der Strecken richtete sich nach dem Gesichtspunkt, zwischen welchen Orten bereits größere Verkehrsströme existierten (*bedarfsorientierte Errichtung*). Einher mit der technischen Zusammenfüh-

rung der einzelnen Eisenbahnstrecken ging gleichzeitig der politische Prozess der Vereinigung Deutschlands.

Wenngleich der dezentrale Aufbau des Systems wie ein Nachteil aussieht und eine effiziente Entwicklung auf anderem Wege – z. B. durch generalstabsmäßige Planung auf der Landkarte für ganz Deutschland – hätte erzielt werden können, so zeigte sich, dass hierdurch eine höhere Akzeptanz, bessere Anpassbarkeit und ein gutes Kosten-Leistungs-Verhältnis erreicht werden konnte.

Bereits während des Ausdehnungsprozesses (1841–1875) begann man die existierenden Strecken zu verbessern, sodass Züge mit höherer Geschwindigkeit fahren konnten. Parallele Gleise wurden gelegt und Weichen für die Lenkung der Züge in und zwischen den Bahnhöfen eingeführt. Aus politischen Gründen konnten manche Bahnlinien nur innerhalb eines Bundeslandes fahren, obwohl die kürzeste Strecke zwischen zwei Orten durch ein anderes Bundesland führte. Solche Verbindungen waren dadurch unattraktiv. Die Regierungen erkannten die Wichtigkeit der Eisenbahn und des Verkehrs und schlossen entsprechende Verträge ab, um die Eisenbahnen voranzubringen (*versorgungsorientierte Erweiterung*).

Bemerkenswert ist, dass vor Einführung der Eisenbahn die Kategorie der Distanz, in der die Menschen dachten, durch Fußmarsch, Pferd oder Kutsche bestimmt wurde. Mit der Eisenbahn waren in kürzerer Zeit dieselben Distanzen zu überwinden. Die Kosten einer Bahnfahrt wurden gegen die Kosten von Schuhsohlen, Übernachtung, Verpflegung und Verdienstausschlag gerechnet. In diesem Kontext war Bahnfahren weitaus günstiger und die Eisenbahn entwickelte sich zum Massentransportmittel.

In Bezug auf Gütertransport war das partikulare Bahnnetz nicht optimal ausgelegt. Für Menschen stellt ein Bahnhofswchsel nur eine Unbequemlichkeit dar, aber das Verladen von Gütern zwischen zwei Bahnhöfen machte deren Transport mit der Bahn unattraktiv. Es war notwendig, die Bahnnetze so zu verbinden, dass Verbindungen Ländergrenzen und Netzgrenzen überschreiten konnten, ohne dass ein Wechsel des Zuges erforderlich wurde. Die Integration der Eisenbahnsysteme wurde notwendig:

“[...] first regulations for standardized tariffs were discussed. In 1847 all railway organizations in Germany founded the ‘Verein Deutscher Eisenbahnverwaltungen’ in order to extend these standardization efforts, and before it was possible to travel from one side of Germany to the other in

the same train, one could make such a journey with several trains but one ticket.”  
(Heinze und Kill 1988, S. 126)

Allerdings folgte auf die Vereinheitlichung eine Stagnation und schließlich eine Degression (*instandhaltungsorientierte Auswahl*, seit 1920). Nach dem ersten Weltkrieg erkannte man, welche Strecken nicht profitabel arbeiteten, und legte diese schrittweise still. Diese Entwicklung wurde nach dem zweiten Weltkrieg fortgeführt und hält bis heute an.

Dieser Entwicklungsprozess kann abschließend als ein vierstufiger Beschleunigungsprozess dargestellt werden (Heinze und Kill 1988, S. 129-130):

1. Das existierende System erreicht seine Kapazitätsgrenze.
2. Eine neue Technik ist verfügbar.
3. Die Nachfrage nach Transportmöglichkeiten führt zu Verbesserungen der alten Technik und erhöht die Benutzung der neuen Technik.
4. Die zusätzliche Nachfrage ermöglicht die Einführung der neuen Technik.

Das Beispiel illustriert nicht nur ein typisches Muster für das Wachstum eines großen technischen Systems, sondern zeigt darüber hinaus die wichtige Rolle der politischen Akteure in dem System. Während sie erst durch Zurückhaltung „eingreifen“, nehmen sie sich nach und nach immer mehr Einfluss heraus. Ihr Wirken ist zu einem bestimmten Zeitpunkt notwendig (Vereinheitlichung, Zusammenschluss), damit aus vielen kleinen Netzen ein großes werden kann.

### **2.1.2 Die deutschen Telefonnetze**

Die fundamentale Änderung eines bestehenden großen technischen Systems und die entscheidende Rolle von Akteuren für die Zielsetzung des Veränderungsprozesses ist Thema einer Untersuchung zur Modernisierung des Telefonnetzes der DDR im Rahmen der Wiedervereinigung (von November 1989). Telefonnetze können als die größten technischen Systeme bezeichnet werden, weil sie nahezu jeden Ort dieser Welt erreichen und eine große Anzahl von zugehörigen Geräten besitzen. Tobias Robischon beschreibt die Aktivitäten nach dem Fall der Mauer, die vor dem Hintergrund

wechselnder politischer Ziele zur Modernisierung des DDR-Telefonnetzes geführt haben (Robischon 1994). Ausgangssituation war hierbei ein in der Bundesrepublik existierendes Telefonnetz – betrieben von der Deutschen Telekom – und ein davon unabhängig existierendes in der Deutschen Demokratischen Republik – betrieben von der Deutschen Post –, welches mit dem westdeutschen nur über wenige Kanäle verbunden war.<sup>1</sup>

Bevor überhaupt feststand, dass es eine Vereinigung der beiden deutschen Staaten geben würde, ging man davon aus, dass das (unabhängige) Telefonsystem der DDR zu modernisieren und stärker mit dem westdeutschen zu verknüpfen sei. Zu diesem Zeitpunkt existierten auf dem Gebiet der DDR mehrere Telefonnetze. Das öffentlich zugängliche Netz besaß eine schlechte Qualität und war nur für wenige verfügbar; deshalb hatten bereits einige Organisationen innerhalb des Landes davon unabhängige, bevorzugt ausgestattete Kommunikationsnetze aufgebaut. Auf Drängen der Politik, der Wirtschaft und der Privatbevölkerung sollte das allgemein verfügbare Telefonnetz dieselbe hohe Qualität wie in westlichen Ländern erreichen; die Parallelnetze sollten aufgegeben werden. Die Deutsche Post verhandelte mit der Politik und sah sich der Aufgabe gegenübergestellt, ein Netz nach aktuellem technischen Standard zu errichten. Entsprechende Kontakte zu ausstattenden Unternehmen wurden weltweit aufgenommen, um Technik einzukaufen. Als offenkundig wurde, dass die deutsche Einheit innerhalb eines Zeitraums von zwei Jahren realisiert werden würde, änderte dies den Bezugsrahmen der Beteiligten erheblich. Die Planer der Deutschen Post sahen die Chance, ihr Netz in Zusammenarbeit mit der Deutschen Telekom zu modernisieren. Ziel war es nun, das westdeutsche Modell zu kopieren.

Die Entscheidung der Deutschen Post, ihr Telefonsystem beginnend bei den Endanschlüssen zu modernisieren, wurde revidiert, als sich abzeichnete, dass im Rahmen der Wiedervereinigung die Deutsche Telekom für beide Netzbereiche gleichermaßen zuständig sein würde. Als neue Strategie wurde eine Top-down-Überlagerung propagiert, bei der zuerst die Langstreckenverbindungen ausgebaut, digitalisiert und in das westdeutsche Netz integriert werden sollten. Durch diesen Ansatz wurden Veränderungen am Netz zuerst nicht sichtbar.

Die von der Deutschen Telekom etablierten Kriterien wurden nun auf

---

1 Um Missverständnissen vorzubeugen: Vor 1990 hieß die Telefongesellschaft der Deutschen Demokratischen Republik „Deutsche Post“ und die der Bundesrepublik Deutschland „Deutsche Telekom“.

das Telefonsystem der DDR angewendet: hohe technische Standards im Bereich der Ortsvermittlungsstellen, breite Verbindungen zwischen Balungszentren und eine geringe Heterogenität bei den technischen Komponenten. Damit wurde von den Planern vermieden, dass bei der Erweiterung ein Flickenteppich von Technik entstand. Mit einem Flickenteppich wäre zwar ein schnellerer Erfolg erzielt worden, gleichzeitig wären aber höhere Kosten im Unterhalt entstanden. Die Planer der Deutschen Telekom argumentierten, dass provisorische Maßnahmen nicht zu ihrem Vorgehen passten (Robischon 1994, S. 131). Sie stünden im Gegensatz zur Kultur der Telekom und ihren Arbeitsgebräuchen, welche sich nach solidem und systematischem Entwickeln des Netzes auf der Basis von vorsichtig und langfristig ausgewählten Entscheidungen richteten. Dies wurde allerdings mit dem Nachteil erkaufte, dass die technische Entwicklung des Bereichs der DDR zuerst auf einer Ebene stattfand, die keinen direkten Nutzen für die Endkunden brachte, sondern zuerst dem Netzbetreiber nützte. Diese Form des Vorgehens wurden von den Endkunden intensiv kritisiert.

Durch die einheitliche Koordination beider Netze wurde der Ausbau auf dem Gebiet der DDR nach den im Westen praktizierten Prinzipien durchgeführt. Hiermit hatten die westdeutschen Lieferanten eine verbesserte Position, da sie sich nun nicht mehr mit anderen, ausländischen Herstellern im freien Wettbewerb um die Aufträge bemühen mussten. Die etablierten Partner auf Seiten der Geräteherstellung wurden für diese Aufgabe herangezogen.

Die grundlegende Wende in der Strategie der Deutschen Telekom bei der Modernisierung des DDR-Telefonsystems war im Rollenwechsel begründet. Zuerst – als noch zwei Telefongesellschaften beteiligt waren – wollte die Deutsche Telekom ihr Interesse verfolgen, indem sie die Zahl der Verbindungsleitungen zwischen beiden Gebieten erhöhte. Dies hätte allerdings das DDR-Telefonsystem überlastet, für das sie jetzt verantwortlich war. Die veränderte Struktur der beteiligten Organisationen hatte zur Folge, dass die Strategie angepasst werden musste. Kompatibilität zwischen beiden Systemen war ein wichtiges Ziel. Dies sollte zuerst auf der Ebene der Endanschlüsse erreicht werden. Die Kompatibilität wäre durch den Ausbau der Hauptverbindungen hergestellt worden, die gleichzeitig eine Basis für den späteren Ausbau der Endanschlüsse bildeten. Von Seiten der Deutschen Telekom wurde hingegen der Ausbau der Langstreckenverbindungen favorisiert.



Das entscheidende hier beobachtete Phänomen ist, dass die ursprünglich eingeschlagene Strategie, das DDR-Telefonnetz schrittweise an die neuen Anforderungen anzupassen, nach dem Wechsel der Akteure fundamental geändert wurde. Durch den Wegfall der Deutschen Post als verantwortlichem Akteur für die Modernisierung wurden die Ziele des neuen Akteurs wirksam. Dieser verfolgte das Interesse, seine bisher etablierten Wertvorstellungen aufrechtzuerhalten und dementsprechend zu übertragen. Als Konsequenz wurde dem System „von außen“ eine neue Gestalt gegeben, indem eine Kopie des westdeutschen Kernnetzes auf dem Gebiet installiert und dieses systematisch erweitert wurde. Andere Akteure – Wirtschaftsunternehmen und Politiker – konnten ihre Ziele in diesem Prozess nicht durchsetzen.

Dies steht in gewissem Gegensatz zu den Erfahrungen von Hughes, bei dem die system builder, die Mitarbeiter der Deutschen Post, und nicht – wie hier – die externen Firmenvertreter der Deutschen Telekom das System maßgeblich verändern (1987, S. 54-64). Die Änderungen am großen technischen System reflektieren somit auch die politischen Änderungen in der Konstellation der beteiligten Akteure.

Das Beispiel zeigt, dass Kooperation zu gegenseitiger technischer Verbindung führt und eine Fusionierung zu Homogenität auf der technischen Ebene. Gleichzeitig ist die Technik und ihre Zusammenstellung nicht nur ein Resultat der Umstände, sondern auch ein Hilfsmittel, um diese zu ändern, denn ein gemeinschaftliches und homogenes Netz hilft, die Existenz von zwei Netzbetreibern aufzuheben. Darüber hinaus begrenzt die Anzahl der verschiedenen Techniken gleichzeitig die Anzahl der beteiligten Akteure. Die in einem Zusammenhang etablierte Kultur wird zum Maßstab für das Vorgehen erhoben und bestimmt Entscheidungen über die Technik.

Die Struktur der technischen Basis des deutschen Telefonsystems kann aufgrund dieser Schilderungen als das Ergebnis der Interaktion einer Gruppe von Akteuren verstanden werden. Ihr Handeln und ihre Interessen entwickelten die technische Konfiguration und die Strategie, um diese konkrete Zusammenstellung von Technik einzuführen.

### **2.1.3 Das amerikanische Luftfahrtsystem**

Ein gutes Beispiel für die graduelle Entwicklung eines großen technischen Systems ist das amerikanische Luftverkehrssystem (USATS). Dessen Ent-

wicklung beschreibt Todd La Porte vom Beginn 1936 bis 1980 (La Porte 1988). Es ist bemerkenswert, dass dieses System im Gegensatz zu den anderen Beispielen neben seinen technischen Einrichtungen, den Standards und Vorschriften sowie dem Personal als Kern einen vollständig virtuellen Bestandteil hat, denn die mittlerweile etablierten Luftfahrtstraßen und Korridore manifestieren sich ausschließlich über Abmachungen aller Beteiligten.

La Porte schlägt vor, die technische und die soziale Perspektive bei der Untersuchung von großen technischen Systemen in der Art zu integrieren, dass sie als soziale Organisationen betrachtet werden. Das USATS kann als ein komplexer Bestandteil der amerikanischen Luftverkehrsindustrie gesehen werden, der allerdings weniger eng in technischer, physischer und administrativer Beziehung an seine Nachbarsysteme angekoppelt ist. Eine Besonderheit dieses Systems ist die Unabhängigkeit, mit der Akteure agieren können:

“Airplanes and pilots can operate with more autonomy than trains, telephone services and electrical power systems. The connective networks are much less dominated by physical objects – rails, wires and power grids. Finally, an air traffic (sub)system is largely a mental rather than a physical construct. [...] The system must be ‘seen in the head,’ a mental construct recognized by thousands of people (controllers, pilots, facilities managers) in order for ‘it’ to be operative.”  
(La Porte 1988, S. 223)

Die besonderen Anforderungen an das System (seine Aufgaben) sind, immer den Sicherheitsabstand zwischen Flugzeugen aufrechtzuerhalten, dafür ein integriertes Netz an helfenden und kontrollierenden Instanzen permanent auszubauen und dieses optimal zu betreiben.

Das Streben nach ständigem Ausbau dient als Beispiel für die Veränderung allgemeiner großer technischer Systeme. La Porte nennt die Spannungsfelder Sicherheit, Zuverlässigkeit und Effizienz. Diese wirken in den verschiedenen Interessengruppen auf zwei Ebenen: dem Bedürfnis, in das System zu investieren bzw. ein zu großes Investment zu vermeiden. Die Pilotenvereinigung steht den Managements der Fluglinien gegenüber, die fliegende Öffentlichkeit den Steuerzahlern, die Luftraumüberwacher ihrem Management und der Kongress dem Büro für Management und Budgetierung. Diese werden als Ansammlung sich gegenüberstehender Fürsprecher und Aufpasser charakterisiert. La Porte konstatiert: “Technical systems, then, are initially shaped by the operating requirements and social

properties of technical operations that are inherent in its technical design.” (La Porte 1988, S. 225)

Die Besonderheiten des Air Traffic Control System und die Unterschiede zu anderen großen technischen Systemen werden von La Porte auf vier Ebenen unterschieden: der Systemebene, dem Maß der nationalen Entwicklung, dem Grad der technischen Integration und dem Grad der Integration des Personals. Das USATS stellt sich auf der Systemebene als ein Sub-System dar, während allgemeine große technische Systeme als ganze (vollständige) Systeme betrachtet werden. Das Maß der nationalen Entwicklung ist verhältnismäßig schnell im Vergleich zu eher regional orientierten Entwicklungen. Der Grad der technischen Integration kann beim USATS eher als gering und losgelöst betrachtet werden, während bei anderen großen technischen Systemen eine hohe Kopplung im Vordergrund steht. Schließlich ist der Grad der Integration des Personals beim USATS durch die vollständige Einbindung der Operateure in den Betrieb hoch, während diese in anderen großen technischen Systemen eher zur Überwachung von Maschinen tätig sind.

Treibende Kraft für die kontinuierliche Anpassung und damit Veränderung des USATS war die ständig steigende Anzahl von Flügen und Flugzeugen. Daraus resultierte ein regelmäßiges Bedürfnis, die Kapazität des Systems zu erhöhen. Technologischer Fortschritt im Bereich der Luftfahrt wurde nicht zuletzt im militärischen Bereich gemacht. Die militärische Luftfahrt war von der zivilen vollständig getrennt. Nach gewisser Zeit führten die dort eingeführten Verbesserungen (z. B. Radar) zu einer Weiterentwicklung der zivilen Luftfahrt, d. h. dem USATS.

Da es sich beim USATS um ein System handelt, bei dem die Sicherheit menschlichen Lebens besonders wichtig ist, wurden strenge Maßstäbe angelegt und hohe Sicherheitsmaßnahmen ergriffen. Das Bedürfnis, Mehrdeutigkeiten oder Konflikte zu vermeiden, ist die treibende Kraft bei der Weiterentwicklung. Radar, Funkkommunikation zwischen Boden und Luft sowie Telekommunikation zwischen Standorten wurden auch deshalb eingeführt. Die neue Technik brachte mehr Sicherheit, erhöhte aber gleichzeitig die Abhängigkeit von der Technik. Da sie eine größere Kapazität des Systems erlaubte, wog ein Ausfall doppelt schwer. Entsprechend wurden Rückfallsysteme notwendig, die ohne oder nur mit geringem Technikeinsatz benutzt werden können. Dazu wurden die alten Techniken und Verfahren, die vor der Einführung der neuen Technik existierten, weiterhin

aufrechterhalten und ihre Nutzung regelmäßig trainiert. Durch die o. g. Verbesserungen konnte zwar die Kapazität erhöht werden, allerdings sank gleichzeitig die Autonomie der Beteiligten.

Mehr Flugverkehr führt zu noch mehr Kontrollen und Regelungen, um die Sicherheit weiterhin aufrechtzuerhalten. Beispiel dafür sind neu entstandene Richtlinien unter dem Namen "positive air traffic control", die für den Luftverkehr in der Höhe von 17.000 bis 35.000 Fuß verbindlich sind. Damit wurde der Raum für unkontrollierten Luftverkehr eingeschränkt und gleichzeitig die Kontrollarbeit vervielfacht. Die erhöhte Kontrolle und die zusätzlichen Regelungen erlaubten es im Gegenzug, eine wesentlich höhere Auslastung zu fahren und die Beteiligten dichter aneinander zu binden. Folge davon war ein insgesamt komplizierteres System. La Porte weist darauf hin, dass die Arbeitsbelastung für die Controller stetig steigt; diese haben sich gewerkschaftlich organisiert, um als ein Akteur im Kräfteverhältnis der Beteiligten mit einer Stimme zu sprechen (La Porte 1988, S. 236). Gleichzeitig wuchs der Druck auf andere Teilnehmer an der Luftfahrt (Sport- und Hobbyflieger), immer mehr Regeln und Leitlinien zu gehorchen und Ausbildungen zu absolvieren. Sie hatten weder das Geld noch die Zeit, im Wettlauf oder im Aufrüsten mitzuhalten (La Porte 1988, S. 237).

Das USATS ist ein gutes Beispiel für ein „virtuelles“ großes technisches System, da – wie bereits eingangs angemerkt – ein Großteil nur durch Vereinbarung zwischen den Beteiligten existiert. Gleichzeitig kann ein wiederholtes Pendeln zwischen Auslastung und Einführung kapazitätserhöhender Technik beobachtet werden. Besonderes Merkmal dieses großen technischen Systems sind die gut erkennbaren Spannungsfelder, in denen sich die Entwicklung abspielt. Es zeigt, welche Mannigfaltigkeit an Interessengruppen involviert ist und welche Kräfte diese freisetzen.

## 2.2 Große technische Systeme

Etwa Mitte der 80er Jahre wurde das Interesse der Sozialwissenschaften an technischen Systemen wiederbelebt. In dieser Zeit entstand eine spezielle Forschungsrichtung, die sich mit großen technischen Systemen (*Large Technical Systems, LTS*) auseinandersetzt (Hughes 1983, Bijker et al. 1987, Hughes 1987, Mayntz und Hughes 1988, Bijker und Law 1992, Summer-ton 1994a). In diese Forschungsrichtung ordnen sich die drei vorangestellten Beispiele ein, die als Vorläufer für eine Infrastrukturdiskussion gese-

hen werden können. Es erscheint deshalb sinnvoll, die Erkenntnisse über Eigenschaften großer technischer Systeme und deren Entstehungsprozesse genauer zu untersuchen.

Zu LTS werden neben den vorgestellten unter anderem Stromnetze und das Internet (Hughes 1983, Abbate 1994) gerechnet. Der Bereich wird von Bernward Joerges wie folgt abgegrenzt:

“LTS in the sense suggested would be integrated transport systems, telecommunication systems, water supply systems, some energy systems, military defense systems, urban integrated public works, etc. Manufacturing technologies, single utilities, office technologies, household technologies and so forth would not qualify as LTS in this sense, no matter at which level of aggregation – except of course to the extent that they form integral parts of such systems.”  
(Joerges 1988, S. 25)

Es stellt sich die allgemeinere Frage, was man unter einem großen technischen System zu verstehen hat. Von Thomas P. Hughes wird dafür eine indirekte Definition angeboten:

“An artifact [...] interacts with other artifacts, all of which contribute directly or through other components to the common system goal. If a component is removed from a system or if its characteristics change, the other artifacts in the system will alter characteristics accordingly. [...] the change in policy in one will bring changes in the policy of the other.”

(Hughes 1987, S. 51)

Bemerkenswert an dieser Definition eines großen technischen Systems ist die dem System attestierte Dynamik als Folge von Veränderung. Sie drückt eine gegenseitige Abhängigkeit aller beteiligten Bestandteile aus. Darüber hinaus wirken sich Veränderungen einzelner Bestandteile auf die übrigen derart aus, dass deren Zusammenstellung, Eigenschaften und Beziehungen neu austariert werden müssen. Dem System wird demnach ein Gleichgewicht unterstellt, das als Stabilität oder Verlässlichkeit interpretiert werden kann. In Hinblick auf die technischen Charakteristika großer technischer Systeme, die in Richtung (informations-)technischer Infrastrukturen lenken könnten, bietet Hughes keine konkreten Aussagen.

La Porte grenzt seine Sicht auf vernetzte LTS unter Berücksichtigung von speziellen Gegebenheiten davon ab. Die angebotene Definition ist wesentlich konkreter. Allerdings muss man den Entstehungshintergrund (US Air Traffic Control System, vergleiche Kapitel 2.1.3) berücksichtigen.

“Networked large technical systems are:

- Tightly coupled technically, with complex ‘imperative’ organization and management prompted by operating requirements designed into the system, i. e., unless operations are conducted in x, y ways, there are no benefits, maybe great harm can be imagined. (This is a kind of soft technical determinism: either do it my way or it won’t work and do good things for you.)
- Prone to the operational temptations of network systems, i. e., drive to achieve maximum coverage of infrastructure, and maximum internal activity or traffic within the network.
- Non-substituable service to the public, i. e. there are a few competing networks delivering the same service. (The more effective the existing system, the more likely its monopoly.)
- The objects of public anxiety about the possible widespread loss of capacity and interrupted service. (The more effective it is, the more likely the anxiety.)
- The source of alarm about the consequences of failures to users and outsiders of serious operating failures, e. g., mid-air collisions, nuclear power station disruptions, etc., and subsequent public expressions of fear and demands for assurances of reliable operations.”

(La Porte 1988, S. 240-241)

Große technische Systeme besitzen demnach auf der technischen Ebene einen großen Zusammenhalt, sie streben nach einer größtmöglichen Abdeckung, sie liefern an ihrem Ort nur schwer ersetzbare Dienste, es gibt die Befürchtung über einen Ausfall und man versucht stetig, ihre Zuverlässigkeit zu erhöhen. Diese angebotene Definition bietet eine gute Grundlage für große technische Systeme, in denen Informationen und Informationstechnologie eine wichtige Rolle spielen, da sie sich ebenfalls auf ein (virtuelles) großes technisches System bezieht, in dem der Umgang mit Informationen im Mittelpunkt steht.

Es erscheint deshalb sinnvoll, die Eigenschaften großer technischer Systeme und die zur Entstehung und zum Betrieb beitragenden Prozesse aufzuarbeiten, um sie als Grundlage für die Bildung eines Infrastruktur-Begriffs zu verwenden.

### 2.2.1 Schlüsselbegriffe großer technischer Systeme

Die Eigenschaften und Bestandteile großer technischer Systeme sind für die Infrastrukturdiskussion interessant, da sie in ähnlicher Form bei Infrastrukturen beobachtet werden können. Hughes nennt Schlüsselbegriffe, die zum Verständnis der Eigenschaften und Veränderungsdynamik großer technischer Systeme geeignet sind (Hughes 1983, Hughes 1987). Zu diesen Schlüsselbegriffen zählen Systemerbauer (*system builders*), Triebkraft (*momentum*), Auslastung (*load factor*) und zurückgefallene Teile (*reverse salients*) zusammen mit kritischen Problemen (*critical problems*).

- Mit Systemerbauern (*system builders*) werden diejenigen Personen bezeichnet, die als Erfinder, Ingenieur, Manager, Financier o.ä. ein System entwickeln, unterstützen oder erhalten:

“One of the primary characteristics of a system builder is the ability to construct or to force unity from diversity, centralization in the face of pluralism, and coherence from chaos. This construction often involves the destruction of alternative systems.”  
(Hughes 1987, S. 52)

- Die Triebkraft (*momentum*) bezeichnet zwei Eigenschaften: die Stabilität und die Bewegung, die ein existierendes System besitzt und es vor Richtungsänderungen in der Entwicklung bewahrt. Häufig wird dies missverstanden und dem technischen System eine Autonomie unterstellt.<sup>2</sup> Technische Systeme entwickeln eine eigene bewegende Kraft und eine eigene Trägheit. Diese Eigenschaften entstehen durch die versammelte „Masse“ an technischen und organisatorischen Bestandteilen und die damit verbundene Verkörperung von Entscheidungen. Dadurch besitzen sie eine Richtung (Ziele) und durch das sichtbare Wachstum eine angedeutete Geschwindigkeit. Die Beständigkeit von Artefakten in einem System legt die Metapher einer Flugbahn nahe, an der sich ein System entwickelt.
- Die Auslastung (*load factor*) bezieht sich auf die genutzte Verarbeitungskapazität eines technischen Systems. Damit ist einerseits die

---

<sup>2</sup> Ein anderer Grund für dieselbe Unterstellung ist häufig Unverständnis gegenüber einem großen technischen System und mangelnde Durchschaubarkeit von durchgeführten Änderungen während des Betriebs.

Kapazität des Systems definiert, andererseits kennzeichnet dies eine Grenze, die gegebenenfalls zu Problemen und zurückgefallenen Teilen (s. u.) führen kann.

“*Load factor* – the ratio of average system output to maximum output over a given period – is held, next to diversity of services and economic mix, to be a critical attribute of LTS which system builders and operators constantly try to improve. ‘Load factor is, probably, the major explanation for the growth of capital-intensive technological systems in capitalistic, interest-calculating societies.’ ” (Joerges 1988, S. 14)

Eine zu geringe Auslastung des Systems schlägt sich häufig in ökonomischen Problemen nieder und kann darauf hindeuten, dass es eine Kluft zwischen Bedürfnissen und Angebot gibt. Eine dauerhaft zu hohe Auslastung offenbart Grenzeigenschaften des Systems, die auf lange Sicht zu Veränderungen am System führen.

- Zurückgefallene Teile (*reverse salients*)<sup>3</sup> und kritische Probleme (*critical problems*) bezeichnen die auftretenden Probleme oder Missstände bei der Entwicklung eines Systems. Beide Phänomene sind entscheidend. Sie bündeln die zur Weiterentwicklung zur Verfügung stehende Energie und erlauben es daher, bei der Interpretation zwischen der Mikroebene (einzelnes Phänomen) und der Makroebene (Bündelung übergreifender Aktivitäten) Verbindungen zu ziehen. Gerade diese Probleme führen in der Entwicklung zur Ausrichtung zukünftiger Veränderungen, indem sie durch zunehmende Größe als eigene Akteure einem Bedürfnis Gehör verschaffen.

“As technical systems become larger, as other powerful interests and actor groups become involved in their expansion, and large organizations are built up for their gestation and drawn in from their environments, a number of phenomena and responses to them typically arise which Hughes subsumes under the term ‘*reverse salients*’. Reverse salients are technical or organizational anomalies resulting from uneven elaboration or evolution of a system. Progress on one front may produce backwardness elsewhere. Reverse salients require the identification and solution of underlying ‘critical

---

3 “Reserve salients” wird übersetzt als „rückwärtig gerichtete Vorsprünge“ oder „zurückgefallene Teile“ oder prägnant „Rückschläge“. Der von Hughes geprägte Schlüsselbegriff bezeichnet ein Teil eines großen technischen Systems. Leider bezeichnet das plakative deutsche Wort „Rückschlag“ ein Ereignis und kann somit nicht unter Einhaltung der notwendigen Präzision stellvertretend verwendet werden.



problems' and they drive continued inventive activity and system growth. In each phase of system development, the reverse salients 'elicit the emergence of a sequence of appropriate types of problem solvers, among them inventors, engineers, managers, financiers, and persons with experience in legislative and legal matters'. " (Joerges 1988, S. 13)

Diese vier Schlüsselbegriffe bieten eine gute Grundlage, um Phänomene von Infrastruktur-Entwicklungsprozessen in der Auswertung zu beschreiben und Eigenschaften zu benennen. Ergänzend dazu werden im Folgenden die Charakteristika für Entstehungsprozesse zusammengefasst.

### 2.2.2 Entstehungsprozesse großer technischer Systeme

In Bezug auf den Entwicklungsprozess eines großen technischen Systems unterscheidet Hughes folgende Phasen, die sich teilweise überlappen und nicht zwingend in nur dieser Reihenfolge auftreten können (Hughes 1987, S. 56): Erfindung, Erschließung, Innovation, Übertragung, Wachstum, Wettstreit und Konsolidierung.

- Die Phase der *Erfindung* wird in „konservativ“ oder „fundamental“ differenziert, wobei Organisationen anders als Einzelpersonen selten eine fundamentale Form der Erfindung bevorzugen. Konservative Erfindungen, treffender als Verbesserungen tituliert, treten auf, wenn entscheidende technische Probleme erkannt werden. Die Fachkenntnis der organisationseigenen Techniker hilft sie zu beheben. Etablierte Organisationen sind selten in der Lage, fundamentale Erfindungen zu machen, weswegen dieser Typus fast ausschließlich bei unabhängigen Experten anzutreffen ist.
- Im Bereich der *Erschließung* wird die soziale Konstruktion der Technik offenkundig: Technik wird durch alle beteiligten Personen gestaltet. Diese Personen eröffnen gleichzeitig ein Spannungsfeld, in dem sie handeln.
- In der Phase der *Innovation* ist die fundamentale Erfindung in eine Organisation eingebettet und wird von begleitenden konservativen Erfindungen auf verschiedenen Ebenen umsäumt. Hierzu zählt häufig ein passender Herstellungs- bzw. Betriebsprozess.

- Eine *Übertragung* von Technik findet immer dann statt, wenn dieselbe Technologie in anderen Zusammenhängen benötigt wird oder die äußeren Umstände sich signifikant geändert haben. Insofern ist das Konzept der *Adaption* eng mit dem der Übertragung verbunden. Insbesondere hat die Übertragung einer Technik immer Veränderungen im Bereich der beteiligten Zielorganisationen zur Folge.
- Die Phasen *Wachstum*, *Wettstreit* und *Konsolidierung* spielen nach Hughes eng zusammen. Wichtige Ursache für Wachstum ist das Streben nach Diversifizierung, Auslastung und einem ökonomischen Mix. Im Wachstum selbst treten Probleme auf, von denen einige als *zurückgefallene Teile* bezeichnet werden können (s. o.). Sie werden als Teile des Systems, die zurückgefallen oder in Bezug auf andere phasenverschoben sind, verstanden. Der Wettstreit findet mit anderen großen technischen Systemen bzw. deren betreibenden Organisationen statt, die dasselbe oder ein ähnliches Ziel verfolgen. Eine Konsolidierung tritt immer dann ein, wenn nach Wachstum und Wettstreit eine Form der Ruhe eingetreten ist und die Besinnung auf wesentliche Kernelemente einsetzt.

Damit steht ein Katalog ausdifferenzierter Phasen bereit, der sich zur genaueren Beschreibung der Entstehung großer technischer Systeme eignet. Allerdings weist Joerges darauf hin, dass durch Phasen die Dynamik eines großen technischen Systems nur unzureichend beschrieben werden kann. Darüber hinaus ist es unpassend, die beteiligten Ingenieure im Erklärungsmodell in einer unabhängigen Rolle zu sehen. Dies betrifft insbesondere die Einbettung von neuen Subsystemen in bereits bestehende LTS. Es wird nach einem neuen Erklärungsmodell gesucht, welches das unterschiedliche Verhalten besser beschreiben kann:

“The task is then to spell out what makes technical systems behave differently, as opposed to organizations, ideologies, moral or legal institutions, knowledge systems, etc. LTS would, by implication, be studied as particular types of such characteristically different social entities. Explanatory issues concerning LTS would be framed in terms of more general theories of technology as phenomenon *sui generis*. The difficulty with this second strategy, which I consider superior, is the dearth of elaborated theories of technology ‘as against’.”

(Joerges 1988, S. 16)

Hierzu werden im kommenden Kapitel 3.3 verschiedene Ansätze vorgestellt.

Darüber hinaus betreffen Entstehungsprozesse häufig die Zusammensetzung großer technischer Systeme aus kleineren Systemen. Existierende technische Systeme werden mit anderen verknüpft. Hierbei entstehen hierarchische Strukturen, die über Schnittstellen miteinander verknüpft werden:

“Inventors, organizers, and managers of technological systems mostly prefer hierarchy, so that systems over time tend toward a hierarchical structure. [...] Within the system the subsystems are linked by internal inputs and outputs, or what engineers call interfaces.” (Hughes 1987, S. 55)

Dieses Verhalten wird im Kontext von Informations-Infrastrukturen ebenfalls beobachtet und ist ein weiterer Hinweis für die konzeptionelle Nähe der beiden Gebiete.

## 2.3 Zusammenfassung und Diskussion

In diesem Kapitel habe ich neben drei Beispielen Schlüsselbegriffe für Eigenschaften und Prozesse großer technischer Systeme vorgestellt und diskutiert. Das Beispiel „Deutsches Eisenbahnsystem“ hat Muster für Wachstum, die wichtige Rolle politischer Akteure und die Notwendigkeit von Zusammenschluss gezeigt. Im Beispiel „Deutsche Telefonnetze“ wurden Kultur, Kompatibilität und Ziele der verschiedenen Akteure deutlich. Darüber hinaus konnte man erkennen, dass Kooperation zu Fusion und Verbindung zu Homogenität im technischen System führt. Das Beispiel „amerikanisches Luftfahrtssystem“ offenbarte die Veränderungen des Systems durch Kapazitätserhöhungen und Spannungsfelder, in denen die Weiterentwicklung stattfindet.

Die in der LTS-Diskussion vorgefundenen Argumente sind für die Softwaretechnik deshalb von zentralem Interesse, weil Entstehung, Aufbau und Weiterentwicklung von großen technischen Systemen evidente Parallelen zur Software-Entwicklung für Infrastrukturen besitzen. Gleichermäßen wird in beiden Gebieten (große technische Systeme und Software-Entwicklung für Infrastrukturen) für nur einen einzigen Einsatz entwickelt – erst bei einer Konsolidierung wird über mögliche Übertragungen, Expansion oder Zusammenschluss nachgedacht. Im Gegensatz dazu geht es z. B. bei Pro-

duktionsprozessen hauptsächlich darum, den immer gleichen Herstellungsprozess (z. B. den Bau eines Autos oder Flugzeugs) möglichst identisch zu reproduzieren und zu optimieren, d. h. ein Großteil der Energie wird in die Reproduzierbarkeit des Prozesses und seine Ausgestaltung im Sinne von Massenfertigung gelegt (vergleiche Sichtweise „Software als Produkt“, Kapitel 6.1). Die eine Wissenschaft optimiert Prozesse auf Wiederholbarkeit mit dem Ziel eines hohen Ausstoßes, die andere optimiert Prozesse in Hinblick auf deren Auswählbarkeit, Flexibilität und Anpassbarkeit im laufenden Prozess, um ein einzelnes Ergebnis zu erstellen.

Ein Defizit des LTS-Ansatzes ist, dass er bisher nur auf das Verstehen der Genese solcher Systeme ausgerichtet ist. Es werden Phänomene im Nachhinein beschrieben, charakterisiert und systematisiert. Für die Diskussion im Kontext von Software-Entwicklung ist es notwendig, diese Schlüsselbegriffe auf die Informatik zu übertragen und gleichzeitig angemessene Methoden und Prozesse für die Software-Entwicklung zu finden, die den Phänomenen im Verlauf der Entwicklung Rechnung tragen. Wie bereits festgestellt, ist ein deskriptives Phasenmodell dafür nicht geeignet, da es die Reihenfolge und das Vorgehen nur im Nachhinein beschreiben kann. Anstelle dessen muss nach einem begleitenden Prozess gesucht werden, welcher auf die individuellen und einmalig auftretenden Ereignisse eingehen kann.

Der Begriff des *großen technischen Systems* ist also für die Informatik auf den Begriff der *Informations-Infrastruktur* zu überführen. Die Verschiebung hin zur Infrastruktur wird bereits von Joerges vorgezeichnet, wenn er bei LTS von „selbstverständlich“ spricht:

“Similarly, as long as LTS function reliably and change only incrementally, they are largely ‘taken for granted’ by those depending on their products and services. Even partial insight into their workings is constituted mainly around failures.”  
(Joerges 1988, S. 26)

Mit dem Begriff Informations-Infrastruktur wird ein anderer Interpretationsschwerpunkt gesetzt (vergleiche Kapitel 5). Große technische Systeme haben vergleichbare Eigenschaften, wie sie im folgenden Kapitel über Infrastrukturen berichtet werden, z. B. dass sie erst bei Ausfall sichtbar werden:

“‘System accidents’, i. e. escalating failures resulting from unexpected and incalculable interactions of system components are related to such structural properties as ‘interactive complexity’ and ‘tight coupling’. These in

turn are related to intrasystem structural features such as degree of centralization or decentralization and, in an attempt to distinguish 'error inducing' systems (e. g. marine transport) from 'error avoiding' systems (e. g. airlines), to system environments." (Joerges 1988, S. 22)

Im kommenden Kapitel wird zuerst versucht, den Begriff *Infrastruktur* in einen klaren Kontext zu setzen und die damit assoziierte Bedeutung zu schärfen. Für den informatischen Bereich werden unterschiedliche Verwendungsfelder skizziert und in Relation zueinander gesetzt. Schließlich werden Methoden vorgestellt und synthetisiert, die bereits im Kontext von Infrastruktur angewendet wurden.



## Analytische Auseinandersetzung mit Infrastruktur

Functional analysis begins with fun.

— unbekannter Autor

Sokrates: Wenn dich jemand fragte, gibt es wohl einen falschen Glauben und einen wahren? Das würdest Du bejahen, denke ich?

Gorgias: Ja.

Sokrates: Wie nun? Auch eine falsche Erkenntnis und eine wahre?

Gorgias: Keineswegs.

Sokrates: Offenbar also ist nicht beides einerlei.

— Platon

Um den fachlichen Hintergrund des Themas – nämlich den Infrastruktur-Begriff – zu untersuchen, werden zuerst ausgewählte Quellen für diese Thematik herangezogen und bearbeitet. Ziel dieses Kapitels ist es ausdrücklich nicht, eine neue Sichtweise auf Infrastruktur einzuführen, neue Aspekte zu identifizieren oder neue Vorgehensweisen zu entwickeln. Vielmehr wird die vorgenommene Begriffsklärung einen Ausgangspunkt für die Rezeption der empirischen Beispiele liefern, indem eine begriffliche Basis für die analytische Auseinandersetzung mit technisch-informatischen Artefakten ausgebreitet wird, die im folgenden Kapitel benutzt werden kann. Eine darüber hinausgehende kritische und zugleich konstruktive Auseinandersetzung mit dem Begriff und den mit ihm assoziierten Methoden erfolgt in Kapitel 5.

Die analytische Auseinandersetzung mit Infrastrukturen in der Informatik kann aus verschiedenen Richtungen betrieben werden. Hierzu zählen nicht zuletzt die Gebiete *Künstliche Intelligenz*, *(Management-)Informationssysteme* sowie *verteilte und vernetzte Anwendungssysteme*. Darüber hinaus stellt sich eine Vielzahl von aus der *Softwaretechnik* kommenden Fragen an den (kommunikativen) Prozess der Software-Entwicklung, der

– begründet durch seine Tradition – auf Methoden und Werkzeuge der Entwicklung, Konfiguration und Anpassung softwaretechnischer Artefakte abzielt (Floyd und Züllighoven 1998, S. 763-764). In dieser Arbeit wird argumentiert, dass entgegen herkömmlicher Softwaretechnik, die sich mit vollständiger Entwicklung, Anpassung oder Konfiguration abgeschlossener Artefakte bzw. Systeme auseinander setzt, die Infrastrukturperspektive eine neue Qualität in die Ausformung des Entwicklungsprozesses einbringt.

Schwerpunkt dieses Kapitels ist ein erstes Verständnis von Infrastruktur, das im Folgenden für die Betrachtung dreier Fallstudien als Grundlage dienen soll. Die sprachliche Auseinandersetzung mit dem Wort *Infrastruktur* leitet zu einem allgemeinen Verständnis, welches in einer ersten Definition mündet. Daran anschließend lege ich zentrale Perspektiven aus der Informatik dar, um in einem Zwischenschritt *informationstechnische Infrastrukturen* zu definieren. Meine Definition einer *Software-Infrastruktur* schränkt diesen Begriff schließlich wieder ein. Darauf aufbauend bildet die Vorstellung von drei Analysetechniken, die im Zusammenhang mit Infrastrukturen bereits in der Informatik eingesetzt wurden, die Grundlage für die Synthese eines kombinierten Ansatzes.

### 3.1 Die Verwendung des Wortes *Infrastruktur*

Das Wort *Infrastruktur* wurde erst relativ spät gebräuchlich, d. h. im 19. Jahrhundert. Seine Verwendung durchzieht dabei die unterschiedlichsten Fachrichtungen; neben militärischen werden wirtschaftliche, politische, öffentliche und wissenschaftliche Einrichtungen mit Infrastruktur in Beziehung gesetzt. Zuerst sollen einige der vielfältigen Begriffsdefinitionen aus den relevanten Sprachen ausgebreitet werden, um danach auf deren analytische Anwendung im konkreten Fachgebiet einzugehen.

Während der Begriff *Infrastruktur* im Deutschen und Englischen zum Ende des 19. und zu Beginn des 20. Jahrhunderts noch nicht aufgeführt wurde (Grimm und Grimm 1877, S. 133; Hefes 1919, S. 367; Meyers Lexikon 1927, S. 437; Brockhaus 1931, S. 111), ist dieser seit etwa dem Ende des ersten Drittels des 20. Jahrhunderts in den Wörterbüchern und Lexika (Klappenbach und Steinitz 1969, S. 1952; Simpson und Weiner 1989, S. 950; Scholze-Stubenrecht und Mang 1999, S. 1937; Mish et al. 1998; Merriam-Webster 2001) zu finden (s. u.).



Die späte erstmalige Erwähnung überrascht, denn ein lateinischer Ursprung ist offenkundig, da beide Wortbestandteile aus dem Lateinischen stammen:

„*īnfra* (abl. sg. f. synk. < \**īn-fērād*; *infērūs*) I. adv. 1. pos. a) (räuml.) unten, unterhalb, darunter [*partes quae infra sunt*, [...] *infra esse* = zur Rechten liegen]; [...] ]

*strūctūrā*, ae f (eigtl. „Zusammenfügung, Ordnung“; *strūō*) 1. Bau; bsd. a) meton. Bauart [*antiqua*]; b) concr. Mauerwerk, Gemäuer [*parietum*]; [...] ] 2. (rhet. u. gramm. t.t.) Aufbau.“<sup>1</sup> (Pertsch 1983, S. 606/1136)

Dies kann aber für das Kompositum nicht weiter belegt werden – es wird in einschlägigen Wörterbüchern nicht verzeichnet. Jedoch bleibt es wichtig festzuhalten, dass der Bedeutungsgehalt des Wortes *Infrastruktur* aufgrund dieses Ursprungs charakterisiert wird durch das *Zugrundeliegen* und *Zur-Rechten-Liegen* eines Artefaktes oder *Aufbaus*. Damit wird unterstrichen, dass das bezeichnete Artefakt als Selbstverständlichkeit gebraucht werden soll.

Heutzutage wird der Begriff in der täglichen Praxis in vielerlei Kontexten relativ unspezifisch verwendet. Meist wird ein implizites Verständnis unterstellt; ein Blick auf die Anzahl der Ergebnisse der Suchmaschinen im Internet gibt einen ersten Eindruck von der Popularität der Verwendung des Wortes *Infrastruktur* in Online-Dokumenten (mehrere hunderttausend Treffer). Es zeigt, dass sowohl der Begriff *Infrastruktur* (zusammen mit seinem englischen Pendant) als auch die eher informatisch motivierte Begriffsbildung *Informations-Infrastruktur* (s. u.) mittlerweile allgemein verwendet wird.

Sucht man nach einer genaueren Definition von *Infrastruktur*, geben Lexika und Wörterbücher einen guten ersten Hinweis. *Infrastruktur* wird dort sowohl als Teil des wirtschaftlichen Lebens als auch der militärischen Organisation gesehen. Im Folgenden wird der Eintrag des Brockhaus wiedergegeben:

„*Infrastruktur*, häufig synonyme Bez. für den Unterbau einer Organisation. Urspr. in der Fachsprache der NATO verwendeter Begriff für ortsfeste Anlagen und Einrichtungen, die den Streitkräften dienen (z. B. Kasernen,

1 Abkürzungserklärung (ebd.): abl. – Ablativ; sg. – Singular; synk. – synkopiert; adv. – Adverb, adverbial; pos. – Positiv; bsd. – besonders; meton. – metonymisch; concr. – Konkretum, konkret; t.t. – terminus technicus.

Flugplätze, Brücken). Erst seit Beginn der 1960er Jahre ist I. ein in den Wirtschaftswissenschaften und verwandten Bereichen (z. B. Raumplanung, Sozialpolitik) gebräuchl. Begriff für die Gesamtheit der Anlagen, Einrichtungen und Gegebenheiten, die den Wirtschaftseinheiten als Grundlage ihrer Aktivitäten vorgegeben sind. In sehr weitem Sinn gehören dazu Wirtschaftsordnung, rechtliche Ordnung, Stand und Entwicklung der sozialen Sicherung, von Bildung und Wiss., Raumordnung, Verkehrserschließung u. ä. Es wird unterschieden zw. *personeller I.* (die v. a. geistigen, unternehmer., handwerk. Fähigkeiten der am Wirtschaftsprozess beteiligten Personen, → Humankapital), *institutioneller I.* (gesellschaftl. Normen, Einrichtungen und Verfahrensweisen) sowie *materieller I.* („social overhead capital“). Diese I. i. e. S. umfaßt die Gesamtheit der staatl. und privaten Einrichtungen, die für eine ausreichende Daseinsvorsorge und die wirtschaftl. Entwicklung eines Raumes erforderlich sind, und gliedert sich in *technische I.* (Einrichtungen des Verkehrs- und Nachrichtenwesens, der Energie- und Wasserversorgung sowie der Entsorgung von Abfallstoffen) und *soziale I.* (Kindergärten, Schulen, Sport- und Erholungsanlagen, Krankenhäuser, Alten- und Pflegeheime, auch kulturelle Einrichtungen). Spezif. Eigenschaften der I.-Investitionen sind u. a. hoher Anteil externer Effekte und polit. Entscheidungskriterien, schwierige Produktivitätsmessung, weshalb die I. auch als öffentl. Gut angesehen wird. Die Schaffung gleichwertiger Lebensbedingungen für die Bevölkerung und guter Standortbedingungen für Unternehmen können als oberste Ziele der staatl. I.-Politik angesehen werden, deren Maßnahmen im einzelnen u. a. den Bereichen Raumordnung und Regionalpolitik, Verkehrs-, Umwelt- und Bildungspolitik, Kommunalpolitik (für die Gemeinden als Träger der I.-Politik) sowie – auch bezogen auf Entwicklungsländer – Wachstums- und Entwicklungspolitik zugeordnet werden können. Theorie u. Praxis der I.-Politik, hg. v. R. Jochimsen u. a. (1970); I., hg. v. U. E. Simonis (1977).“ (Brockhaus 1989, S. 501)

Die Wörterbücher des Dudenverlages liefern eine weniger ausführliche, aber ebenso fokussierte Definition:

„In|fra|struk|tur die:

1. *notwendiger wirtschaftlicher u. organisatorischer Unterbau als Voraussetzung für die Versorgung u. die Nutzung eines bestimmten Gebiets, für die gesamte Wirtschaft eines Landes:* Aus dem Schema heraus fallen auch die Praxiskliniken, in denen eigenständige Ärzte die für das ambulante Operieren notwendige I. gemeinsam finanzieren und nutzen (Woche 4.4.97, 21); Bozen verfüge über eine ‚weit ausgreifende touristische Infrastruktur‘ (Spiegel 11. 1984, 78).

## 2. *Gesamtheit militärischer Anlagen.*“

(Scholze-Stubenrecht und Mang 1999, S. 1937)

Die Definitionen verwenden Begriffe wie *Gegebenheiten*, *Unterbau* oder *Anlagen* und bleiben damit in der Anwendung des Wortes relativ unspezifisch. Gleichwohl lässt sich erkennen, dass eine Bedeutungsverschiebung vom wirtschaftlichen und militärischen Zusammenhang hin zu einer breiteren Übertragung des Wortes auf allgemeine gesellschaftliche Einrichtungen (hier: Praxiskliniken, Tourismus) stattgefunden hat.

Ein Blick in die englischen Wörterbücher zum Begriff *infrastructure* zeigt ergänzend folgendes Bild:

“infrastructure the basic structures and facilities necessary for a country or an organisation to function efficiently, e. g. buildings, transport, water and energy resources, and administrative systems: *investment in infrastructure* ◦ *improve the country's commercial/financial infrastructure.*”

(Crowther 1995)

Der relevante Eintrag aus dem *Oxford English Dictionary* zeigt neben der Begriffsdefinition zusätzlich anhand von lesenswerten von der Redaktion ausgewählten Verwendungsbeispielen, wie unterschiedlich das Wort über die Jahre hin gebraucht wurde; es umfasst die Bereiche Bauwerke, Verkehrswege, abstrakte Beziehungsgeflechte, militärische Anlagen und Vorhaben, musikalische Maße, gesellschaftliche Grundlagen sowie Bewertungen von Sprachen:

“infrastructure. [Fr. (1875 in Robert), f. *infra-* + *structure sb.*] A collective term for the subordinate parts of an undertaking; substructure, foundation; *spec.* the permanent installations forming a basis for military operations, as airfields, naval bases, training establishments, etc.

1927 Chamber's Frnl. 14 May 374/2 The tunnels, bridges, culverts, and 'infrastructure' work generally of the Ax to Bourg-Madame line have been completed. 1950 W. S. Churchill in Hansard Commons CDLXXVI. 2145 In this debate we have had the usual jargon about 'the infrastructure of a supra-national authority'. 1951 European Rev. Oct. 2/1 This new term 'infrastructure' ... denotes fixed military facilities such as airfields, base installations and transport systems. 1956 D. Noakes tr. Hodeir's Jazz 197 What I call the infrastructure is the regularly produced two- or four-beat meter (2/2 or 4/4 measure) that characterizes any jazz performance. 1957 T. Kilmartin tr. Aron's Opium of Intellectuals iv. 133 Thirty years ago, the

dominant school of thought in the Soviet Union undertook ... the task of analysing the infrastructure of society. 1960 Times 9 Dec. 14/2 Part of the Nato infrastructure programme. 1971 Inside Kenia Today Mar. 15/1 A.I.D. assistance will be focused on Vihiga Division and will ... upgrade the infrastructure of roads and other social services. 1971 J. Spencer Eng. Lang. W. Afr. 31 A very complex infrastructure of scores of vernacular languages.”  
(Simpson und Weiner 1989, S. 950)

Das Merriam-Webster Dictionary bietet darüber hinaus eine knappe, aber dennoch nützliche Definition an:

“Main Entry: in-fra-struc-ture

1. the underlying foundation or basic framework (as of a system or organization)
2. the permanent installations required for military purposes
3. the system of public works of a country, state, or region; also: *the resources (as personnel, buildings, or equipment) required for an activity.*”

(Mish et al. 1998, Merriam-Webster 2001, Hervorhebung eingefügt)

Der hervorgehobene Teil des Auszugs ist die weiteste Definition für den Infrastruktur-Begriff. Sie schließt neben den Gütern und Anlagen auch Personen (wie bereits Dudenredaktion 1990, s. o., auf *Arbeitskräfte* verweist) mit ein und stellt einen Zweckbezug her (“required for an activity”).

Als Arbeitsdefinition für *Infrastruktur* soll im Folgenden genommen werden:

#### *Definition 1 (Infrastruktur)*

Eine Infrastruktur ist eine einer Gruppe von Personen zur Verfügung stehende Zusammenstellung von Personen, Organisationen, Gerätschaften, Installationen, Regelungen, Standards und damit verbundenen Dienstleistungen, die zur Umsetzung von Aktivitäten langlebig zur Verfügung steht und als selbstverständlich betrachtet werden kann.

Unter einer Infrastruktur kann man demnach z. B. die Wasserversorgung eines Landes verstehen, wenn man darunter neben dem Leitungsnetz und dem Wasser selbst die damit verbundenen Standards über Wasserdruck, Leitungsquerschnitt, die regelmäßigen Messungen der Wasserqualität, das zugeordnete Vertriebssystem, die Gesetze und Regularien und die damit verbundenen Tätigkeiten zur Aufrechterhaltung der Versorgung fasst.

Zusammenfassend lässt sich also feststellen, dass ein allgemeines Verständnis von einer „unterhalb liegenden Struktur“ schon seit langer Zeit existiert, aber erst zum Ende des 19. und zu Beginn des 20. Jahrhunderts wurde der Infrastruktur-Begriff verzeichnet. Der lateinische Ursprung charakterisiert einen gewissen Unterbau, der aus der Sicht des Betrachters allgegenwärtig und verfügbar ist. Die unterschiedlichen Anwendungsbeispiele zeigen, dass neben den hauptsächlich genannten Bereichen Wirtschaft und Militär ebenso Gebiete aus anderen Disziplinen wie Kunst, Artefakte und Prozesse als Infrastruktur gesehen werden können. Insofern bietet die zuletzt genannte Definition gerade wegen ihrer Offenheit und dem Tätigkeitsbezug eine gute Grundlage für die Informatik. Im Folgenden will ich mich mit dem gewonnenen begrifflichen Verständnis von Infrastruktur einer informatischen Perspektive zuwenden, um daran anschließend ein analytisch-methodisches Verständnis aufzuzeigen.

### **3.2 Perspektiven auf Informations-Infrastrukturen**

Bereits seit Beginn der 80er Jahre versuchen verschiedene Forscherinnen und Forscher sich anhand von Fallbeispielen (siehe z. B. Kapitel 2) einen Zugang zu Infrastrukturen zu erarbeiten. Die Fallbeispiele können als Vorläufer informatischer Infrastrukturen gesehen werden. Seit Mitte der 90er Jahre nimmt die Politik Einfluss auf informatische Infrastrukturgestaltung. Die verschiedenen Herangehensweisen sollen hier exemplarisch vorgestellt werden, um einen Überblick über den aktuellen Stand der Diskussion zu geben und eine Einordnung des Forschungsschwerpunktes meiner Arbeit darzulegen. In Bezug auf informationstechnische Infrastrukturen muss ich feststellen, dass die hier aufgeführten Autoren aus den verschiedenen Gebieten bisher kein einheitliches Bild für die Auseinandersetzung mit Infrastrukturen zeichnen (Ciborra 2000a). Sie haben sich jeweils einen eigenen analytischen Zugang erarbeitet und bieten keine konstruktive Orientierung.

„Informations-Infrastrukturen“ oder „information infrastructures“ haben politische Impulse aufgrund der Pläne der amerikanischen Regierung (Gore 1994) und der Europäischen Union (Bangemann 1997) bekommen. Sie sind zum Thema öffentlicher Diskussion geworden. Diese Entwicklung wurde von der OECD aufgegriffen. Hierbei wird von den genannten Institutionen unter einer Informations-Infrastruktur die ökonomische und technische Ausstattung der entwickelten Länder verstanden, um Bürgerinnen und Bür-

gern den hinderungsfreien Zugang zu Informationen aller Art zu ermöglichen (OECD 1996). Man verspricht sich dadurch positive Effekte auf vielerlei Ebenen. Ausgangspunkt des beschriebenen Zugangs ist die Idee einer *Informationsgesellschaft*. Dafür wird eine der Gesellschaft zu Grunde liegende Infrastruktur gefordert, die als eine stabile Struktur, eine gemeinsame Ressource und ein gemeinsamer Standard verstanden wird. Diese Idee kontrastiert z. B. Dahlbom durch das Konzept einer “nomadic society” (Dahlbom 2000). Die betrachtete Ebene zeigt, in welcher Breite das Thema „Infrastruktur“ diskutiert wird, kann aber für die Konstruktion von Software nur wenig beitragen.

Gasser und Huhns haben mit ihren Arbeiten zur Infrastruktur eine Nähe zur Distributed Artificial Intelligence entwickelt, indem sie sich mit komplexen verteilten Software-Anwendungen beschäftigt haben, die aus teilweise autonom agierenden so genannten “blackboard systems”, “problem solvers” und “multiagents” bestehen (Gasser und Huhns 1989). Diese der Informatik nahe stehenden Arbeiten liegen nicht im Interessensbereich meiner Forschung. Eine Infrastruktur mit dem Schwerpunkt auf Informationsbeschaffung und Bewertung wird von Gluchowski und anderen unter dem Begriff Data Warehouses zusammengefasst (Gluchowski 1997, Pantelic und Nohr 2000). Sie rechnen Management-Informationssysteme zu diesem Gebiet dazu und sehen als Zweck der komplexen Software-Anwendung die Zusammenführung von Informationen von Unternehmensstandorten, die im Management für Entscheidungsprozesse verwendet werden können. Auch dieser Betrachtungsweise werde ich mich im Rahmen dieser Arbeit nicht zuwenden.

Hanseth und Monteiro haben sich schwerpunktmäßig mit “information infrastructures” beschäftigt. Sie nennen sechs Aspekte, die es meiner Ansicht nach insbesondere für Software-Infrastrukturen zu berücksichtigen gilt:

- “1. Infrastructures have a supporting or enabling function.
2. An infrastructure is shared by a larger community (or collection of users and user groups).
3. Infrastructures are open.
4. I[nformation]I[nfrastructure]s are more than “pure” technology, they are rather socio-technical networks.
5. Infrastructures are connected and interrelated, constituting ecologies of networks.

6. Infrastructures develops [sic] through extending and improving the installed base.” (Hanseth und Monteiro 1998b, S. 33-42)

Ich werde mich an die folgende Charakterisierung von Informations-Infrastruktur anschließen, da sie einen klaren Bezug zum Arbeitshandeln mit den verbundenen Werkzeugen und in Gruppen etablierter Arbeitspraxis herstellt:

“Information infrastructures provide the tools – words, categories, information processing procedures – with which we can generate and manipulate knowledge. They also reify particular configurations of work practice by shaping the world within which tools can be used. Both the silences and the explicit categories are important for linking work and infrastructure – infrastructure development is both social and informatic.”

(Star und Bowker 1995, S. 41)

Aufgrund der unterschiedlichen Begrifflichkeit, die von verschiedenen Autoren verwendet wird, will ich in dieser Arbeit zuerst den Begriff „informationstechnische Infrastruktur“ klären. Ich möchte ihn von dem Begriff „Informations-Infrastruktur“ abgrenzen, der vielfach in der Literatur verwendet wird. *Informationstechnische Infrastruktur* wird von mir in folgendem Sinne verwendet:

*Definition 2 (informationstechnische Infrastruktur)*

Eine *informationstechnische Infrastruktur* ist eine durch die Sichten von Personen konstituierte Infrastruktur (vergleiche Definition 1 auf Seite 40) und besteht in der Zusammenstellung von Personen, Organisationen, Technik, Vereinbarungen und damit verbundenen Dienstleistungen für die Erledigung von Aufgaben, bei denen Informationstechnik, Informationsverarbeitung und Informationen im Vordergrund stehen.

*Personen* bezeichnet hierbei sowohl die Nutzenden als auch die Betreibenden. *Organisationen* verweist auf alle im Kontext der Dienstleistung beteiligten Unternehmen mit den relevanten Personen und der relevanten Technik. *Technik* bezeichnet die notwendigen Artefakte wie Hard- und Software, die zum Funktionieren der Infrastruktur benötigt werden. *Vereinbarungen* sind alle zwischenmenschlichen Konventionen und technischen Standards, die zum Zusammenarbeiten benötigt werden. *Dienstleistungen* sind von Maschinen oder Personen ausgeführte Aktionen, die zur Nutzung der Technik notwendig sind.

Im Zentrum meiner Arbeit steht eine softwaretechnische Perspektive. In deren Kontext ist es für mich notwendig, aufbauend auf der oben gegebenen Definition von informationstechnischer Infrastruktur eine Definition von *Software-Infrastruktur* wie folgt einzuschränken:

*Definition 3 (Software-Infrastruktur)*

Eine *Software-Infrastruktur* ist eine informationstechnische Infrastruktur (vergleiche Definition 2 auf der vorherigen Seite), bei der nur die anwendungsbezogenen Software-Bestandteile betrachtet werden und die existierende Hardware und Systemsoftware vorausgesetzt wird.

Diese *Software-Infrastruktur* ist meines Erachtens ein wichtiger neuer Forschungsgegenstand der Softwaretechnik, mit dem ich mich von nun an auseinander setzen möchte. Herkömmliche softwaretechnische Methoden und Vorgehensweisen befassen sich mit der Herstellung einzelner Software-Produkte (McDermid 1991, Sommerville 2001, Floyd und Züllighoven 1998) oder der von Software-Familien (Parnas 1999). Einige betrachten Software als Plattform, d. h. ein Produkt zur Entwicklung hoch entwickelter Anwendungen für skalierbare, hoch-verfügbare Dienste (Carey und The Propel Platform Team 2001). In jüngster Zeit gewinnt die Frage nach angemessenen Software-Entwicklungsmethoden für langlebige Software und insbesondere Infrastrukturen zunehmend an Bedeutung. Dies hat u. a. Weber aufgegriffen<sup>2</sup> und mit dem Stichwort *Continuous Engineering* benannt. Allerdings adressiert er das beschriebene Problem schwerpunktmäßig mit formalen Methoden und vernachlässigt damit ein ganzes Spektrum von aus meiner Sicht relevanten Faktoren (vergleiche hierzu Floyd et al. 1999). Die bereitgestellten Methoden erfassen derzeit aus meiner Sicht nicht die speziellen Probleme der Software-Infrastrukturen.

Für mich als Infrastruktur-Entwickler stellen sich Fragen folgender Art:

---

2 „Daneben sind Informations- und Kommunikations-Infrastrukturen die Voraussetzung dafür, unabhängig voneinander entwickelte Systeme zu integrieren. Sie dienen also auch der ‘Vernetzung (technischer) Systeme’, d. h. durch sie entstehen aus zunächst autonomen kommerziellen Anwendungssystemen, Dispositions- und Entscheidungsunterstützungssystemen, Bürokommunikations- und Dokumentenverarbeitungssystemen, Führungs- und Leitsystemen, Multimedia-Systemen, Datenhaltungs- und Datenverwaltungssystemen, Kommunikationssystemen, Werkzeugen und Werkzeugumgebungen, hochkomplexe integrierte Systeme, in denen vielfältige Wechselwirkungen zwischen den Einzelsystemen auftreten.“ (Weber 1999, S. 17)



- Wie werden Themen, Anlässe oder Ausgangspunkte für relevante Entwicklungsvorhaben an einer bestehenden Software-Infrastruktur identifiziert?
- Wie können für ein gegebenes Entwicklungsvorhaben die relevanten Akteure ermittelt und in den Entwicklungsprozess einbezogen werden?
- Wie gehen die Software-Entwickler mit unklaren und gegensätzlichen Zielvorstellungen um?
- Wie kann man in einem Spannungsfeld bestehen, das einerseits keine Technologie auf Vorrat einzubinden verlangt (“keep it simple, stupid”) und andererseits möglichst offen für zukünftige Entwicklungen sein soll?
- Wie erreicht man es, dass die Software-Infrastruktur im Spielraum der Benutzer in den für den Anwendungskontext notwendigen Grenzen anpassbar ist?
- Wie kann eine bestehende Software-Infrastruktur reproduziert werden?
- Wie erkennt man, ob die Software-Infrastruktur für die nutzenden und entwickelnden Personen ausreichend alltagstauglich ist und wie kann man diese Eigenschaft konstruktiv erhöhen?
- Wie erschließt man existierende Gegebenheiten und bezieht diese in die Entwicklung ein?
- Wie können existierende Standards ausreichend berücksichtigt werden?
- Wie lassen sich negative Wechselwirkungen bei der Entwicklung frühzeitig erkennen und/oder gegebenenfalls vermeiden?

Ich möchte mich zuerst an den bisherigen Bemühungen orientieren. Mir stehen insbesondere die Ansätze von Kling, Bowker und Star nahe (Kling und Scacchi 1982, Kling 1987, Kling und Iacono 1989, Kling 1992, Bowker 1994, Bowker 1994, Bowker und Star 1994, Star und Bowker 1995, Star und Ruhleder 1996, Bowker und Star 1998, Bowker und Star 1999), da sie mit menschenzentrierten Ansätzen einer anwendungsorientierten Software-technik besonders gut harmonieren. Darüber hinaus sind die Konzepte von

Latour, Callon, Bijker und Law (Latour und Woolgar 1986, Callon et al. 1986, Latour 1999, Bijker und Law 1992, Latour 1996a) aus der Wissenschaftsforschung von Interesse. Ich befasse mich im Folgenden also mit Theorieansätzen, die sowohl die technischen als auch die sozialen Faktoren berücksichtigen.

### 3.3 Analytische Konzepte

Als Vorbereitung für die Auseinandersetzung mit den empirischen Erfahrungen möchte ich drei Ansätze vorstellen, die als erstes analytisches Hilfsmittel für das Anwendungsgebiet verwendet werden können: Das *Web-Modell* von Rob Kling, die *Infrastruktur-Kriterien* von Star und Ruhleder und die *Akteur-Netzwerk-Theorie* von Bruno Latour, Michell Callon und anderen.

Aus einer philosophischen Perspektive sieht Welsch als eine Eigenschaft der Moderne die künstliche Trennung zwischen Technik und Gesellschaft (Welsch 1997). Hier erkenne ich eine Parallele zu verschiedenen Bemühungen, Infrastrukturen klarer zu charakterisieren. Einige Ansätze gehen dabei so vor, dass sie die technischen Gegebenheiten von den Menschen getrennt betrachten. Durch diese Trennung wird das eine häufig dem anderen vorgezogen. Kling zeigt in der Beschäftigung mit Technik bei Analysemethoden denselben Gegensatz auf. Er differenziert zwischen *sozio-technischen* Ansätzen, die die Relation zwischen Dingen und Menschen in den Vordergrund stellen, und *diskreten Analysemodellen*, die auf den Besitz und die Anzahl von verfügbaren Ressourcen abzielen. Die diskreten Analysemethoden legen nahe, Infrastrukturen als bestehend aus IT-Komponenten, beteiligter „sozialer Infrastruktur“ (Menschen, Qualifikationen etc.), gemeinsam genutzten IT-Dienstleistungen und gemeinsam genutzten Programmen zu sehen (Weill und Broadbent 1998). Diese Bestandteile betrachten sie voneinander unabhängig, lassen Beziehungen unbeachtet und nehmen keinen ausreichenden Bezug auf den Entwicklungsprozess. Diese Interpretation wird in wirtschaftlichen Untersuchungen vielfach verwendet.

Ich halte es für notwendig, bei der Auseinandersetzung mit Infrastruktur Technik und Menschen zusammen zu betrachten und insbesondere auf die Relation zwischen diesen einzugehen. Verschiedene Autoren suchen deshalb seit längerem nach Zugängen, die in erster Linie die erwähnte Trennung vermeiden und gleichzeitig den relationalen Charakter stärker in den

Vordergrund stellen. Das unterstreicht die Relevanz der hier vorgestellten Ansätze. Sie wurden darüber hinaus ausgewählt, weil sie bereits vielfach im IT-Bereich im Zusammenhang mit Infrastruktur-Untersuchungen verwendet worden sind. Wann immer Informations- oder Software-Infrastrukturen im Zusammenhang mit sozialen Phänomenen betrachtet werden, nehmen die relevanten Autoren eines der hier vorgestellten Konzepte. Es erscheint mir deshalb nicht angebracht, aus einer softwaretechnischen Perspektive einen radikal neuen Ansatz zu entwickeln. Vielmehr ist es sinnvoll, die bestehenden zu untersuchen und ihre Charakteristika zusammen mit ihren Stärken und Schwächen für die empirischen Fälle zu beleuchten und gegebenenfalls zu synthetisieren.

Kling sieht in den von ihm wegen der Kombination von sozialen Akteuren und technischen Artefakten als sozio-technisch charakterisierten Ansätzen den Vorteil, dass sie stärker auf die Beziehungen der beteiligten Akteure fokussieren. Er konstatiert folgende Nachteile der diskreten Ansätze:

“Discrete-entity models lack the conceptual richness to easily explain phenomena such as these:

- a. [...] service will often be differentially provided, and more powerful participants will often receive better levels of service where the provider has discretion. [...]
  - b. [...] Another division of the firm chose a different primary vendor, so they would have a plausible rationale to acquire or develop their own (incompatible) software. [...]
  - c. [...] When performance criteria are ambiguous, social definitions may dominate decisions about adoption.
  - d. Sometimes there are conflicts between the kind of computer-based service a consumer desires and what a provider sees as his line of work or the going concern of his organization. [...]
- (Kling 1992, S. 32-33)

Kling beschreibt “computing infrastructure” als ein organisationsbezogenes Phänomen. Zu dieser Auffassung kommt er, indem er ein “web of social relationships” (Kling 1992, S. 34) verfolgt und charakterisiert. Die von ihm als adäquat herausgestellte Technik zur Analyse ist folglich ein Web-Modell.

Die Herangehensweise von Star und Ruhleder stellt den relationalen Charakter von Infrastruktur in den Mittelpunkt der Analyse (Star und Ruhleder 1994), indem sie sich an Bateson orientieren. Mit „relational“ betonen

sie, dass es auf die Beziehungen zwischen den betrachteten Dingen und Menschen ankommt. Dabei betrachten sie die auftretenden Probleme auf drei aufeinander aufbauenden Ebenen. Sie geben neun Kriterien an (Star und Ruhleder 1996, S. 113), um den infrastrukturellen Charakter von Artefakten zu bewerten.

Die Akteur-Netzwerk-Theorie wird primär für die Analyse sozio-technischer Phänomene verwendet. Sie passt insbesondere deshalb gut zum Infrastruktur-Ansatz, da sie die Betrachtung als Netz zu Grunde legt und gleichzeitig Mittel zum Vergrößern anbietet. Darüber hinaus setzt sie einen Schwerpunkt in der Beziehung von Akteuren zueinander, was ebenso im Mittelpunkt der Infrastruktur-Kriterien von Star und Ruhleder steht.

Im Folgenden werden diese Ansätze auf die von ihnen angebotenen Mittel untersucht. Die Ausdrucksmöglichkeiten und Grenzen sollen dabei aufgezeigt werden. Abschließend wird versucht, die Tauglichkeit für die hier vorliegende empirische Basis einzuordnen.

### 3.3.1 Das Web-Modell

Kling unterstreicht die Wichtigkeit von Computer-Infrastrukturen, zu denen er neben den „harten“ Ressourcen auch die „weichen“ rechnet, wie z. B. Qualifikation. Er definiert eine Computer-Infrastruktur als eine Zusammenstellung von physischen, technischen und sozialen Ressourcen. Für die Analyse hält Kling so genannte Web-Modelle für das geeignete Mittel. Unter einem Web-Modell versteht er ein natürliches Modell eines offenen Systems, bei dem „natürlich“ ausdrückt, dass das Modell ausgehend vom interessierenden Gegenstand wächst. Das Modell expliziert die interessierende Technik und deren konkrete Benutzer. Darüber hinaus werden die sozialen Beziehungen zu anderen sozialen Gruppen und Organisationen, in denen die Technik entwickelt, betrieben und angewendet wird, beschrieben. In dieser Art von Modell werden Computersysteme von einem Netz von Produzenten und Konsumenten entwickelt, betrieben und benutzt. Deswegen können die Computersysteme für sich allein betrachtet als diskrete Komponenten nicht angemessen analysiert werden.

“Web models define a social context for computer-based technologies by taking into account:

- the *social relations between a set of participants* who can influence the adoption, development, or use of the focal computing technologies;

- the *infrastructure* available for their support;
- the *history* of commitments made in developing and operating related computer-based technologies.” (Kling 1992, S. 4)

Web-Modelle sind deshalb besser geeignet, weil sie den Kontext einbeziehen und sich nicht an einzelnen Bestandteilen orientieren. Hochwertige Hardware zu besitzen würde sonst beispielsweise automatisch heißen, in Bezug auf Infrastruktur wohlhabender zu sein, selbst wenn man diese Hardware nicht benutzen kann. So kann man beim Austausch von alten Geräten durch höherwertige real ärmer werden, wenn man dadurch bekannte Nutzungsmöglichkeiten verliert. Es geht also nicht darum, isolierte Entitäten zu analysieren und aufzusummieren, wie es diskrete Analysemethoden vorschlagen, sondern organisatorische Arrangements als Teil eines größeren Systems zu begreifen.

Das nachfolgende Zitat fasst die statischen Eigenschaften von Computer-Infrastrukturen aus der Sicht Klings zusammen:

- “a. Infrastructural resources are layered. The focal computing resource for one participant (e. g., the operating system as seen by a systems programmer) is infrastructure for participants further up the (production) chain. (Infrastructure is a *relationship* between a focal resource and a supporting resource.) The computing infrastructure in organisations has often become more complex with the spread of multiple systems with different components in different organisational units and systems connected through communications networks. The communications wiring diagrams often just hint at the social complexity of the infrastructure for specific applications which draw data from one system, process it on another, and print it on third. When system components are owned by different departments and located in different physical locations, practical usage is sometimes complicated.
- b. Infrastructural resources for computing may be coextensive with or part of other organisational procedures and resources. At various points, the mail room which disseminates reports, the systems which provide electrical power and air conditioning, procedures for hiring and firing staff, or for procuring equipment, can be drawn into the support of a focal computer system. [...]
- c. Some of the resources are relatively concrete objects (e. g., a document, contract), or people (e. g., skilled programmers). Others are capabilities for reproducing these more concrete resources on demand (e. g., the

ability to hire or train skilled programmers, the ability to purchase, or write new documents).”

(Kling 1992, S. 28, Hervorhebung im Original)

Diese reichen seiner Ansicht nach allerdings nicht aus, um die Infrastruktur hinreichend zu beschreiben. Dafür sind Web-Modelle notwendig, die sich wesentlich besser zur Analyse eignen, da sie diejenigen Dinge, Entscheidungen und Zusagen berücksichtigen, die vor dem Untersuchungszeitraum bereits existierten. Insbesondere können diese Dinge nicht leicht verändert werden. Vielmehr werden Infrastrukturen nach Kling ausgehend von einer bestehenden Basis inkrementell weiterentwickelt.

Kling unterscheidet im Zusammenhang mit Computer-Infrastrukturen zwischen Tätigkeiten und Ressourcen, Strategien und Benutzung (Kling 1992, S. 10–17). Unter *Tätigkeiten und Ressourcen* versteht er Daten, Komplexität, Geld und physische Umgebung sowie die damit verbundenen Tätigkeiten. Unter *Strategien* werden subsummiert: Ausgabenminimierung, technologische Fähigkeiten, Einfachheit der Handhabung und Wartung, Unterstützung der geografischen Verteiltheit und Einbettung der eigenen Computer-Arrangements in die der Kooperationspartner. Er sieht als *Benutzung* Dokumentenerstellung, Mitarbeiterausbildung und Computeradministration. Benutzung wird als ein Geflecht von sozialen Objekten betrachtet, welche durch die Beziehungen Einfluss nehmender Teilnehmer strukturiert werden. Die Idee des Web-Modells liegt darin, explizit den Fokus auf die Beziehungen zu setzen.

Kling differenziert bei der Analyse von Entwicklungen im Bereich der Rechnerausstattung zwischen drei konzeptionellen Ebenen: Situation, Kontext und Prozess (Kling 1992, S. 17). Eine Situation ist die Betrachtungseinheit, in der Einzelne oder Gruppen ihren Aktivitäten nachgehen und dazu Computer-basierte Technik verwenden. Jede Nutzung von Computer-basierter Technik definiert automatisch eine *Situation*, indem Teilnehmende, Geräte, Orte und Zeiträume konkretisiert werden. Situationen können sich durch die Anzahl der Teilnehmenden, die relevanten Gegenstände, die durchgeführten Tätigkeiten, den Zeithorizont und den sozialen Prozess unterscheiden.

Die Situation dient als Grundlage für die Definition des Kontextes. Der *Kontext* wird dabei durch die sozialen Beziehungen und Strukturen innerhalb einer größeren Situation beschrieben. Es gibt dabei drei Kriterien zur

näheren Bestimmung der größeren Situation, die durch die beteiligten Personen, die Ausstattung, sowie räumliche und zeitliche Rahmenbedingungen (engl. population, equipment, spatial and temporal elements, PEST) eingegrenzt sind. Das erste Kriterium ist, dass die Akteure in der zu betrachtenden Situation von den PEST-Elementen der umgebenden Situation abhängen. Das zweite Kriterium drückt aus, dass die Akteure auf die PEST-Elemente Bezug nehmen, wenn sie in der zu betrachtenden Situation handeln. Das dritte Kriterium schließlich besagt, dass die PEST-Elemente der größeren Situation die Bedingungen festlegen, unter denen die Akteure in der zu betrachtenden Situation handeln.

Dieser so definierte Kontext bietet den Rahmen für ein Verständnis von Prozessen, die in Organisationen ablaufen. Dazu werden die Pfade verfolgt, an denen Handlungen entlanglaufen, die mit Techniknutzung in Verbindung stehen. Im Mittelpunkt des Interesses stehen die organisatorischen Rahmenbedingungen, in denen die Prozesse stattfinden, und die Muster von Anreizen und Bedingungen, die sie beeinflussen.

Kling beschreibt abschließend ein Verfahren, mit dem sich die Beziehungen innerhalb einer Infrastruktur erschließen lassen. Er gibt drei Wege an, um das von ihm beschriebene Gitterwerk, den “production lattice”, zu identifizieren. Erstens kann die Kette von sozialen und technischen Interaktionen von der interessierenden Computer-Ressource nach außen zu den verschiedenen Bereitstellern zurückverfolgt werden. Zweitens können die Eingaben oder Ressourcen einbezogen werden, die Verbesserungen in der Qualität der Ergebnisse der interessierenden Computer-Ressource bringen. Und schließlich können die Eingaben und Ressourcen aufgenommen werden, die die Qualität der Ergebnisse verschlechtern.

In seiner abschließenden Aussage verdeutlicht Kling allerdings, welchen Einschränkungen die Analyse nach dem Web-Modell unterliegt: Es gibt weder eine feste Methode noch einen definierten Prozess, sondern das Modell beschreibt lediglich einen Rahmen und einen Satz von Begriffen.

Das Web-Modell eignet sich also für die Analyse der empirischen Projekte insofern, als dass es den methodischen Rahmen liefert, mit dem, ausgehend von den jeweils interessierenden technischen Ressourcen, das relevante Umfeld ermittelt werden kann. Der Schwerpunkt liegt hier in der Beschreibung der jeweiligen Beziehungen und des abhängigen Aufbaus eines Geflechts. Es fehlen spezifische Begriffe, um die Charakteristika von Infrastrukturen ausreichend zu beschreiben. Dafür sollen die Infrastruktur-

Kriterien von Star und Ruhleder im nächsten Abschnitt herangezogen werden.

### 3.3.2 Die Infrastruktur-Kriterien von Star und Ruhleder

Star und Ruhleder untersuchen die beeinflussende Eigenschaft von Technik in Organisationen (Star und Ruhleder 1996). Dafür haben sie die *Infrastruktur-Kriterien* formuliert, mit denen sich die beobachteten Phänomene besser beschreiben lassen. Technik und damit Infrastruktur besitzt eine paradoxe Natur: Sie ist zugleich Motor und Barriere für Wandel; anpassbar und starr; sowohl innerhalb wie auch außerhalb von organisatorischer Praxis; sie ist Produkt und Prozess (Star und Ruhleder 1996, S. 111). Es fällt deshalb schwer, Infrastruktur mit einem einfachen Modell zu charakterisieren. Mit der fortschreitenden Durchdringung des gesellschaftlichen und beruflichen Lebens mit technischen Geräten und durch die Verbreitung von Technik und deren stärkere Dezentralisierung wird das Bedürfnis nach Standards und gleichzeitig nach Anpassbarkeit und Flexibilität immer stärker (Star und Ruhleder 1996, S. 112). Somit wächst der widersprüchliche Charakter der Bedürfnisse in gleichem Maße an und es entsteht ein Spannungsfeld zwischen Technik und Organisation.

Ein Grund für das Spannungsfeld liegt in der Tatsache, dass Personen fast immer Mitglied in mehr als einer Gemeinschaft sind. Daraus folgen unterschiedliche Nutzungsmuster, die jeweils unterschiedliche Anforderungen an die Anpassbarkeit bzw. Standardisierbarkeit von Werkzeugen stellen (Star und Ruhleder 1996, S. 112). Star und Ruhleder stellen deshalb fest, dass dafür eine andere Betrachtungseinheit notwendig ist. Der von Wenger geprägte Begriff der *community of practice* wird deshalb von ihnen verwendet (Wenger 1998). Unter einer "community of practice" versteht Wenger eine (kleine) Gruppe von Menschen, die über eine gewisse Zeit zusammen gearbeitet und durch intensive Kommunikation ein gemeinsames Verständnis herausgebildet hat. Im Deutschen werde ich dafür den Begriff *Praxisgemeinschaft* verwenden.

Star und Ruhleder bieten ein analytisches Vokabular an, mit dessen Hilfe die Frage nach dem Verhältnis zwischen großen Infrastrukturen und Praxisgemeinschaften beantwortet werden kann. In nachfolgendem Zitat finden sich die neun Punkte wieder, die von ihnen benannt werden: Eingebundenheit (*embeddedness*), Transparenz (*transparency*), Reichweite (*reach or*



*scope*), erlernt in der Mitgliedschaft (*learned as part of membership*), verbindet mit Konventionen des Handelns (*links with conventions of practice*), Verkörperung von Standards (*embodiment of standards*), errichtet auf existierender Basis (*built on an installed base*), wird sichtbar durch Ausfall (*becomes visible upon breakdown*) und wird inkrementell festgelegt (*is fixed in modular increments, not all at once or globally*).

- “• *Embeddedness*. Infrastructure is sunk into, inside of, other structures social arrangements, and technologies
- *Transparency*. Infrastructure is transparent to use in the sense that it does not have to be reinvented each time or assembled for each task, but invisibly supports those tasks.
- *Reach or scope*. This may be either spatial or temporal – infrastructure as reach beyond a single event or one-site practice;
- *Learned as part of membership*. The taken-for-grantedness of artifacts and organisational arrangements is a sine qua non of membership in a community of practice (Lave und Wenger 1991, Star und Ruhleder 1996). Strangers and outsiders encounter infrastructure as a target object to be learned about. New participants acquire a naturalised familiarity with its objects as they become members.
- *Links with conventions of practice*. Infrastructure both shapes and is shaped by the conventions of a community of practice; for example, the ways that cycles of day-night work are affected by and affect electrical power rates and needs. Generations of typists have learned the QWERTY keyboard; its limitations are inherited by the computer keyboard and thence by the design of today’s computer furniture (Becker 1982).
- *Embodiment of standards*. Modified by scope and often by conflicting conventions, infrastructure takes on transparency by plugging into other infrastructures and tools in a standardised fashion.
- *Built on an installed base*. Infrastructure does not grow de novo; it wrestles with the inertia of the installed base and inherits strengths and limitations from the base. Optical fibres run along old railroad lines, new systems are designed for backward compatibility; and failing to account for these constraints may be fatal or distorting to new development processes (Monteiro et al. 1994).
- *Becomes visible upon breakdown*. The normally invisible quality of working infrastructure becomes visible when it breaks: the server is down, the bridge washes out, there is a power blackout. Even when there are backup mechanisms or procedures, their existence further highlights the now visible infrastructure.

- *Is fixed in modular increments, not all at once or globally.* Because infrastructure is big, layered, and complex, and because it means different things locally, it is never changed from above. Changes take time and negotiation, and adjustment with other aspects of the systems involved.”

(Star und Ruhleder 1996, S. 113)

Genauso wie ein Werkzeug *in situ* als solches entsteht, ist eine Infrastruktur ein Gebilde, das durch das (gemeinsame) Handeln von Personen entsteht. Daraus folgt, dass eine Infrastruktur keine Sache sein kann, sondern eine Beziehung darstellt. Etwas wird zu einer Infrastruktur durch organisatorisches Handeln. Demnach verbietet sich die Frage „Was ist eine Infrastruktur?“, vielmehr sollte man fragen: „Wann ist etwas eine Infrastruktur?“

Um die Kommunikation der Beteiligten über Infrastruktur genauer zu untersuchen, verwenden Star und Ruhleder die von Bateson unterschiedenen drei Ebenen in kommunikativen Systemen (Bateson 1972). Auf der ersten Ebene werden Aussagen über Fakten gemacht, z. B.: „Der Computer ist mit dem Internet verbunden.“ Auf der zweiten Ebene werden Aussagen über Aussagen gemacht, wie z. B.: „Es war eine Lüge, als ich sagte, dass der Computer mit dem Internet verbunden ist.“ Und auf der dritten Ebene werden Aussagen über den Kontext von Aussagen gemacht, z. B.: „Es gibt viele Möglichkeiten herauszufinden, ob ich über die Internetverbindung gelogen habe oder nicht.“ Prinzipiell ließe sich diese Kette unendlich fortsetzen. Allerdings scheinen diese drei ersten Ebenen und die mit ihnen verbundene Komplexität für die Betrachtungen auszureichen (Star und Ruhleder 1996, S. 117).

Werden diese Ebenen auf Infrastruktur angewendet, dann zeigt sich, dass Angelegenheiten erster Ordnung durch Verteilung oder Verfügbarmachen von Ressourcen gelöst werden können. Probleme, die hier entstehen, liegen in fehlenden Informationen begründet, treten durch schlechte Zugangsmöglichkeiten auf oder werden durch mangelnde oder unpassende Qualifikation induziert. Vordergründig können Probleme dieser Kategorie einfach gelöst werden. Jedoch sind sie häufig mit noch unerkannten der höheren Stufen verknüpft (Star und Ruhleder 1996, S. 118-120).

Probleme auf der zweiten Ebene rühren von unvorhergesehenen Effekten im konkreten Kontext her, z. B. dem Zusammentreffen von zwei Problemen der ersten Ebene. Hierzu zählen Probleme technischer Vorlieben (Bevorzugung bestimmter Hard- oder Software), aufeinander treffende kulturelle Gegensätze, Paradoxien in der Infrastruktur (schlechte Technik – gute

Nutzung; gute Technik – schlechte Nutzung) sowie Vertröstungen. Lösungen können durch verbesserte Kommunikation erfolgen, indem z. B. eine Hotline bereitgestellt wird. Allerdings können die Probleme zweiter Ordnung leicht zu welchen dritter Ordnung werden, wenn es z. B. um Fragen der Finanzierung einer Hotline geht (Star und Ruhleder 1996, S. 120-121).

Auf der dritten Ebene werden hauptsächlich politische Fragen diskutiert oder grundsätzliche Streitigkeiten herangezogen. Dies beinhaltet auch Triangulation (Flick 1998, S. 249-251) und Definition von Gegenständen, die Zuordnung von Bedeutung, die Interpretation von Daten und das Abgrenzen von Gebieten. Grundsätzlich sind diese Eigenschaften auf der dritten Ebene Merkmale komplexer Gemeinschaften. Sie werden erst nach einer gewissen Zeit erkennbar, sind allerdings ausdrücklich stabiler als die anderen beiden Ebenen (Star und Ruhleder 1996, S. 123-126).

Die Stärke der Infrastruktur-Kriterien liegt in der Aussagekraft bei der Analyse von technischen Infrastrukturen in sozialen Kontexten mit einem Schwerpunkt auf dem Verstehen und Erlernen der Nutzung. Die Kriterien sind deshalb ein wertvolles Analysemittel zur Beschreibung von Verstehensprozessen und damit verbundenen Effekten in der Infrastrukturnutzung. Darüber hinaus bieten die Kriterien einen Katalog, an dem existierende Infrastrukturen erkannt und bewertet werden können. Sie ergänzen damit das von Kling vorgestellte Web-Modell. Keine Hilfestellung bieten die Kriterien auf der Prozessebene, da sie keine Begrifflichkeit für einen Analyseprozess bereitstellen. Die Akteur-Netzwerk-Theorie liefert einen ganzen Satz von Begriffen und Verfahren für den Umgang mit solchen Netzen, sodass sie als drittes analytisches Konzept vorgestellt werden soll.

### 3.3.3 Die Akteur-Netzwerk-Theorie

Die *Akteur-Netzwerk-Theorie*<sup>3</sup> (ANT) wurde von Bruno Latour, Michel Callon und anderen entwickelt (Latour und Woolgar 1986, Callon et al. 1986, Latour 1999), um die Soziologie von Wissenschaft näher zu untersuchen. Später wurde sie auf Technik (Bijker und Law 1992, Latour 1996a) und Informationstechnik (Latour 1996b, McMaster et al. 1998, Monteiro und Hepsø 1998) bezogen. Ihre Kernkonzepte sind neben Akteur/Aktant

---

3 Die Übersetzung des englischen Begriffs "network" ins Deutsche wäre korrekt „Netz“. Leider hat sich bereits der Begriff „Netzwerk“ durchgesetzt. Deshalb bleibe ich in diesem Zusammenhang bei der etablierten Wortwahl.

(*actor*) und Netzen (*network*) auch Kapselung (*black box*), Delegierte (*delegates*), Eintragung (*inscription*), Beitritt (*enrolment*), Übersetzung (*translation*), Unumkehrbarkeit (*irreversibility*) und unveränderliches Bewegliches (*immutable mobile*) (Walsham 1997). Akteure treten grundsätzlich in Netzen auf und ihre Beziehung zueinander wird mithilfe der o. g. Konzepte modelliert.

Wegen der sprachlichen Nähe des deutschen Wortes *Akteur* zum englischen *actor* wird dieses in deutschen Texten vielfach verwendet. Leider besitzt es in der deutschen Sprache eine Konnotation von menschlichen oder sozialen Akteuren. Um eine Verwechslung zu vermeiden, werde ich eine unpersönliche Ausdrucksweise wählen und das ebenfalls in der Theorie gebräuchliche Wort *Aktant* im Allgemeinen an dessen Stelle verwenden. Sollte eine Einschränkung auf soziale Aktanten gewünscht sein, werde ich von *Akteuren* bzw. *sozialen Akteuren* schreiben.

Nach ANT besteht die Gesellschaft aus Netzen von unterschiedlichen Aktanten, die sowohl nicht-menschlich als auch menschlich sein können. Es wird argumentiert, dass beide Aktant-Formen, das Soziale und das Technische, mit denselben Konzepten behandelt werden sollen, weil sie untrennbar miteinander verbunden seien:

“It is no longer clear if a computer system is a limited form of organization or if an organization is an expanded form of computer system. Not because, as in the engineering dreams and the sociological nightmares, complete rationalization would have taken place, but because, on the opposite, the two monstrous hybrids are now coextensive.” (Latour 1996b, S. 302)

Gesellschaft, Organisationen, Handelnde und Maschinen sind alles Phänomene, die durch die Interaktion eines Aktant-Netztes entstehen. Eine Person kann demzufolge nicht als eine isolierte Entität verstanden werden, sondern sie ist immer in ein heterogenes Netz von Ressourcen und Handelnden eingebunden, die in ihrer Zusammenstellung diese spezielle Person definieren. Ohne z. B. die Geräte, das Labor oder die sozialen Beziehungen verliert ein Wissenschaftler seine Identität als Wissenschaftler.

Obwohl die Akteur-Netzwerk-Theorie aus der Forschung um wissenschaftliches Handeln entstanden ist, wird sie jetzt als allgemeines Mittel zum Verstehen von Informationstechnik im Kontext verwendet (Monteiro 2000, Tuomi 2001, Klischewski 2001). Ein wissenschaftliches Labor kann gesehen werden als ein Netz von Reagenzgläsern, Protokollen, wissenschaftlichen Publikationen, Fördermitteln und Wissenschaftlern, wobei

jedes Element seine eigenen Zuständigkeiten und Widerstände besitzt. Wissen wird in diesem Netz produziert und wird damit selber zu einem Aktant, indem z. B. neue Konzepte und Beobachtungen in Journalen veröffentlicht oder in neuen wissenschaftlichen Instrumenten bzw. Software vergegenständlicht werden. Law hat gezeigt, dass dies auch auf Familien, Organisationen, Computersysteme, die Wirtschaft und Technik im Allgemeinen übertragen werden kann (Law 1992).

Weil die Summe aller Aktanten in einem Netz schnell sehr groß und damit das Netz komplex werden kann, braucht man ein Mittel zur Reduktion dieser Komplexität. *Kapselung* bezeichnet einen Prozess, in dem man eine Substruktur des Netzes durch einen einzelnen Aktanten repräsentiert, d. h. die komplexe Substruktur wird in einer *Kapsel* (engl. *black box*) versteckt. Man kann z. B. in dem einen Zusammenhang von „dem Rechenzentrum“ sprechen, ohne dass man genauer wissen muss und will, welche Prozesse im Inneren der Organisation ablaufen und welche Personen dazugehören. Es ist üblich, z. B. Organisationen nach außen hin durch eine einzelne Person zu vertreten (Firmen, Regierungen etc.). Gleiches passiert, wenn komplexe Zusammenhänge wie z. B. die Vorgänge bei der Steuerberechnung durch eine Software gekapselt werden.

Eine Kapsel bedeutet also, dass komplexe Subnetze zu einem Knoten verdichtet werden und aus der Sicht von Aktanten, die mit den verdichteten Subnetzen interagieren, diese als eine Einheit auftreten. Gleichzeitig werden die so überführten Subnetze zu Ressourcen. So kann z. B. Software zur Steuerberechnung ohne das komplexe Wissen um die Steuergesetzgebung, Berechnungsvorschriften und Programmierkonzepte eingesetzt werden. Die Kapsel unterstreicht damit die Selbstverständlichkeit, mit der gekapselte Netze verwendet werden können. Darüber hinaus stellt die Kapselung eine Ausübung von Macht und Kontrolle dar. Die Auswirkungen des zusammengefassten Subnetzes werden zu Ressourcen, die verortet und kontrolliert werden können. Durch den Prozess des Kapselns wird das vergrößerte Netz dargestellt, als ob es dem Akteur, der die Kapselung durchgeführt hat, gehören würde.

Das Konzept der *Delegierten* bezeichnet Aktanten, die für einen bestimmten Standpunkt stehen, der in sie *eingetragen* wurde. Das kann z. B. eine Software sein, die eine bestimmte Art des Vorgehens festlegt und somit stellvertretend für die Praxis steht, die bei der Programmierung erdacht wurde. Ein häufig angeführtes Beispiel für Delegierte sind Straßenschwel-

len, die zur Geschwindigkeitsbegrenzung installiert wurden. Sie vergegenständlichen die Absicht, an diesem Ort langsames Fahren durchzusetzen, ohne dass der Urheber der Absicht anwesend sein muss.

Ein weiteres Schlüsselkonzept in der Akteur-Netzwerk-Theorie sind *Beitritt* und *Übersetzung*. Ein Aktant übersetzt sein Interesse oder Ziel in das Ziel eines anderen Aktanten mit der Absicht, mit ihm eine Allianz zu bilden, weil er z. B. allein das Ziel nicht erreichen könnte. Übersetzung bedeutet also, die Interessen eines anderen Aktanten zu kennen und die eigenen in seine zu übersetzen. Beitritt ist dabei die Tätigkeit, bei der sich einzelne Aktanten dem Ziel anschließen.

*Unumkehrbarkeit* bezeichnet den Grad, wie schwer es ist, in der Entwicklung zurückzugehen, bis frühere Alternativen wieder möglich werden. Nach der Anschaffung eines bestimmten Rechnertyps ist es z. B. relativ schwierig, in kurzer Zeit auf eine andere Architektur zu wechseln.

Das Konzept des *unveränderlichen Beweglichen* steht für Aktanten, die in Zeit und Raum wandern und die zu ändern einen besonders großen Aufwand bedeutet. Das viel zitierte Beispiel hierfür ist die Landkarte, die nach ihrer Erstellung von Seefahrern in ihr Heimatland gebracht wird und dort z. B. unverändert für eine lange Zeit die Grundlage für politische Entscheidungen darstellt und auf andere Reisen mitgenommen wird.

Die Akteur-Netzwerk-Theorie hat einen wichtigen Beitrag zu der Diskussion um große technische Systeme geleistet. Sie hilft, die besonderen Eigenschaften großer technischer Systeme und – wie noch zu zeigen sein wird – Infrastrukturen besser zu verstehen. Vorteile von ANT sind insbesondere das Kapseln und die Möglichkeit, zeitweise nicht zwischen menschlichen und nicht-menschlichen Systembestandteilen zu unterscheiden, sondern von Aktanten zu sprechen. Damit erlaubt sie eine wesentlich elegantere Interpretation:

“The third approach, associated with the work of Michel Callon, Bruno Latour, and John Law at L'Ecole des Mines, Paris, attempts to extend this perspective one step further. It does this by breaking down the distinction between human actors and natural phenomena. Both are treated as elements in 'actor networks'. Also, this approach ostensibly reverses the usual relationship between participant and analyst and casts the engineers as sociologists. [...] A characteristic that all these approaches share is the emphasis on 'thick description' [...].” (Bijker et al. 1987, S. 4-5)

Die Stärke von ANT liegt in dem flexiblen Umgang mit Aktanten innerhalb

eines Netzes. Entsprechend der Notwendigkeiten können Vergrößerungen und Detaillierungen genutzt werden. Außerdem werden die Beziehungen der Aktanten im Netz flexibel charakterisiert. Dies erleichtert es, die Interessen von Personen oder Organisationen den relevanten Aktanten zuzuschreiben und zu erkennen, wie diese Inschriften das Handeln anderer limitieren und bestimmen.

### **3.4 Synthese der Analysekonzepte**

Wie bereits der Blick auf den Infrastruktur-Begriff gezeigt hat, existieren Infrastrukturen in einem komplexen Geflecht von Technik, Personen und Praxisgemeinschaft, das über einen langen Zeitraum entstanden ist.

Die vorgestellten Analysekonzepte haben alle das Defizit, dass sich aus ihnen keine operationalisierbaren Handlungsanweisungen ableiten lassen, da es nicht ihr Anspruch ist, Handlungswissen zu vermitteln. Darüber hinaus bieten sie im Umgang mit dem empirischen Material jeweils einzeln keine ausreichende Reichhaltigkeit in der Untersuchung, die ich für die Überleitung zu einem softwaretechnischen Entwicklungsprozess gebraucht hätte. Im Mittelpunkt der Methoden steht jeweils eine unterschiedliche Fragestellung.

Als methodische Grundlage für die Auswertung der Fallstudien soll deshalb eine Synthese der hier vorgestellten analytischen Konzepte verwendet werden. Dabei liefert das Web-Modell ein Basiskonzept für den Umgang mit Computer-Infrastrukturen, dessen Schwerpunkt in der Erschließung der relevanten Artefakte, Personen, Organisationen usw. liegt. Es erlaubt mir, ausgehend von einem Geschehnis die in Beziehung stehende Infrastruktur sukzessive aufzubauen. Allerdings bietet das Web-Modell bei einer großen Anzahl beteiligter Personen, Gruppen, Organisationen und technischer Gegenstände keine direkte Möglichkeit, Vergrößerungen einzubetten, um die Komplexität zu vermindern. Die Akteur-Netzwerk-Theorie hält ein umfangreiches Spektrum an Methoden bereit, um die erschlossene Infrastruktur analytisch zu bearbeiten. Damit kann flexibel im erschlossenen Netz auf unterschiedlichen Granularitäten operiert werden. Beziehungen und Wirkungen lassen sich auf elegante Weise charakterisieren. Damit ergänzen sich Web-Modell und Akteur-Netzwerk-Theorie bei der analytischen Arbeit direkt. Um in diesem Zusammenhang einen Begriffskatalog zur Verfügung zu haben, mit dem speziell Infrastruktur-Phänomene benannt werden

können, habe ich die Infrastruktur-Kriterien von Star und Ruhleder hinzugezogen. Diese stützen sich auf ein Beziehungsgeflecht, welches gerade durch die beiden oben genannten Methoden aufgebaut worden ist. Somit können alle drei Ansätze in Einklang gebracht werden und ergänzen sich im Infrastruktur-Umfeld geeignet. Die synthetisierte Analyse-Methodik wird somit helfen, die Prozessdynamik in Zeit und Raum zu erschließen, den relevanten Rahmen zu erkennen und ausgehend von den Basiseigenschaften von Infrastrukturen die Spezifika für die Software-Entwicklung zu erkennen.

Damit hat der erste Teil des vorliegenden Textes geholfen, eine Basis für ein analytisches Verständnis von Infrastrukturen zu legen. Im nachfolgenden zweiten Teil werden drei Fallstudien als Software-Entwicklungsprojekte aus der Perspektive der Infrastruktur näher mit der eben vorgestellten Kombination der beschriebenen Methoden untersucht. Meine Aufgabe für den dritten Teil wird es sein, eine konstruktive Orientierung für Software-Infrastrukturen zu geben. Es ist meine Absicht in dieser Arbeit, aufbauend auf den ausgewerteten Erfahrungen, eine operationalisierbare Empfehlung für die Entwicklung von Software-Infrastrukturen zu geben.



## **II**

### **Kritische Reflexion der Projekterfahrungen**



## Erfahrungen aus Infrastrukturprojekten

Jeder Schüler kann in der Physikstunde durch Versuche nachprüfen, ob eine wissenschaftliche Hypothese stimmt. Der Mensch aber lebt nur ein Leben, er hat keine Möglichkeit, die Richtigkeit der Hypothese in einem Versuch zu beweisen. Deshalb wird er nie erfahren, ob es richtig oder falsch war, seinem Gefühl gehorcht zu haben. — Milan Kundera

### 4.1 Empirische Grundlagen

Freedom is the right to tell people what they do not want to hear. — George Orwell

Wann nennen wir etwas eine Infrastruktur? In welchen Kontexten entstehen Infrastrukturen und wie werden sie im Alltag verwendet? Welche Rolle übernehmen die Beteiligten? Wie entwickeln sich Infrastrukturen über die Zeit? Diese und weitere Fragen stellen sich bei der Beschäftigung mit dem Infrastruktur-Gedanken. In diesem Abschnitt werde ich die zur Darstellung der empirischen Beobachtungen herangezogenen Ansätze einführen. Die nachfolgenden drei Abschnitte stellen jeweils ein Projekt anhand von exemplarischen Vorfällen dar. Die Darstellung nimmt dabei Bezug auf die analytischen Konzepte des vorangegangenen Kapitels. Dadurch wird ein Verständnis über die beteiligten Aktanten, deren Beziehung untereinander und die Entwicklung dieser Beziehung im jeweiligen Projekt aufgebaut. Dies soll als Basis dafür dienen, die Eigenschaften von Informations-Infrastrukturen im Kontext von Software-Entwicklung zu erschließen. Eine ausführliche Beschreibung der drei gewählten Projekte findet sich im Anhang. Dort wird jeweils über einen historischen Zugang in den konkreten Zeitplan eingeführt und eine umfassende Darstellung der beteiligten Akteure gegeben. Die Anforderungen an die Infrastruktur, Probleme bei der

Nutzung und Rückwirkungen auf die beteiligten sozialen Akteure werden im Anhang ebenfalls charakterisiert.

Die hier aufgezeigten Erkenntnisse gründen sich auf Projekte, in denen ich in den vergangenen Jahren Erfahrungen gesammelt habe. Zu nennen sind u. a. die Entwicklung einer Fax-Server-Software bei der Firma InterCope (1993-1995), die Entwicklung von Call-Center-Software bei der Firma Micrologica (1995-1997, Bleek 1997), die initiale Mitarbeit an der Entwicklung eines Rahmenwerkes für interaktive Anwendungen (1996-1999, Bleek et al. 1999a), die Mitentwicklung der webbasierten Lernplattform *CommSy* für die universitäre Lehre (1999-heute, Bleek et al. 2000), die Beratung eines Entwicklungsteams zur Entwicklung einer webbasierten Software für Stadtinformationssysteme (2000-2002, Bleek 2001), der Aufbau und Betrieb einer technischen Lehr- und Lernplattform in den Gegebenheiten der Universität Hamburg für eine internationale Sommeruniversität für postgraduierte Studentinnen, die *Internationale Frauenuniversität* (1999-2000), sowie die Mitarbeit an der Konzeption und Entwicklung einer Lernplattform für die webbasierte Unterstützung universitärer Lehre (1999-2002) in einem mittlerweile vom Bundesministerium für Bildung und Forschung geförderten Forschungsprojekt (2001-2003, Oberquelle et al. 2001).

Von diesen Projekten greife ich drei mit besonderem Bezug zum Thema Infrastruktur heraus, um an ihnen die Beobachtungen zu verdeutlichen, die zu der thematischen Auseinandersetzung dieser Arbeit mit ihren Fragestellungen und Entwicklungen geführt haben. Dafür stehen meine Projekterfahrungen und die in den Projekten erstellten Dokumentationen zur Verfügung, die ich unter dem Gesichtspunkt der Infrastruktur mit den in Kapitel 3 vorgestellten Mitteln näher untersuchen werde.

#### **4.1.1 Das empirische Material**

Für das Projekt *Internationale Frauenuniversität* stehen als Material drei Langinterviews, eine Vielzahl von kurzen Gesprächsprotokollen, vier aufgezeichnete Gruppeninterviews, persönliche Tagebuchaufzeichnungen und eine persönliche Sammlung von ca. 1500 E-Mails zur Verfügung. Die Langinterviews habe ich als Leitfadeninterviews konzipiert und mit Mitarbeiterinnen der *ifu* durchgeführt. Grundlage für die Gesprächsprotokolle waren sowohl meine „Flurgespräche“ mit Mitarbeiterinnen als auch mit Studentinnen. Die Gruppeninterviews fanden etwa zur Hälfte der Präsenzzeit

statt und wurden von mir zusammen mit Diplomanden mit den jeweiligen Projektgruppen des Projektbereichs Information durchgeführt. Dafür haben wir ebenfalls Leitfäden vorbereitet und die Gruppen ausführlich über das Vorgehen informiert. Ein “letter of consent” bildete die formale Grundlage für das Vorgehen und eine Absicherung für die Studierenden (vergleiche Abbildung 14 auf Seite 217). Meine persönlichen Tagebuchaufzeichnungen resultierten aus den vielen Gesprächen, die während der Arbeit im Projekt geführt wurden. Die E-Mail-Sammlung enthält alle von mir gesendeten und empfangenen E-Mails in diesem Projektkontext.

Im Projekt *hamburg.de* besteht das Material neben einer persönlichen Sammlung von ca. 560 E-Mails aus 14 Protokollen von ganztägigen Sitzungen, mehreren projektbezogenen Fallstudien, Konzepten sowie persönlichen Aufzeichnungen. Die Protokolle wurden regelmäßig von einem der teilnehmenden Kooperationspartner angefertigt und elektronisch an alle Beteiligten verteilt. Meine persönlichen Aufzeichnungen sind in den Sitzungen und zu Gesprächen oder Studien entstanden. Der E-Mail-Verkehr umfasst meine Korrespondenz mit allen Projektbeteiligten. Die Fallstudien wurden im Rahmen des Projektes von mir zur Evaluation und Konzeption von Teilprojekten durchgeführt. Diesem Projekt liegt zwar der Infrastrukturgedanke zu Grunde, das Datenmaterial wurde aber nicht mit diesem Fokus erhoben.

Die Lernplattform *CommSy* haben wir im Team als technische Ergänzung für Lehrveranstaltungen entwickelt. Während der Entwicklung haben wir als Materialien neben Gesprächsprotokollen ca. 450 E-Mails, ca. 20 Interviews, Workshop-Protokolle, Metaplanwände und versionierte Quell-Codes gesammelt. Die gemeinsamen Gesprächsprotokolle sind in Besprechungen der Entwickler entstanden. Der von mir gesammelte E-Mail-Verkehr bezieht sich auf Diskussionen zwischen den Entwicklern und auf die Kommunikation mit den Anwendern sowie auf organisatorische Tätigkeiten. Als Grundlage für die Interviews dienten Leitfäden (Flick 1998, Lamnek 1989), die von uns sukzessive weiterentwickelt wurden. Die Workshops hat das *CommSy*-Kernteam veranstaltet. Sie wurden mit dem weiteren Kreis von Entwicklern und Anwendern durchgeführt und durch Aufzeichnung, Tafelbilder, Protokolle und Metaplankarten dokumentiert. Metaplanwände sind in verschiedenen spontanen und geplanten Entwurfssitzungen entstanden. Kurz nach Fertigstellung des ersten lauffähigen Prototypen haben wir den Quell-Code komplett in einer Versionsverwaltung gelagert, wodurch

sich viele Veränderungen detailgenau nachvollziehen lassen. Bei der Erhebung des Materials stand für uns die Evaluation der Angemessenheit von *CommSy* als Lernplattform im Vordergrund.

#### 4.1.2 Methodischer Ansatz zur Auswertung

Wer sich der Praxis hingibt ohne Wissenschaft, ist wie  
der Steuermann, der ein Schiff ohne Ruder und Kom-  
pass besteigt und nie weiß, wohin er fährt.

— Leonardo da Vinci

Die als Fallstudien ausgewählten Projekte *Internationale Frauenuniversität*, *hamburg.de* und *CommSy* haben zeitlich überlappend stattgefunden. Sie waren alle geprägt durch eine intensive Arbeit unter zeitlich engen Bedingungen. Innerhalb der drei Projekte habe ich verwandte Schwierigkeiten erlebt, die ich bei anderen Projekten in derselben Art so nicht wahrgenommen habe. Dies hat mich bei der Auswahl des Forschungsschwerpunktes für diese Arbeit geleitet. Keines der Projekte wurde explizit als Infrastruktur-Projekt betrieben. Rückblickend betrachtet, war ich jeweils als Betroffener in der Situation, Infrastruktur zu entwickeln. Aber erst im Nachhinein betrachte ich diese Projekte unter dem Gesichtspunkt „Infrastruktur“. Für die methodische Fundierung bedeutet dies, dass ich erst nachträglich das vorhandene Material unter diesem Gesichtspunkt reflektieren konnte.

Um diese Situation und mein Vorgehen abzusichern, habe ich nach Methoden gesucht, die es erlauben, das vorliegende Material der Projekte systematisch auszuwerten. Da in der Informatik begleitende und nachträgliche Projektauswertungen häufig anzutreffen sind, gibt es bereits andere Autoren, die hierfür Herangehensweisen beschrieben haben. Lars Mathiassen bietet dafür *Reflection in Action* an, welche sich durch den Wechsel von Projektteilnahme („Action“) und Phasen der kritischen Reflexion, z. B. durch Tagebucheinträge („Reflection“), auszeichnet (Mathiassen 1998). Die von Glaser und anderen beschriebene *gegenstandsbegründete Theorie* (engl. „grounded theory“) ermöglicht es mir, die vorliegenden Materialien durch inhaltsanalytische Verfahren auszuwerten (Glaser und Strauss 1967, Strauss und Corbin 1996). Das oben aufgeführte empirische Material bietet dafür eine ausreichende Grundlage sowohl in Bezug auf die verschiedenen Formen als auch die Überdeckung des Projektverlaufs.

Meine Theorie bilde ich methodisch kontrolliert in direkter Auseinandersetzung mit dem Forschungsgegenstand (Flick et al. 1991, Flick 1998). Einzelfallbeschreibungen bilden den Ausgangspunkt für die hier durchgeführte Theoriebildung. Ich gehe dazu jeweils von einem *Vorfall* aus, der in Form eines *Schnipsels* – einer als Fragment ausgewählten Aussage – motiviert wird. Um diesen Vorfall werden die relevanten Aktanten in Form von Personen, Institutionen, Einrichtungen und Technik erkundet und beschrieben. Dieses Vorgehen ist der im Web-Modell von Rob Kling beschriebenen Vorgehensweise entliehen und fächert das Geschehen und die Beziehungen auf (Abschnitt 3.3.1). Im Gegensatz zum Web-Modell, welches den Terminus "*situation*" verwendet, wurde hier der Schwerpunkt auf einen *Vorfall* gesetzt, um auszudrücken, dass ausschlaggebend für die Beschäftigung mit der Situation ein Vorfall war, der über die normalen Geschehnisse hinausging. Als Terminologie verwende ich die im Gegensatz zum Web-Modell stärker elaborierte Begrifflichkeit der Akteur-Netzwerk-Theorie (Abschnitt 3.3.3). Damit stehen mir u. a. Möglichkeiten zur Verfügung, aufgebaute Netze zu vergrößern und zu verfeinern. Die Infrastruktur-Kriterien von Star und Ruhleder (Abschnitt 3.3.2) erlauben einen dem Fokus der Arbeit angemessenen analytischen Zugang. Auf derselben Ebene finden die Eigenschaften der großen technischen Systeme ihre Anwendung (Abschnitt 2.2.1).

Die im Anschluss an jeden Vorfall herausgearbeiteten Eigenschaften und Phänomene dienen dabei als Interpretationshilfe bei der Konstruktion gemeinsamer Eigenschaften der Infrastrukturen. Diese gemeinsamen Eigenschaften werden theoretisch motiviert vor das erhobene Datenmaterial gehalten und können damit gegenstands begründet revidiert werden. Ziel der Analyse ist also die Erarbeitung zentraler Eigenschaften von Infrastrukturen durch Zusammenfassung ähnlicher Eigenschaften und ihre Einbettung in einen angepassten Theorierahmen (Mayring 1983, Mayring 1990).

Es handelt sich bei der Darstellung der empirischen Projekte um eine subjektive Perspektive. Gewählt wurde eine Trennung der Darstellung in zwei Teile. In den zentralen Abschnitten sind Vorfälle das kompakte Darstellungsmittel jeweils eines Geschehnisses. Im Anhang wird ein Gesamtüberblick des jeweils referenzierten Projektes präsentiert. Der Leserschaft ermöglicht diese ausführliche Darstellung, die punktuellen Geschehnisse im Kontext nachzuvollziehen. Um die Rahmenbedingungen zu explizieren, habe ich deshalb die folgende Form der Beschreibung für die Anhänge ge-

wählt: Jede Fallstudie beginnt mit einer allgemeinen Projektbeschreibung und nennt die beteiligten Akteure. Die konkrete Herausforderung des Projektes, das Projektgeschehen und meine Rolle in diesem Projekt bilden den Hauptteil jeder Beschreibung. Die konkreten Anforderungen, Entwurfsentscheidungen und die entstandenen Probleme werden explizit aufgeführt. Entsprechend Klings Feststellung “Any system analysis must characterize the computer-based technologies, the social setting where they are used, and the social forces which shape their use” (Kling 1992, S. 3) werden diese drei Felder beleuchtet.

#### 4.1.3 Weiteres Vorgehen zur Auswertung

Im Folgenden werden die drei ausgewählten Projekte unter dem Gesichtspunkt „Infrastruktur“ ausgewertet. Die Projekte und die herausgegriffenen Vorfälle beleuchten, wenngleich fast alle Themen in allen Projekten auftraten, unterschiedliche Problemschwerpunkte unterschiedlich intensiv und dienen somit als Stellvertreter. Das Projekt *Internationale Frauenuniversität* wird aus der Perspektive der Hard- und Software-Bereitstellung sowie der Einbettung in die zur Verfügung gestellten Räume betrachtet. Damit werden nicht die Inhalte der Lehre betont, sondern die Vernetzung der Lehrenden und Lernenden mit ihren anderen Arbeitskontexten. Gleichermaßen bedeutet dies eine Fokussierung auf die technische und organisatorische Einbettung des Projektes am Veranstaltungsort. Beim *hamburg.de*-Projekt bedeutet der Gesichtspunkt „Infrastruktur“ in meiner Auswertung einen Schwerpunkt bei der Entwicklung und Bereitstellung der Infrastruktur für eine große Benutzergruppe. Die Nutzung durch Bürger werde ich in diesem Rahmen nicht betrachten. Das *CommSy* aus der Infrastruktur-Perspektive zu betrachten, ist insofern ungewöhnlich, als dass es sich um eine web-basierte Lernumgebung handelt. Sie wird hier von mir als ein Infrastruktur-Bestandteil in einer Lehr- und Lern-Infrastruktur interpretiert.

Mein Erkenntnisinteresse ist es herauszufinden, welche Software-Entwicklungsprozesse jeweils stattgefunden haben und in welchem Kontext sie abgelaufen sind. Ich möchte die jeweiligen Anlässe und das damit verbundene Umfeld untersuchen. Aus diesem Beobachten möchte ich Schlüsse für die Software-Entwicklung ziehen können. Zum einen soll mir die Beschäftigung mit den Projekten Hinweise geben, welche typischen Phäno-



mene auftreten, und zum anderen soll dies eine Einschätzung erlauben, mit welchen Ad-hoc-Prozessen sich die Beteiligten in der Situation behelfen.

## 4.2 Das Infrastrukturprojekt *Internationale Frauenuniversität*

Er haderte mit sich, bis er sich schließlich sagte, es sei eigentlich ganz normal, daß er nicht wisse, was er wolle. Man kann nie wissen, was man wollen soll, weil man nur ein Leben hat, das man weder mit früheren Leben vergleichen noch in späteren korrigieren kann. — Milan Kundera

Die *Internationale Frauenuniversität* war ein postgraduiertes Universitäts-Projekt (ifu 2000a), welches als Teilprojekt der Expo 2000 in mehreren deutschen Städten stattfand (Expo 2000). Dessen Projektbereich *Information* (ifu 2000b) war in Hamburg angesiedelt. Die Veranstalterinnen wollten an dem Forschungsthema „soziale Konstruktion von Information“ arbeiten, was durch das Thema “information as a social resource” ausgedrückt wurde. Meine Aufgabe in diesem Projekt war das Einrichten einer *computer infrastructure*, die die notwendigen Voraussetzungen für eine wissenschaftliche Arbeit auf hohem Niveau mit “information” während der dreimonatigen Präsenzphase in Hamburg erlaubte. Die dafür geschaffene Position wurde *technology advisor* genannt. Eine zentrale Anforderung dabei war, dass diese Infrastruktur in wesentlichen Teilen nach der Präsenzphase weiterhin zur Verfügung steht, um begonnene Arbeiten fortzusetzen, von ferne auf die Arbeit zuzugreifen und über die Infrastruktur zu kommunizieren. Eine ausführliche Darstellung des Projektes findet sich im Anhang A.

Um die *computer infrastructure* zu errichten, habe ich es vorgezogen, die bereits im Arbeitsbereich Softwaretechnik (SWT) des Fachbereichs Informatik an der Universität Hamburg verfügbare und erprobte Infrastruktur in großen Teilen zu kopieren. Dadurch konnte ich mir sicher sein, dass die Infrastruktur für wissenschaftliches Arbeiten geeignet ist und gleichzeitig die Zusammenstellung von Software und Hardware funktioniert. Zum Aufbau der neuen Infrastruktur wurden Geräte angeschafft und für einen begrenzten Zeitraum Geräte von anderen Fachbereichen am Hauptcampus kostenlos überlassen. Die gesamten Arbeitsplätze der Studentinnen und

Lehrenden waren in Räumen regulärer Fachbereiche der Universität angesiedelt und für diese Zeit inklusive der Ausstattung geliehen. Als Software kamen für Universitäten freie sowie von der Universität Hamburg bereits angeschaffte Produkte zum Einsatz. Die Infrastruktur für den Projektbereich *Information* bestand also aus der Hard- und Software, die verteilt über den Hauptcampus der Universität Hamburg zur Verfügung stand, den Räumen, den zuständigen Personen und damit verbundenen Dienstleistungen.

Zielgruppen für die aufgebaute Infrastruktur waren neben den Mitarbeiterinnen der *ifu*, die Planungs- und Verwaltungsaufgaben übernommen hatten, die Lehrenden, Gastdozentinnen und die Studentinnen. Diese nutzten die Infrastruktur intensiv für die Alltagsarbeit und für die Projektarbeit innerhalb der Präsenzzeit von 3 Monaten. Teile der Infrastruktur waren an den Arbeitsbereich SWT angebunden, aber von diesem räumlich getrennt.

Die exemplarisch ausgewählten Vorfälle werden in chronologischer Reihenfolge präsentiert. Im ersten Vorfall wird die Problematik thematisiert, die durch die Distanz zwischen Server und Clients hervorgerufen wurde. Der zweite Vorfall geht auf eine Entscheidung zurück, eine neuere Version in der endgültigen Installation zu verwenden. Die Entwicklungsarbeiten an einem anderen Bestandteil der Infrastruktur und ihre Auswirkungen sind Thema im dritten Vorfall. Die im vierten Abschnitt zusammengestellten Vorfälle gehen auf die kulturellen Besonderheiten dieses Projektes ein.

#### 4.2.1 Unerklärliches Verhalten: Der Server wird gebootet

Date: Tue, 25 Jan 2000 10:17:30 +0100  
Subject: Computerprobleme

Guten Morgen, Wolf-Gideon,  
long time no see...  
1. seit letzter Woche Donnerstag stürzt mein PC mehrmals täglich ab, in der Regel bei normaler Wordanwendung, ein oder zweimal ist es auch passiert während ich nur das Netscape Fenster geöffnet hatte. Ich glaube auch [...] hatte das Problem. [...]  
Ok, das waren die wichtigsten Punkte. Bis die nächsten Tage dann, [...]

(Beleg E-Mail 25-01-2000 tb)

##### *Vorfall*

Die Vorbereitungen für die *ifu* begannen bereits im Herbst des Jahres 1999. Nachdem das *ifu*-Büro von seinem vorläufigen Standort, dem Informatik-

Campus, auf den Hauptcampus umgezogen war, nutzten die Mitarbeiterinnen immer noch den Server in der Informatik als Ort zur Speicherung für ihre Dokumente und als Basis zur Kommunikation. Dies wurde durch das campusweite Netz ermöglicht, das vom Regionalen Rechenzentrum (RRZ) betrieben wird und IP-basierte Verbindungen unterstützt. Das Netz bietet twisted-pair 100-MBit-Anschlüsse für die Arbeitsplätze in den Universitätsräumen und einen Backbone mit 2 GBit zwischen den Standorten. Dadurch wurde für die Mitarbeiterinnen kein Unterschied in Bezug auf Geschwindigkeit im Zugriff auf den Server erkennbar. Das RRZ ist verantwortlich für die Bereitstellung aller Netzdienste auf dem Hauptcampus. Es stellt darüber hinaus Konten, Datenspeicher und Basisdienstleistungen zur Verfügung. Einige Fachbereiche, darunter die Informatik, betreiben Netz und Computer selbst, um speziellen Anforderungen gerecht zu werden.

Der Administrator des Arbeitsbereichs SWT startete den Server üblicherweise montags morgens (vor 7 Uhr) neu, um während der Woche Abstürze zu vermeiden, da die Erfahrung gezeigt hatte, dass der Server mindestens 7 Tage problemlos funktionierte. Darüber hinaus musste der Computer von Zeit zu Zeit neu gestartet werden, wenn neue Software auf dem Server installiert wurde. Für diese Fälle hatte sich eine gemeinsame Praxis herausgebildet: Eine kurze Runde auf dem Flur wurde benutzt, um allen anwesenden Mitarbeitern mitzuteilen, dass der Server für einige Minuten nicht verfügbar sein würde. Diese speicherten zuerst alle noch offenen Dateien und passten ihre Arbeit entsprechend an.

Diese Praxis funktionierte allerdings nicht für das *ifu*-Büro auf dem Hauptcampus, da die Information über einen Neustart des Servers dort nicht ankam. Die Mitarbeiterinnen arbeiteten häufig schon vor 7 Uhr und erfuhren wiederholt Situationen, in denen sie sich nicht anmelden konnten oder „nur sehr langsamen Zugriff auf E-Mail und das Web“ hatten. Weder erkannte der Administrator, dass sie zur selben Zeit den Server benutzten, noch besaßen sie die Möglichkeit, den Grund für ihr Problem herauszufinden. Obwohl der Administrator des Arbeitsbereichs die Zusage gegeben hatte, dass der Server von der *ifu* mitbenutzt werden könne, konnte er diese Zusage nicht auf die obigen Situationen übertragen. Die Folge war, dass die Mitarbeiterinnen ihren Arbeitsstationen eine geringe Zuverlässigkeit zuschrieben.

### *Reflexion*

Die in diesem Vorfall beteiligten Akteure sind neben den Mitarbeitern und dem Administrator des Arbeitsbereichs die Mitarbeiterinnen des *ifu*-Büros. Zu den Aktanten zählen der Server, die Arbeitsstationen, die auf beiden betriebene Software und das diese miteinander verbindende Netz. Die Verpflichtung des Administrators, des Systemerbauers und -betreibers, gegenüber den betroffenen Personen, den Benutzern des Systems, bei einem planbaren kurzzeitigen Ausfall des Servers besteht in einer rechtzeitigen Information. Diese Verpflichtung wird über die räumliche Nähe zwischen Mitarbeitern und Administrator aufrechterhalten. Durch den Umzug der Mitarbeiterinnen an den Hauptcampus geriet die eingegangene Verpflichtung ihnen gegenüber in Vergessenheit.

Auf einer allgemeineren Ebene betrachtet lässt sich feststellen: Weder lieferte der Server Informationen darüber, welche Personen mit ihm gearbeitet hatten und dies verbunden mit einem stimmigen Mittel, um diese über die kommende Aktion zu informieren, noch verhielt sich die Software auf den Arbeitsstationen verständlich. Die Eigenschaft der Software, diese Informationen über sich zur Verfügung zu stellen, möchte ich *Selbstbeschreibungsfähigkeit* nennen, und die Möglichkeit, mit Werkzeugen diese Informationen zu bearbeiten, *Wartbarkeit*.

Aus der Perspektive der Software-Entwicklung führt diese Betrachtung zu zwei relevanten Fragen: erstens der Frage nach angemessenen Installations- und Update-Verfahren für Server-basierte Software im Rahmen von Innovation, Wachstum und Konsolidierung und zweitens nach einer angemessenen Information von Benutzern verteilter Infrastrukturen. Mit Bezug auf das erste sollte eine Software-Infrastruktur Mittel zur Verfügung stellen, Bestandteile während der Nutzung auszutauschen. Der zweite Punkt erfordert es, mithilfe der Infrastruktur über die Infrastruktur entsprechende Informationen den Endanwendern zukommen zu lassen.

## **4.2.2 Langsames Netz: Unnötiger Dateitransfer**

Date: Sat, 5 Aug 2000 04:59:52 -0500 (CDT)  
Subject: 118 slow to dying

Hi Wolfie,  
I am working in RRZ 118. The system slows down badly here. Last night with everyone working, it got pathetic, evidently. One woman lost several hours of work. She

tried to save her work and it simply disappeared. They were all here quite late, and the student was upset.

Right now ComSy is very slow and Netscape is slow, but Telnet is very fast. I do not know why that is.

Just keeping you informed!

*(Beleg E-Mail 05-08-2000 mw)*

### *Vorfall*

Bereits zu Anfang der Infrastrukturentwicklung wurde entschieden, dass Server-basierte Profile genutzt werden. D. h. die Benutzereinstellungen wurden auf dem Server abgelegt, um sie an jedem Arbeitsplatz zur Verfügung zu stellen. Windows NT stellt dafür einen Mechanismus zur Verfügung, der bei jeder Anmeldung die Einstellungen vom Server auf die Arbeitsstation überträgt und beim Abmelden diese (geändert) wieder zurückkopiert. Die Zusammenstellung von Software, die auf den Arbeitsstationen bereitgestellt werden sollte, war bereits einige Wochen vor der Präsenzphase ausführlich getestet worden, um sicherzustellen, dass sie problemlos mit den Arbeitsstationen und dem Server zusammenarbeitete. Zu Beginn der Präsenzphase sollte diese Zusammenstellung auf alle Arbeitsstationen verteilt werden. Allerdings wurde als eine Entscheidung in letzter Minute anstelle der getesteten Version 4.0 die neue Version 5.0 des Microsoft Internet Explorers für die Zusammenstellung ausgewählt.

Diese Änderung führte zu einer großen Anzahl von Dateien, die regelmäßig über das Netz übertragen wurden und damit eine unnötige hohe Last auf den Server legten. Die alte Version des Internet Explorers speicherte die temporären Internet-Dateien (den lokalen Cache) in einem lokalen Systemverzeichnis, das für alle Anwender gleich war. Die neue Version hingegen speicherte diesen Cache benutzerbezogen im Profil jedes einzelnen Anwenders. Mit einer Standardeinstellung von ca. 25 MByte musste jedes Mal beim Anmelden und beim Abmelden diese Menge an temporären Dateien nutzlos übertragen werden. Darüber hinaus hatte sich bei den Studentinnen eine Rechnernutzung herausgebildet, bei der alle annähernd zur selben Zeit die Arbeitsstationen benutzten, sodass durch deren gleichzeitige Anmeldung große Dateimengen vom Server angefordert wurden. Da die Übertragung der nutzlosen Dateien zu fast allen Arbeitsstationen zur selben Zeit stattfand, brach der Server unter der Last beinahe zusammen.

Die Ursache für dieses Problem war schwer zu identifizieren, da erstens die Zusammenstellung zuvor problemlos funktioniert hatte und zweitens das Netz zwischen Arbeitsstationen und Server komplex und weit verteilt war. Zuerst beklagten sich einige Studentinnen über ein langsames Netz bei ihren Kursleiterinnen, aber der Grund konnte nicht ermittelt werden. Dann wurde vermehrt berichtet, dass das An- und Abmelden sehr lange dauerte. Dies lenkte die Aufmerksamkeit auf die Profile, in deren Speicher-verbrauch die Ursache für das Problem identifiziert wurde. Als Verursacher des Speicherverbrauchs konnte schnell die neue Version des Internet Explorers ausgemacht werden. Als erste Maßnahme wurden die temporären Dateien im Profilverzeichnis auf dem Server für alle Konten gelöscht. Im zweiten Schritt wurde über direkte Gespräche und E-Mail-Kommunikation versucht, die Studentinnen aufzufordern, die Größe des Internet Explorer Caches auf null zu setzen. Der dritte Schritt bestand darin, Studentinnen, die persönlich bei der Rechnernutzung angetroffen wurden, zu bitten, in diesem Moment die Einstellungen zu ändern.

### *Reflexion*

Wegen der Komplexität des hier beschriebenen Akteur-Netzes im Kontext der genannten Software-Infrastruktur war es besonders schwierig, die Ursache des Problems zu ermitteln. Das Netz besteht u. a. aus den Studentinnen, den Lehrenden, den Arbeitsstationen, der auf den Arbeitsstationen verwendeten Software, dem Netzwerk, dem Server und der auf dem Server verwendeten Software. Aus Sicht des Systemerbauers, des *ifu*-Administrators, war die Zusammenstellung der Software für die Arbeitsstationen eine Kapsel, die mit der anderen Software problemlos zusammenarbeitete. Erst das Öffnen der Kapsel offenbarte die Ursache. In diesem Fall reiste ein unveränderliches Bewegliches von Redmond, USA, nach Hamburg zur *ifu*. Eine Änderung in den Entwurfsentscheidungen dieses Artefaktes wirkte sich auf die zuvor ausgewogene Zusammenstellung der Software für die Arbeitsstationen und den Server aus, obwohl es so schien, als ob einzelne Produktbestandteile durch neuere Versionen ersetzt werden könnten. Die Browser-Software entwickelte sich zu einem kritischen Problem, indem sie die Auslastung des Servers an sein Maximum trieb. Dadurch wurde der Server zu einem zurückgefallenen Teil.

In Bezug auf die *Entwicklungsfähigkeit* der Software-Infrastruktur wurde ihr durch die Entwurfsentscheidung der Server-basierten Profile zusam-

men mit der vorhandenen Netzwerk-Technologie Grenzen gesetzt, die das Volumen an Daten im persönlichen Profil betrafen. Diese implizite Grenze wurde sowohl individuell als auch bezogen auf alle Teilnehmenden überschritten. Gleichzeitig war die *Konfigurierbarkeit* der Infrastruktur in Bezug auf die Profilnutzung der betroffenen Software nicht leicht erschließbar. Obwohl es höchstwahrscheinlich möglich war, die Nutzung des Profilverzeichnis auszuschalten, war es dennoch schwer erschließbar. Man kann von einer *Störung* sprechen, die zwischen der Entwicklung der Software-Infrastruktur für die *ifu* und der Entwicklung des Produktes Internet Explorer stattgefunden hat.

Dieses Beispiel ist darüber hinaus aus der Perspektive des Continuous Software Engineering interessant, da es einen Effekt illustriert, der typischerweise beim Aktualisieren von Software auftritt. Damit werden Fragen bezüglich der Änderungen von Entwurfsentscheidungen bei neuen Software-Versionen genauso aufgeworfen wie auch bezüglich der Integration von Veränderungen mit massiven Auswirkungen auf abgestimmte und laufende Infrastrukturen.

#### 4.2.3 E-Mail-Aussperrung: Aktivierung der Firewall

Am 9. August 2000 meldeten sich Studentinnen und Lehrende, dass sie keine E-Mail mehr verschicken konnten. Ein Test ergab, dass es nicht mehr möglich war, eine Verbindung zum Server in Berlin über SMTP aufzubauen, obwohl dessen andere Dienste (SSH, HTTP, POP3, IMAP4 usw.) weiterhin funktionierten. Eine erste Vermutung war, dass dessen Message Transfer Agent (MTA) ausgefallen war.

(*Beleg E-Mail 09-08-2000 hp*)

##### *Vorfall*

Als Teil einer Identitätsbildung wurde jeder *ifu*-Studentin eine persönliche E-Mail-Adresse gegeben. Um dies zu realisieren, wurde in einem angeschlossenen Forschungsprojekt ein spezieller Server aufgesetzt, der für alle Projektbereiche der *ifu* verschiedene Kommunikationsmittel unterstützte (E-Mail, Mailinglisten, Web-Mail, Web-basierte Diskussionsforen). Dieser Server war örtlich in Berlin an der Humboldt-Universität angesiedelt, etwa 300 km entfernt vom Hamburger Campus. Der Server stellte Mailboxen zur Verfügung, die über die Protokolle POP3 und IMAP4 abgefragt werden konnten und der Server ermöglichte das Verschicken von E-Mails per SMTP. Zur Erleichterung wurden die entsprechenden Einstellungen für die

Hamburger Studentinnen bereits im Rahmen der Konfiguration der lokalen Software-Infrastruktur voreingestellt.

Während des Sommers – der vorlesungsfreien Zeit der Universität Hamburg – setzte das RRZ seine neue Strategie um, mit der Spam-E-Mail und anderer E-Mail-Missbrauch verhindert werden sollte. Im Detail erlaubte diese neue Strategie das Versenden von E-Mail von Arbeitsstationen innerhalb des Universitätsnetzes ausschließlich über den universitätseigenen E-Mail-Server. Darüber hinaus legte sie fest, dass kein externer SMTP-Server aus dem Universitätsnetz erreichbar war, damit sichergestellt werden konnte, dass keine unsichere Arbeitsstation für Spam-E-Mail missbraucht werden konnte. Diese starre Strategie blockierte die Nutzung des E-Mail-Servers in Berlin vollständig und somit das Versenden von E-Mail von den Hamburger Arbeitsstationen.

Das RRZ verfolgte die Absicht, seine Software-Infrastruktur sowohl von innen zu sichern als auch nach außen abzuschirmen. Dieser Entwicklungsprozess wurde einer allgemeinen Öffentlichkeit nicht bekannt gegeben. Die Mitarbeiter des RRZ gingen davon aus, dass die Personen auf dem Campus der Universität ihre E-Mail mit dem lokalen SMTP-Server verschicken und keine Änderung bemerken würden.

### *Reflexion*

Der Vorfall zeigt, wie die Infrastruktur für die *ifu* implizit auf der Infrastruktur der Universität Hamburg aufsetzte (“built on an installed base”, s. S. 53) und dabei die darunter liegende Infrastruktur im Moment des Nicht-Funktionierens in Erscheinung tritt (“becomes visible upon breakdown”, ebd.). Aus Sicht der Administration der *ifu*-Software-Infrastruktur war das campusweite Netz eine Kapsel, die benutzt werden konnte, um einen externen E-Mail-Server zu erreichen bzw. irgendeine Art von Daten über das IP-Protokoll zu übertragen. Man ging davon aus, mit der Nutzung des Netzes auch die Zusage darüber zu haben, dass jede Art von Daten übertragen werden konnte. Diese Zusage wurde nun nicht mehr eingehalten. Die Innovation des RRZ, eine Firewall einzuführen, veränderte die Beziehung der beiden Akteure, *ifu* und RRZ.

Es war unrealistisch, als Reaktion auf die Aktivierung der Firewall die Einstellungen jeder einzelnen Studentin per Fernadministration oder persönlich zu ändern, da die Installation dieser Software eine eigene Triebkraft entwickelt hatte. Deshalb begann man mit dem RRZ in Verhandlung



zu treten, um die Mitarbeiter zu überzeugen, die Abschottung bis zum Ende der *ifu*-Präsenzphase aufzuschieben. Das Interesse der Mitarbeiter war es, der *ifu* gegenüber unterstützend dazustehen, das Interesse der *ifu*-Administratoren war es, so wenig Arbeit wie möglich aufzuwenden. Der Aushandlungsprozess wurde aufgrund der Abwägung von Arbeitsaufwänden entschieden, da es unmöglich gewesen wäre, die Einstellungen jeder Benutzerin zentral zu ändern. Die Art der Speicherung war im Detail nicht dokumentiert. Dies ist ein weiteres Beispiel für die notwendige Eigenschaft *Konfigurierbarkeit* einer Software-Infrastruktur.

Das Beispiel unterstreicht, dass in eine Infrastruktur Annahmen eingeschrieben sind. Nur einige dieser Annahmen sind dokumentiert. Entscheidend ist es herauszufinden, welches die relevanten und wichtigen Annahmen sind und wie diese aufgedeckt werden können. Ergänzend ist für einen Software-Entwicklungsprozess interessant, wie sich diese Annahmen über die Zeit entwickeln.

#### **4.2.4 Chinesisches Windows nötig: Kulturelle Unterschiede**

In der Vorbereitungswoche vor der Präsenzphase der *ifu* wurde der Administrator von einer Lehrenden angesprochen, ob es möglich wäre, ein chinesisches Windows zu installieren, da die von ihr mitgebrachte Software ausschließlich unter dieser Version von Windows laufe.

*Beleg fn 29. Juni 2000*

##### *Vorfall*

Da die *ifu* ein internationales Studienprojekt war, kamen die Studentinnen aus den unterschiedlichsten Ländern der Erde. Die Studiensprache der *ifu* war Englisch. Deshalb wurde versucht, alle verwendete Software in einer englischen Version zur Verfügung zu stellen.

Bereits dies stellte eine Schwierigkeit dar, da es mit den vorhandenen Lizenzen meist nur möglich war, die europäisch-englische Version (Großbritannien) zu bekommen, die sich in der Wahl einiger Worte (z. B. "Trash" vs. "Recycle Bin") signifikant unterschied. Anleitungen und andere Texte, die in diesem Rahmen angefertigt wurden, mussten darauf Rücksicht nehmen. Für die Hardware-ansteuernde Software (z. B. Druckertreiber, Scanner-Treiber) war allein die Auswahl eines englischen Druckertreibers keine ausreichende Lösung, da sich die Meldungen des Druckertreibers auf Schalter, Knöpfe und Schubladen bezogen, die bei den deutschen Druckern anders beschriftet waren.

Die verwendete Hardware stammte aus der Standardausstattung der Universität Hamburg und war somit für den deutschen Markt gefertigt. Sowohl die Bedienknöpfe des Computers als auch des Monitors waren meistens mit deutschen Texten beschriftet. Die Tastaturen besaßen Umlaute und hatten – was schwerer wog – das „y“ und „z“ vertauscht sowie eine andere Anordnung der Sonderzeichen.

Bei den Verbrauchsmaterialien war insbesondere das DIN-A4-Papier ein mehrschichtiges Problem. Erstens war den wenigsten Studierenden und Lehrenden bekannt, dass in Deutschland ausschließlich DIN-A4-Papier verwendet wird. Zweitens zogen sie nicht die notwendige Konsequenz und formatierten ihre Dokumente entsprechend. Drittens bot z. B. die verwendete englische Version von Microsoft Word nicht die Option, Letter-formatierte Texte automatisch auf DIN A4 anzupassen, was in der deutschen Version mittlerweile standardmäßig eingeschaltet ist. Und schließlich zeigten die Drucker eine deutschsprachige Fehlermeldung mit der Aufforderung, (nicht vorhandenes) Letter-Papier einzulegen, die von ausländischen Studierenden nicht verstanden wurde.

Die von Lehrenden mitgebrachte Software stammte aus verschiedenen Ländern. Einige Programme liefen auf dem bereitgestellten englischen Windows. Für andere war es erforderlich, die Betriebssystemversion für das entsprechende Land bereitzustellen (s. o.). Die notwendigen Einstellungen (z. B. Netzkonfiguration) mussten daraufhin in Zusammenarbeit mit einer der jeweiligen Sprache mächtigen Person erfolgen. Rechner, auf denen diese Software installiert war, standen für den normalen Studienbetrieb zeitweise nicht zur Verfügung.

Eine Anforderung, die von Studierenden herangetragen wurde, war die Möglichkeit, im Browser E-Mails mit nicht-lateinischen Schriftzeichen zu lesen. Die installierte Version des Web-Browsers konnte nur lateinische Schriftzeichen darstellen. Für weitere Schriften sind spezielle Betriebssystemerweiterungen und Browser-Plugins notwendig. Die Schwierigkeit dieser Anforderung bestand darin, dass hierfür auf jedem Rechner eine entsprechende Erweiterung hätte installiert werden müssen. Neben dem Aufwand der Installation wäre ein ausführliches Testen und der dazugehörige Support nahezu unmöglich gewesen, da niemand unter den Administratoren der betreffenden Sprachen mächtig war. Außerdem hätte dies bei mehr als 50 Nationalitäten weitere Wünsche zur Folge gehabt, wobei unklar war, ob sich die verschiedenen Plugins gegenseitig vertragen hätten.

*Reflexion*

Auf den ersten Blick erscheinen „die Studierenden“ „den Administratoren“ gegenüber wie ein Akteur. Durch die (verständlichen) Wünsche bzgl. der Nutzung von Software zur Unterstützung der eigenen Sprache forderten sie einen Aufwand, der aus ihrer Sicht nicht groß ist. Für die Administratoren wird dadurch der eine Akteur „Studierende“ zu vielen Akteuren mit unterschiedlichen Bedürfnissen, deren Aufwand sich multipliziert und seinem Interesse als Systembetreiber nach einer einfachen, geordneten und kontrollierbaren Struktur zuwiderläuft. Die Vorfälle zeigen, wie intensiv sich *kulturell* begründete Anforderungen auf eine Infrastruktur auswirken, insbesondere wenn, wie hier, keine gemeinsame Kultur, z. B. Konsens über Englisch als einzige Sprache, bei der Nutzung der Infrastruktur etabliert wurde, sondern alle Teilnehmenden ihren eigenen kulturellen Hintergrund bei der Rechnernutzung mitbringen. Studierende und Lehrenden stammen aus unterschiedlichen Praxisgemeinschaften und formen im Verlauf der *ifu* eigene Praxisgemeinschaften. Gleichzeitig thematisieren die Vorfälle *unerwartete Anforderungen* bei der Infrastrukturentwicklung, indem sie offen legen, wie Benutzerinnen und Benutzer der Infrastruktur ihren eigenen Bedürfnissen Ausdruck verleihen und Innovation fordern.

**4.3 Das Infrastrukturprojekt *hamburg.de***

I don't even have an e-mail address. I have reached an age where my main purpose is not to receive messages. — Umberto Eco

Das Projekt *hamburg.de* ist ein Stadtentwicklungsprojekt, das von der Stadt Hamburg ins Leben gerufen wurde (Freitag und Ehlers 1999). Hamburger Bürgern und Unternehmen soll damit ein leichter Zugang zum Internet ermöglicht und gleichzeitig die Qualität der Website und die angebotenen Inhalte verbessert werden. Dafür hat die Stadt eine Public-Private-Partnership ins Leben gerufen. Die gegründete Gesellschaft soll frei zugängliche elektronische Dienste bereitstellen. Dabei sollen städtische genauso wie privatwirtschaftliche Dienstleistungen durch dasselbe Webportal personalisiert elektronisch bereitgestellt werden. Um diese Dienste kontextbezogen anzubieten, wurde das Konzept der Lebenslagen (Bleek 2001, Bleek

2002) bei der Entwicklung herangezogen. Meine Rolle in diesem Projekt war die eines externen fachlichen Beraters, der in die regelmäßig tagende technische Steuerungsgruppe integriert war. Meine Aufgabe war es sicherzustellen, dass das entwickelte System alle Voraussetzungen erfüllt, um die Lebenslagenunterstützung nach dem Start anbieten zu können. Während des Entstehungs- und Entwicklungsprozesses veränderte sich aufgrund der Bedürfnisse des Projektes und meines technischen Wissens meine Rolle stärker hin zu einem technischen Berater. Eine ausführliche Darstellung des Projektes findet sich im Anhang B.

Das *hamburg.de*-Projekt ist ein Beispiel für eine Infrastruktur, die einer sehr großen anonymen und offenen Nutzergruppe zur Verfügung gestellt wird. Sie bietet standardisierte Internet-Dienste und über diese hinaus lokale Informationen und Dienstleistungen.

Die ausgewählten vier Vorfälle sind aus der Projekthistorie motiviert. Der erste Vorfall fand direkt nach der Einführung der kostenlosen E-Mail-Adresse statt. Die Entwurfsentscheidungen des zweiten Vorfalls mussten nachträglich zur Einführung von Webmail gefällt werden. Der dritte Vorfall geht auf die Interdependenzen der Infrastrukturbestandteile ein. Schließlich ist die Mandantenfähigkeit Thema des vierten Vorfalls.

#### 4.3.1 Missbrauch von E-Mail-Adressen: Offener Versand

Über „Hamburg.de“ lassen sich mit einfachen Handgriffen von jedem x-Beliebigen E-Mails versenden, die vortäuschen, vom Finanzamt, von der Polizei oder sogar vom Bürgermeister selbst zu kommen. Ortwin.Runde@hamburg.de<sup>1</sup> – kein Problem.

(*Hamburger Abendblatt, Röttger 2000*)

##### *Vorfall*

Die Domain *hamburg.de* wurde von der Stadt Hamburg an eine Betreiber-gesellschaft mit der Verpflichtung vermietet, innerhalb eines festen Zeit-raums eine öffentliche Internet-Plattform zu schaffen. Ein wichtiger Be-standteil der *hamburg.de*-Plattform ist der so genannte Free-E-Mail-Anteil. Darunter fällt eine Web-basierte Registrierung, mit der man eine – in ge-wissen Grenzen frei wählbare – E-Mail-Adresse „@hamburg.de“ erhält.

Neben der Software zur Registrierung gehört hierzu eine Gruppe von E-Mail-Servern, die E-Mails in das Internet verschickt und E-Mails ent-

---

1 Zu diesem Zeitpunkt war Ortwin Runde der Hamburger Bürgermeister.

gegennimmt und in Postfächer einsortiert. Um diesen Teil der Plattform realisieren zu können, bedurfte es Geräte, die den ein- und ausgehenden E-Mail-Verkehr behandeln, sowie einer großen Menge Festplattenspeichers, auf dem die eingehenden E-Mails gespeichert werden können. Die Hardware wurde von einer Betreibergesellschaft angeschafft, die die Geräte wiederum bei einer Betriebsgesellschaft abgestellt hat, wo sie mit dem Internet verbunden wurden. Mitarbeiter beider Firmen haben die Rechner installiert und konfiguriert. Ein weiteres Unternehmen implementierte die Web-Anmeldungs-Software, die den Benutzer in die entsprechenden Datenbanken einträgt, um das E-Mail-Konto gültig zu schalten.

Benutzer der kostenlosen E-Mail-Adresse bekommen nur Namen angeboten, bei denen Vor- und Zuname durch ein Sonderzeichen getrennt sind. Spezielle Adressen (z. B. *funktion@hamburg.de*, *kürzel@hamburg.de*, *nachname@hamburg.de*) können nicht kostenlos reserviert werden. Alle Benutzer können zum Versenden ihrer E-Mail den SMTP-Server von *hamburg.de* benutzen. Dieser wurde eingerichtet, damit Mitarbeiter der Stadt und private Benutzer, die sich über einen Internet-Provider ohne eigenen SMTP-Server einwählen, E-Mails verschicken können.

Technisch und historisch bedingt ist es möglich, mit einem SMTP-Server E-Mails mit jeder Art von Absender zu verschicken. Eine übliche und hier verwendete Einschränkung ist, den Versand nur mit der Absenderkennung der eigenen Domäne zuzulassen. In der Definition des Protokolls ist keine Prüfung des Teils vor dem @-Zeichen vorgesehen. Diese Schwäche wurde im o. g. Zeitungsartikel beschrieben. Der Autor konnte eine E-Mail mit einem Absender verschicken, dessen Eigentümer er nicht war. Nach diesem Vorfall wurde das so genannte SMTP-after-POP eingeführt, welches den Versand von E-Mails von der eigenen IP-Adresse nur zulässt, wenn der gewünschte Absender zuvor E-Mail abgerufen und sich somit authentifiziert hat.

Der Vorfall zeigt, dass selbst bei der Übertragung einer bereits erprobten Zusammenstellung von Software die Erfahrungen aus anderen Einsatzkontexten zu berücksichtigen sind. Gleichzeitig fordert der öffentliche Kontext von den Betreibern mehr Sorgfalt, und die Benutzer stellen höhere Ansprüche. Neben ausführlichen Testläufen hätten gezielte „Angriffe“ eine Liste von Sicherheitsmängeln ergeben, bei denen man einzeln hätte entscheiden können, welche üblich, welche akzeptabel und welche nicht hinnehmbar sind.

*Reflexion*

Für Außenstehende stellt sich das von den Systemerbauern zur Verfügung gestellte Netz als eine Kapsel dar. Diese besaß ein kritisches Problem. Gründe für dieses Problem können beim Auffächern des Netzes erkannt werden. Es zeigt zuerst, dass der Akteur Betreibergesellschaft gegenüber dem Akteur Stadt seine zeitliche Verpflichtung einhalten musste. Das Interesse, keine Konventionalstrafe zu zahlen, wurde umgesetzt in ein Interesse, die Software pünktlich zum Laufen zu bringen. Das kritische Problem wurde in das Netz eingebaut, indem Internet-Standards ohne kritische Hinterfragung in das Netz eingebaut wurden. Das Interesse der lokalen Presse war es, Mängel an der Technik und Vertrauenswürdigkeit der Dienstleistung aufzudecken. Dies gelang durch eine vielfach bekannte Schwäche in der Protokolldefinition. Das Interesse der Stadt und der Betreibergesellschaft, keine negative Berichterstattung zu haben, war hingegen viel größer als das Interesse bzgl. einer Konventionalstrafe, wurde aber zu spät artikuliert.

Der Vorfall zeigt, wie gut *Übertragbarkeit* im Kontext von Infrastrukturen funktioniert. Die gewählte Zusammenstellung von Software arbeitet in vielen Einsatzkontexten. Aufgrund der verwendeten Internet-Standards ist der systeminterne Anschluss weiterer Software genauso möglich wie der unabhängige Einsatz von Software für die Endnutzer. Allerdings zeigt dieses Beispiel, dass trotz *Standardisierung* der Aufbau von simplen Infrastrukturen mit großen Unsicherheiten verbunden ist. Diese können nicht durch die Berufung auf Standards ausgeschlossen werden. Vielmehr ist ein tiefes Verständnis der Standards notwendig, um das unentbehrliche Prototyping und Testen der Infrastruktur sinnvoll auszuüben. Drittens ist an dem Vorfall erkennbar, wie *unerwartete Anforderungen* auf Infrastruktur-Entwicklung zukommen. Die Anforderung, nur mit berechtigtem Absender E-Mail-Versand zuzulassen, wurde vorher nicht antizipiert und musste im laufenden Betrieb der Infrastruktur im Rahmen einer Innovation und Konsolidierung eingebaut werden.

### 4.3.2 Nachträgliche Einführung von Webmail: Wechselnde Anforderungen

Ein Gespräch mit der [Firma] hat ergeben, dass die Datenbanklösung auf dem bestehenden System nicht realisierbar ist (Grund: fehlender Speicherplatz).

*(Beleg Protokoll vom 23. Juni 2001)*

#### *Vorfall*

Die Bereitstellung von Webmail war eine logische Konsequenz aus der Bereitstellung von kostenloser E-Mail, da eine große Anzahl von Anbietern kostenloser E-Mail ebenfalls Webmail zur Verfügung stellten. Allerdings wurde dies bei der Errichtung der E-Mail-Server zuerst nicht berücksichtigt, sodass nicht schon bei deren Konzeption Vorbereitungen getroffen werden konnten. Da die E-Mail-Plattform auf frei verfügbaren Produkten mit einer Vielzahl von Standards aufsetzt, war der erste Ansatz, nach Produkten zu suchen, die auf Grundlage dieser Standards Webmail realisierten.

Das Angebot des Herstellers des verwendeten Content-Management-Systems setzte auf dem POP3-Standard auf. Damit wurden alle eingegangenen E-Mails abgeholt und in einer Datenbank abgelegt, um sie effizient im Web zu visualisieren. Diese Lösung besaß den Nachteil, dass die in die Datenbank übertragenen E-Mails nicht mehr für andere Programme des Endanwenders zur Verfügung standen. Ein weiterer Nachteil dieses Ansatzes war es, dass die Datenbank ein Speichervolumen benötigte, das etwa dem der E-Mail-Plattform entsprach. Dies war im Entwurf nicht eingeplant worden.

Das Angebot des Datenbankherstellers war eine Lösung, bei der alle E-Mails ausschließlich in der Datenbank gespeichert würden. Das Programm für eingehende E-Mails würde durch ihres ersetzt, das sich an den Standard hält. Die Prozesse für E-Mail-Abfragen würden ebenfalls durch ihre ersetzt und böten die Protokolle POP3 und IMAP4. Auf der Grundlage des IMAP4-Protokolls würde eine Webmail-Applikation laufen, die somit auf dieselbe Mailbox wie der Anwender zugreift. Nachteil dieser Lösung ist, dass als Speichermedium für die E-Mails ausschließlich die Datenbank verwendet wird. Eine Migration ist besonders aufwändig, und die existierende Hardware passte nicht für diese Lösung. Hinzu kommt der Nachteil, dass aktualisierte Versionen nur noch von einem Hersteller geliefert wer-

den können im Gegensatz zu frei verfügbarer Software, bei der eine ganze Community Fehlerkorrekturen diskutiert und bereitstellt.

Schließlich wurde zur Realisierung ein Weg gewählt, bei dem die existierende Infrastruktur nicht verändert werden musste. Ein IMAP-Server wurde dem E-Mail-Cluster hinzugefügt, sodass über dieses Protokoll intern auf die Mailboxen zugegriffen werden konnte. Im existierenden Content-Management-System wurde daraufhin eine Anwendung in Eigenentwicklung erstellt, die auf dieser Schnittstelle Webmail zur Verfügung stellte.

### *Reflexion*

Das am Anfang von den Systemerbauern zusammengestellte Netz aus technischen Geräten besitzt eine Triebkraft, indem die zu Grunde gelegten Entscheidungen z. B. über angeschaffte Geräte und ausgewählte Standards nur schwer revidierbar sind. Dieser Vorfall unterstreicht, wie sehr bei der Entwicklung einer Infrastruktur die *Gegebenheiten* eine Rolle spielen. Sie wirken mit Beharrlichkeit und engen somit Spielräume ein. Trotzdem wurde die *Entwicklungsfähigkeit* der vorliegenden Infrastruktur dadurch sichergestellt, dass im Rahmen einer Innovation erneut Standards zum Einsatz kamen. Mithilfe von *Standardisierung* (Format der Mailboxen) konnte sichergestellt werden, dass eine Vielzahl von Produkten existiert, die darauf arbeiten kann. Hier wurde im Wachstum, Einführung von Web-Mail, wiederum ein Produkt gewählt, das Standards genügt, um Teile des Systems zu koppeln.

### **4.3.3 Webserver zu langsam: Cache funktioniert nicht**

„Um 12 Uhr mittags sind wir immer am langsamsten, da ist der meiste Zugriff.“

(Beleg fn August 2001)

### *Vorfall*

Zentraler Bestandteil der Internet-Plattform *hamburg.de* ist das Web-Portal, auf dem aktuelle Informationen, Webmail und stadtbezogene Applikationen laufen. Einfachste und meistgenutzte Applikation ist dabei die Suche. Zur Realisierung des Web-Portals wurde ein Content-Management-System (CMS) eingesetzt, das als besonderes Merkmal passend zur Dynamik der Personalisierung einen ausgefeilten Cache-Mechanismus bereitstellt, der trotz der Personalisierung ganze und Teile von Webseiten effizient vorbe-



reitet speichern kann, sodass nicht jedes Mal die gesamte Seite berechnet werden muss.

Das CMS verwendet zur Generierung der Seiten die Skriptsprache Tcl mit einem Anschluss an Java. Das mit der Entwicklung beauftragte Unternehmen entschied sich, die gesamte Logik in Java zu implementieren und diese aus den Schablonen des CMS aufzurufen. In diesem Zusammenhang stellte sich die Frage, wie die Sitzungen verwaltet werden. Sowohl das CMS als auch der Java-Applikationsserver stellten Sitzungen zur Verfügung. Da wenig Know-how in Bezug auf die CMS-Sitzungen verfügbar war, wurden die des Applikationsservers verwendet.

Später stellte sich heraus, dass durch diese Entscheidung der gesamte Caching-Mechanismus des Systems ausgeschaltet wurde, da das Caching von der Implementierung der Sitzung abhängig war. Dies hatte zur Folge, dass jede Seite nahezu vollständig generiert werden musste, um die Personalisierungs-Informationen in den Quelltext der Seite einzubauen.

### *Reflexion*

Die Aktanten in diesem Vorfall sind neben den Benutzern die Mitarbeiter der Entwicklerorganisation, das Content-Management-System und der Applikationsserver. Durch das Abrufen vieler Seiten zur Mittagszeit entsteht eine hohe Auslastung, die sich zu einem kritischen Problem entwickelt. Zurückgefallene Teile sind in diesem Zusammenhang Entwurfsentscheidungen bei der Programmierung von Schablonen und Applikationen. In beiden o. g. Produkten sind Entscheidungen eingeschrieben, deren Änderbarkeit nicht leicht zu erkennen ist. Beide beanspruchen für sich die Verwaltung der Sitzung. Die Kenntnis der Entwickler definiert nun den Widerstand der Produkte und die Flexibilität, in der agiert werden kann. Da nicht ausreichend Wissen über das CMS und dessen Sitzungsverwaltung existierte, wurde die Entscheidung zugunsten des anderen Produktes gefällt.

In diesem Vorfall können die aufeinander treffenden Ansprüche der beiden Produkte als *Wechselwirkung* gesehen werden, deren Auswirkung erst später erkennbar wurde. Gleichzeitig wurde mit der Entscheidung die *Entwicklungsfähigkeit* der Infrastruktur behindert, da unnötig viel Last auf die die Webseiten ausliefernden Maschinen gegeben wurde. Das dadurch entstehende Problem kann als *kritisches Problem* im Sinne der LTS-Diskussion verstanden werden.

#### 4.3.4 Preismodell konterkariert Implementierung: Mandantenfähigkeit

„Gestern waren die Vertriebsbeauftragten da und nachdem wir ihnen unser Vorhaben erklärten, hatten sie Dollarzeichen in den Augen. Die hatten schon einen Stapel Auftragsblöcke mitgebracht!“

*(Beleg fn Februar 2001)*

##### *Vorfall*

Die Firmenorganisation zum Betrieb von *hamburg.de* geht von einer *hamburg.de* GmbH aus, die von der Stadt, lokalen Kreditinstituten und einer Betreibergesellschaft gegründet wurde. Diese GmbH hat an die Betreibergesellschaft den Auftrag gegeben, eine Plattform zu entwickeln und zu betreiben. Dasselbe Modell hat die Betreibergesellschaft mit anderen Ländern und Kommunen praktiziert. Die Betreibergesellschaft hat zur Entwicklung der Software ein Entwicklungsunternehmen beauftragt, die notwendige Software von Softwareherstellern zugekauft und mit einer Betriebsgesellschaft einen Vertrag geschlossen, Hardware ans Internet anzuschließen und zu betreiben.

Mit diesem komplexen Modell möchte der Dienstleister aus den gleichartigen Betreiberverträgen Synergieeffekte gewinnen und dieselbe Software auf derselben Hardware für verschiedene Kunden betreiben. Dafür muss die Software mandantenfähig sein. Mandantenfähig in diesem Zusammenhang bedeutet, dass dieselbe Software, die mehrere Mandanten versorgt, sich abgeschlossen und individuell an die Bedürfnisse eines Mandanten angepasst präsentiert.

Die Entwicklung der Internet-Plattform wurde zuerst für zwei Mandanten vorgenommen. Es stellte sich heraus, dass ein erheblicher Aufwand in die Unterscheidung der Mandanten gesteckt werden musste. Entgegen der anfänglichen Auffassung war dies nicht nur „ein Feld in der Datenbank“. Darüber hinaus stellten die Mandanten unterschiedliche Forderungen in Bezug auf die Realisierung einzelner Bestandteile, sodass Sonderlösungen entwickelt werden mussten.

Gegen das Vorhaben Mandantenfähigkeit sprechen die Lizenzmodelle der Software-Hersteller, die neben Kosten pro Prozessor und Maschine ebenfalls pro bedienter Domain Lizenzgebühren verlangen. Diese Bedingung wurde nicht in das Kalkül mit einbezogen. Die Forderungen schmäl-

lerten die Synergieeffekte und machten einige Vertragsbeziehungen unattraktiv.

### *Reflexion*

Relevante Aktanten des Netzes sind hier die Software-Produkte, deren Vertragsbedingungen, die Entwickler und Mitarbeiter der Kostenplanung. In diesem Vorfall ist wiederholt die Stabilität der errichteten Software-Infrastruktur ein wesentlicher Grund für die auftretenden kritischen Probleme. Der Wechsel von einmal ausgewählten und bereits betriebenen Produkten ist nur schwer zu vollziehen. Dadurch wiegt die Forderung aufgrund der Lizenzbedingungen schwer. Das Interesse des Akteurs CMS-Hersteller ist es, sein unveränderliches Bewegliches in das Netz des Betreibers einzubringen, um dadurch seine Interessen wahrzunehmen und als relevanter Akteur aufzutreten. Die Aufgabentrennung innerhalb der Betreiberorganisation zwischen Entwicklung und Finanzen hat sich in diesem Zusammenhang als besonders nachteilig erwiesen. Weder auf den Ebenen *Entwicklungsfähigkeit* noch auf der Ebene der *Konfigurierbarkeit* kann der Software ein nachteiliges Verhalten attestiert werden.

## **4.4 Die Lernplattform *CommSy* als Infrastrukturbestandteil**

The more I live – the more I learn.  
The more I learn – the more I realize  
The less I know.  
Each step I take –  
Each page I turn –  
Each mile I travel only means  
The more I have to go.  
“A Peace of Sky” — Alan and Marilyn Bergman

Das Community System *CommSy* (Bleek et al. 2000, Pape et al. 2002a, Janneck und Bleek 2002) ist aus einem universitären Projekt zum Wissensmanagement entstanden, in dem unterstützende Dienste zur Projektarbeit und für Arbeitsgruppen entwickelt wurden. Sein technischer Kern ist eine dynamische Webserver-Applikation, die die Kommunikation zwischen den

Mitgliedern eines Projektes erlaubt. Das System wurde in einer großen Anzahl von Projekten benutzt und wissenschaftlich ausgewertet. Die Entwicklung des Prototypen begann mit zwei bis fünf Personen für eine extracurriculare Arbeitsgruppe. Die frühen Ergebnisse ermutigten die Beteiligten, weitere Entwicklungen des Systems durchzuführen, sodass es für universitäre Lehre-Projekte genutzt werden konnte. Die Qualität der Projektarbeit konnte durch die Anwendung des Systems verbessert werden: sowohl bezüglich der Kommunikationsmöglichkeiten als auch der Bindung zwischen allen Beteiligten. Seitdem haben viele Lehrveranstalter das *CommSy* in Veranstaltungen unterschiedlicher Universitätskontexte verwendet (u. a. bei der *Internationalen Frauenuniversität*). Das System wurde aufgrund der Projektanforderungen stetig weiterentwickelt. Ein nationales Forschungsprojekt konnte akquiriert werden, welches vom Bundesministerium für Bildung und Forschung (BMBF) gefördert wird. Eine ausführliche Darstellung des Verlaufs dieser Aktivitäten findet sich im Anhang C. „Projekt“ oder „Arbeitsgruppe“ bezeichnet dabei die physischen Aktivitäten. Der Begriff „Projektraum“ bezieht sich auf einen durch das technische Medium bereitgestellten virtuellen Raum für eine geschlossene Personengruppe.

Das *CommSy* kann als ein Infrastruktur-Bestandteil eines Lehr-Lern-Kontextes gesehen werden. Lehrende und Studierende passen sich diesen Bestandteil in ihre Arbeit ein. Aus der Infrastruktur-Perspektive stellt es darüber hinaus einen Bestandteil dar, der sich in eine bestehende Infrastruktur einfügt und diese um neue Möglichkeiten ergänzt.

Die hier vorgestellten fünf Vorfälle sind chronologisch geordnet. Ihre Auswahl habe ich nach meiner Meinung nach wesentlichsten Vorkommnissen im Rahmen des Betriebs und der Entwicklung des Infrastruktur-Bestandteils *CommSy* getroffen. Die Vorfälle decken dabei die Spannweite des hier zu Grunde gelegten Infrastruktur-Begriffs ab. Im ersten Vorfall wird thematisiert, wie bereits während der Entwicklung die vom Fachbereich Informatik gelebten Organisationsgrenzen die Infrastrukturentwicklung behinderten. Der zweite Vorfall zeigt, wie die Entwicklung eines Bestandteils des *CommSys* Auswirkungen auf den gesamten Einsatz der Software hatte. Der dritte Vorfall beschreibt die gegenläufigen Entwicklungen für zwei parallel laufende Einsatzkontexte. Im vierten Vorfall geht es um die Übertragung des Einsatzes auf einen kommerziellen Kontext mit unbekannten Nutzergruppen. Schließlich nimmt der fünfte Vorfall die Probleme in der Kommunikation mit Benutzern in jenem System auf.

#### 4.4.1 Reibungsverluste: Überschrittene Organisationsgrenzen

Die Entwicklung des Community Systems fand unter der Beteiligung zweier Wissenschaftlicher Mitarbeiter aus unterschiedlichen Arbeitsbereichen (ASI und SWT) statt. Die dabei gelebte Organisationsgrenze, die in der Infrastruktur des Fachbereichs Informatik reflektiert wurde, stellte vielfach ein Hindernis bei der Entwicklung und beim Betrieb des *CommSys* dar.

(Beleg fn 9. November 1999)

##### *Vorfall*

Die Entwicklung des *CommSys* entstand aus der extracurricularen Arbeitsgruppe *KnowNet*, in der Studierende und Mitarbeiter aus zwei Arbeitsbereichen mitwirkten. Der erste Prototyp wurde in einer kleinen Gruppe von zwei Mitarbeitern und wenigen Studierenden entwickelt. Als technische Basis für die Entwicklung wurde das am Fachbereich vorhandene Betriebssystem Unix verwendet. Es kam eine Zusammenstellung von Server-Software zum Einsatz, von der Teile bereits vorkonfiguriert waren und auf den Rechnern des Arbeitsbereichs SWT liefen. Die an der Entwicklung beteiligten und alle anderen Teilnehmer der Arbeitsgruppe besaßen durch entsprechende Unix-Gruppen allerdings nur Zugriffsrechte auf die Bereiche des Arbeitsbereichs ASI.

Die Nutzung der Unix-Software war an den Organisationsgrenzen der Arbeitsbereiche ausgerichtet. Jede Person am Fachbereich bekommt ein Grundkonto, das mit minimalen Möglichkeiten ausgestattet ist. Die Zugehörigkeit als Mitarbeiter zu einem Arbeitsbereich ermöglicht den Zugriff auf dessen Infrastruktur. Die Eintragung in eine Lehrveranstaltung ermöglicht den Zugriff auf die für diese Lehrveranstaltung vorgesehenen Ressourcen. Somit muss eine Veranstaltung einem einzigen Arbeitsbereich zugeordnet werden. In diesem Modell gibt es weder Mitarbeiter, die in mehreren Arbeitsbereichen gleichwertig tätig sind, noch Lehrveranstaltungen, die gleichermaßen von mehreren Arbeitsbereichen betrieben werden.

Das Problem bestand nun wiederholt darin, dass Studierende und Mitarbeiter nicht sofort an der Entwicklung teilnehmen konnten, da sie keinen vollständigen Zugriff auf die als Infrastruktur gewählten Bestandteile hatten. Hier mussten Studierende in Unix-Gruppen des Arbeitsbereichs eingetragen werden, der die Software-Infrastruktur für das *CommSy* betrieb, obwohl sie in einem anderen Arbeitsbereich z. B. ihre Studienarbeit schrieben. Darüber hinaus war es während des Betriebs eines *CommSy*-Projektraums

notwendig, in der Anfangsphase Teilnehmende einzutragen und Konfigurationsarbeiten durchzuführen. Dies konnte nur von Mitarbeitern des Arbeitsbereichs, in dem die Software lief, durchgeführt werden. Wiederholt musste der Wissenschaftliche Mitarbeiter, der die Lehrveranstaltung betreute, sich an Kollegen aus dem anderen Arbeitsbereich wenden, um diese Eintragungen und Änderungen durchzuführen. Einige Aktionen konnten erst verspätet durchgeführt werden, weil kein Ansprechpartner vorhanden war.

Als Lösung wurde versucht, den Mitarbeitern des anderen Arbeitsbereichs möglichst gleichwertige Zugriffsrechte für die Infrastruktur zu geben. Das war nicht bis zur letzten Konsequenz möglich, da ihnen dadurch ebenfalls der volle Zugriff auf interne Bereiche des Arbeitsbereichs gegeben worden wäre. Es blieb also die Notwendigkeit, während der Entwicklung und des Betriebs des *CommSys* auf die Hilfestellung der Mitarbeiter zurückzugreifen.

### *Reflexion*

Das Beispiel zeigt, dass die Entwicklung eines Infrastruktur-Bestandteils von der Basis-Infrastruktur abhängig ist (“built on an installed base”). Dabei wird die Form der neuen Infrastruktur und ihre Entstehung durch die vorhandene Infrastruktur geprägt. Aus Sicht der LTS-Diskussion kann von einem fundamentalen Erfindungsprozess gesprochen werden, dessen Systembauer sich trotz der durch die Basis vorgegebenen Rahmenbedingungen von ihrem Vorhaben nicht abbringen ließen. Aus der ANT-Perspektive zeigt sich die oben beschriebene Konstellation als ein Netz von Aktanten, in dem die Abteilungsgrenzen durch die existierende technische Infrastruktur wiederholt werden. Die Trennung in Abteilungen wiederholt sich in der Technik sowohl in den separaten als auch in den gemeinsam genutzten Geräten. Der Grad der Unumkehrbarkeit ist dabei so hoch, dass die Akteure ihr Handeln den äußeren Begrenzungen anpassen mussten. (Dieser Umstand ist ein Grund, das Forschungsprojekt *WissPro* zu schaffen und in selbstbestimmten Rahmenbedingungen weiterarbeiten zu können.) Der erste Vorfall verdeutlicht, dass Infrastruktur-Entwicklung immer von den *Gegebenheiten* abhängig ist, in denen die Entwicklung stattfindet. Diese Gegebenheiten entstehen durch die existierenden Praxisgemeinschaften. Entscheidend ist, in welchem Maße diese Gegebenheiten passend zu den aktuellen Bedürfnissen *konfigurierbar* sind.

#### 4.4.2 Langsame Diskussionsforen: Ineffiziente Datensammlung

Ein Kernbestandteil des *CommSys*, die Diskussionsforen, gab nach einer Re-implementierung Anlass für Verärgerung, da die Generierung der Webseiten für das Diskussionsforum besonders langsam war. Dies lag an ineffizienten Datenbankzugriffen und einer für Browser nicht angemessenen Struktur der darstellenden HTML-Seiten.

(Beleg E-Mail 7. März 2000)

##### *Vorfall*

Die *CommSy*-Entwicklung fand (und findet) in einem Netz verschiedener Akteure statt. Zu diesen Akteuren zählen neben den Professoren und Wissenschaftlichen Mitarbeitern Studierende, die sich freiwillig engagieren und Studien- und Diplomarbeiten schreiben. Insofern hängt die Weiterentwicklung nicht zuletzt davon ab, welche Themen für Studien- und Diplomarbeiten gewählt werden.

Ein Beispiel für eine solche Arbeit ist die Re-Implementierung der Diskussionsforen, die im Rahmen einer Studienarbeit durchgeführt wurde. Die Arbeit wurde von einem Wissenschaftlichen Mitarbeiter betreut, der zu dem Zeitpunkt nicht im Kern des Entwicklerteams stand. Der Studierende lieferte nach einigen diskutierten Prototypen eine neue Implementierung ab und spielte sie in die Versionsverwaltung und einen laufenden Projektraum ein. Dabei wurden andere laufende Projekträume und das Verhalten der neuen Implementierung unter realen Anforderungen nicht berücksichtigt. Verschärft wurde das Problem dadurch, dass zu dieser Zeit alle laufenden Projekträume separat installiert waren.

Aus dieser Handlung folgten mehrere Konsequenzen. Erstens waren die Anwender des laufenden Projektraums mit der Visualisierung und Handhabung, aber nicht mit der Geschwindigkeit der neuen Lösung zufrieden. Deren Ansprechpartner konnten aber an dem Umstand nichts ändern, da der verantwortliche Student nicht erreichbar war und nicht angemessen schnell reagieren konnte. Andere laufende *CommSy*-Projekträume wollten die neue Visualisierung und Handhabung ebenfalls nutzen. Hierfür war allerdings die notwendige Migration der bestehenden Daten nur unzureichend dokumentiert.

### *Reflexion*

Dieses Beispiel dokumentiert, wie die Entwicklung aus einer Studienleistung Einfluss auf unterschiedliche Einsatzkontexte des *CommSy* ausübt. Es offenbart das Akteur-Netz, in dem das *CommSy* als Infrastruktur-Bestandteil für die Lehre existiert: Die neuen Diskussionsforen erhöhen die Auslastung und lassen dadurch ein kritisches Problem entstehen. Bereits laufende *CommSy*-Projekträume üben deshalb Druck auf die ansprechbaren Vertreter aus. Sie fordern eine Änderung des aktuellen Zustands (zu langsame Foren, keine moderne Visualisierung), während der Auslöser für die Veränderung sich als relativ unabhängig vom Verpflichtungsgefüge sieht.

Für die Projektveranstalter sind die Wissenschaftlichen Mitarbeiter Delegierte, die für das System sprechen. In diesem Sinne werden Forderungen an die Delegierten gerichtet, wie auch Änderungen den Delegierten zugesprochen werden. In die implementierten Änderungen sind Eigenschaften eingetragen, deren Beharrlichkeit bzw. Maß an Unumkehrbarkeit von der Nähe zur Entwicklung der Änderung abhängt: Je weiter man von der Änderung entfernt ist, umso weniger besteht Einfluss zur Änderung. Insofern kann eine solche Änderung auf Software-Ebene als unveränderliches Bewegliches gesehen werden, dessen Wirkungskreis weit über den lokalen Bereich hinausgeht. In diesem Beispiel ist die *Performanz* für die Anwender besonders wichtig.

### **4.4.3 Gegenläufige Entwicklungen: Zwei Projekte**

Im Frühjahr 2000 entschieden sich zwei Entwickler, dass das *CommSy* einerseits für den Studiengang Wirtschaftsinformatik und andererseits für die *Internationale Frauenuniversität* eingesetzt werden sollte. Beide Projekte erforderten von ihnen substantielle Anpassungen und Weiterentwicklungen an dem *CommSy*. Die dafür von ihnen initiierten Entwicklungsprojekte waren nicht aufeinander abgestimmt, sodass die Entwicklungen im einen Projekt die des anderen Projektes behinderten oder konterkarierten.

*(Beleg fn 21. Mai 2000)*

### *Vorfall*

Neben der Gruppe von *CommSy*-Interessierten und Entwicklern, die ein hoch frequentiertes *CommSy* für sich betrieben, versprach man sich von den Studierenden der Wirtschaftsinformatik und dem Projekt *Internationale Frauenuniversität* jeweils einen interessanten Nutzungskontext. Beide



Projekte sollten beginnend vom Sommer 2000 durch *CommSy* unterstützt werden, wofür jeweils ein Wissenschaftlicher Mitarbeiter eintrat.

Die Anforderungen der beiden Entwicklungsprojekte an das *CommSy* waren zu Anfang unscharf und wurden nur wenig im Kreise der Entwickler diskutiert. Für das eine Projekt war die Benutzerverwaltung in einer Datenbank geplant, bei dem anderen der Anschluss der Benutzerverwaltung an einen Directory Server. Für das eine Projekt stand das Beantragen und Verwalten von Benutzerkennungen im Vordergrund, bei dem anderen die Speicherung mehrerer Projekträume in derselben Datenbank und die Unterstützung von Mehrsprachigkeit. Nachdem beide Entwicklungsvorhaben sich mehrfach behinderten und wenig Raum für eine kontinuierliche Kommunikation gesehen wurde, entschloss man sich, zwei getrennte Entwicklungspfade zu verfolgen (Juni 2000).

Eine Folge der Trennung der Entwicklungspfade war nun, dass andere Installationen des *CommSys* sich gezwungenermaßen entscheiden mussten, welchen der Pfade sie weiterverfolgten. Dies führte zu dem Dilemma, dass in vielen Fällen beide Entwicklungspfade wünschenswerte Verbesserungen und Fehlerkorrekturen enthielten, wovon letztendlich nur eine Zusammenstellung direkt nutzbar war. Nach fünf Monaten getrennter Entwicklung und getrenntem Einsatz in den genannten Kontexten wurden die beiden *CommSy*-Versionen im Herbst wieder zusammengeführt (November 2000). Damit bestand wieder eine gemeinsame Basis, die für die Entwicklung und den Einsatz in allen Kontexten verwendet werden konnte.

### *Reflexion*

Dieses Beispiel zeigt, wie die äußeren Umstände auf einen inneren Entwicklungsprozess einwirken. Das aufgebaute Akteur-Netz der *CommSy*-Entwicklung und des *CommSy*-Einsatzes wird erweitert. In diesem Vorfall wirken die zwei Einsatzkontexte als Aktanten auf das (gekapselte) *CommSy*-Akteur-Netz mit so starker Kraft, dass sie die vorher als Einheit auftretende Entwicklung in zwei unabhängige Aktanten trennen. Kennzeichnend für die gesamte Entwicklung des *CommSys* ist, dass wiederholt neue Akteure aufgetreten sind, mit denen die Entwicklergruppe schrittweise umzugehen lernte, indem sie in das Netz integriert wurden.

Bei den einzelnen Veränderungen am *CommSy* im Rahmen der o. g. Entwicklungsprojekte wurde die *Entwicklungsfähigkeit* nicht immer berücksichtigt, sodass teilweise notwendige Entwicklungspfade als Optionen ver-

sperrt wurden. Insgesamt können die beschriebenen Entwicklungsprozesse als Beispiel für *Wechselwirkungen* bei der Infrastruktur-Entwicklung dienen, die auftreten, wenn Innovationen mit unterschiedlicher Zielsetzung an derselben Infrastruktur durchgeführt werden. Diese Zielsetzungen werden über unterschiedliche Praxisgemeinschaften motiviert, die mit dem *CommSy* arbeiten. Für eine Konsolidierung war in diesem Rahmen kein Platz vorgesehen.

#### 4.4.4 Neuer Kooperationspartner: Aufbau eines öffentlichen Servers

Ein Internet-Unternehmen beauftragte die *CommSy*-Entwickler, das System so weit zu entwickeln, dass es auf dessen öffentlichem Server als Ergänzung zu anderen Dienstleistungen betrieben werden konnte. Zum Aufbau eines autarken *CommSy*-Servers auf der Hardware des Internet-Unternehmens war es notwendig, substantielle Veränderungen im administrativen Bereich durchzuführen.

(Beleg Arbeitspakete 30. August 2000)

##### *Vorfall*

Die Entwicklung des *CommSys* ist ausgehend von einem Prototypen über einzelne Installationen und eine wachsende Anzahl von Installationen hin zu Installationen mit mehreren *CommSy*-Projekträumen historisch gewachsen. Kennzeichnend für diese Entwicklung ist, dass schrittweise versucht wurde, den steigenden Installations- und Betriebsaufwand zu reduzieren (Bleek und Pape 2001, Bleek et al. 2003). In der ersten Installation wurde die Arbeit auf die Mitglieder der initialen Arbeitsgruppe verteilt. Bei weiteren Installationen wurde jeweils versucht, entsprechende Verantwortliche für den Betrieb und die Betreuung zu finden. Die Wissenschaftlichen Mitarbeiter lernten dabei, dass sie in vielen Kontexten einen Großteil der anfallenden Arbeiten leisteten. Nur wenige Personen waren in der Lage, einen *CommSy*-Raum einzurichten und zu konfigurieren.

Der autarke Server des beauftragenden Unternehmens sollte es nun ermöglichen, dass Lehrende die notwendigen Informationen für einen Projektraum selbst hinterlegten und die Kommunikation mit Studierenden gleichermaßen über sie gepflegt würde. Die Aufgabe des betreibenden Unternehmens ist darin zu sehen, zu entscheiden, wer einen solchen Raum bekommen soll. Beides sollte als Bestandteil des *CommSys* zur Verfügung

gestellt werden. Für die Entwickler blieb die Rolle der technischen (Weiter-)Entwicklung des Systems.

### *Reflexion*

Zentral für diesen Vorfall ist, dass die in den ersten Projekten unscharfen Rollenaufteilungen expliziert werden: Lehrende und Teilnehmende einer Lehrveranstaltung, das Unternehmen, das den Server betreibt, das Team, das die Software entwickelt, das Unternehmen, das die Hardware betreibt usw. Diese treten alle als Akteure in einem Netz auf und haben Erwartungen über die Leistungen der anderen. Zusätzlich sehen nicht alle Beteiligten die detaillierte Aufgliederung. Die Lehrenden und Teilnehmenden einer Lehrveranstaltung wissen z. B. nichts über die Trennung zwischen Host, Entwickler und Betreiber. Das unter dem Namen *CommSy* gekapselte Akteur-Netz muss für die beschriebene Übertragung geöffnet und die Rollen und Beziehungen einzelner Akteure müssen expliziert werden. Dies löst einen Prozess der Konsolidierung aus.

In diesem Projektkontext stellt sich die Frage nach der *Erlernbarkeit* eines Infrastruktur-Bestandteils, d. h. in diesem Fall, wie das *CommSy* als ein Bestandteil einer Lehr-Lern-Infrastruktur in die alltägliche Nutzung eingepasst werden soll. Wie kann das *CommSy* genutzt werden, welche Funktionalitäten gibt es, welche etablierten Nutzungsschemata sind vorhanden? Entscheidend hierfür ist, ob dieser Infrastruktur-Bestandteil ein hohes Maß an *Selbstbeschreibungsfähigkeit* besitzt, sodass Benutzer sich anhand der erkennbaren Funktionen deren Umfang und Zusammenspiel erschließen können.

Für Infrastrukturentwickler und -betreiber steht die *Reproduzierbarkeit* von Infrastruktur hier im Mittelpunkt des Interesses. Das bezeichnet einerseits die Frage nach der Installation auf einem fremden System in einer anderen technischen Umgebung. Andererseits bezieht sich die Reproduzierbarkeit auch auf die Anwendungskontexte, in denen *CommSy* als Infrastruktur-Bestandteil einer Lehr-Lern-Umgebung genutzt wird.

Für Entwickler, Host und Betreiber hat *Wartbarkeit* im Rahmen einer Konsolidierung die höchste Priorität. Entwickler fordern hier eine gut unterstützte Fehlersuche und leichtes Einspielen von Updates. Host erwarten geringe Komplikationen im Zusammenspiel mit anderen Infrastruktur-Bestandteilen ihrer technischen Landschaft und eine klare Trennbarkeit von anderen Akteuren und Zusammenarbeit mit anderen Akteuren. Aus der Be-

treiberperspektive wird die Wartbarkeit primär an dem Aufwand gemessen, der für die Wartung notwendig ist (Pape et al. 2002b).

#### 4.4.5 Update einspielen: Benutzer informieren

Nach Installation und Bereitstellung des *CommSys* auf einer Vielzahl von Servern stellte sich bei Updates das Problem, wie man die dortigen Benutzer sinnvoll über die Update-Prozedur informiert.

(Beleg E-Mail 7. August 2001)

##### *Vorfall*

Ein zentrales Thema, das in allen hier beschriebenen Infrastrukturprojekten wahrgenommen wurde, ist die Kommunikation der Infrastruktur-Betreiber und -Entwickler mit den Benutzern. Im Kontext des *CommSy*-Projektes bezieht sich dies auf zwei Standardsituationen, nämlich das Ankündigen von Update-Einspielungen und unvermeidliche Ausfallzeiten. Beide Situationen rühren aus normalen Tätigkeiten bei der Wartung des Servers her. Ausfallzeiten entstehen durch notwendige Neustarts z. B. wegen eingespielter Betriebssystempatches oder durch Updates des *CommSys*, wenn Fehler zu relevanten Problemen behoben werden konnten.

Erstmals relevant wurde die Problematik um die Kommunikation mit den Benutzern beim Einsatz des *CommSys* bei der *Internationalen Frauenuniversität* und danach beim autarken Server (s. o.). In beiden Fällen bestand kein persönlicher Kontakt zu den Studierenden, die das System nutzten, und nur ein spärlicher Kontakt zu den Projektraumleitern. Updates oder Unterbrechungen wurden jeweils einige Tage im Voraus geplant, so dass der Zeitpunkt im Allgemeinen ca. 5-7 Tage vorher feststand. Die Problematik bestand darin, die relevanten Personen zu informieren. (Es steht außer Frage, dass die Information nur auf elektronischem Wege erfolgen kann.) Hierbei treten verschiedene Probleme auf. Einerseits stellt sich die Frage, wen man informiert und mit welchem Absender. Andererseits ist unklar, auf welchem Wege man die Beteiligten informiert.

Verworfen wurde die Idee, alle Benutzer per E-Mail anzuschreiben. Dagegen sprechen mehrere Gründe: Erstens würden auch Personen informiert, die das System nicht benutzen; diese könnten sich gestört fühlen. Zweitens würden Personen informiert, die das System im relevanten Zeitraum nicht nutzen; diese würden unnötig den Eindruck von „häufigen“ Ausfällen bekommen. Drittens ist unklar, mit welchem Absender die E-Mail zu

schreiben wäre, da die meisten nur Kontakt zu ihrem Lehrenden haben. Die Entscheidung wurde dahingehend getroffen, die Projektraum-Leiter zu informieren. Allerdings erhielten keine von diesen Rückmeldung. Nicht erprobt wurden verschiedene Möglichkeiten, die Wartung anzukündigen.

### *Reflexion*

Aktivitäten im Rahmen von Wachstum und Konsolidierung erfordern immer wieder eine Aktualisierung des Systems. Aus der Sicht von ANT wird das um den *CommSy*-Einsatz gebildete Netz komplex, wenn die Benutzer als einzelne Akteure aufgenommen werden. Selbst wenn man je Projektraum nur einen Akteur für die Benutzer sieht, ist deren Zahl groß. Diese zu informieren muss demnach systematisch betrieben werden. In der Forderung, möglichst wenig Ausfallzeiten zu haben, um keine kritischen Probleme zu erzeugen, und gleichzeitig ein fehlerfrei funktionierendes System nutzen zu können, vereinigen sich diese Akteure zu einem, der einen großen Druck ausübt. Aus dessen Sicht stellt sich die Lernplattform als ein Bestandteil in ihrer Lern-Infrastruktur dar, dessen komplexe Realisierung nicht von Interesse ist. Die Eigenschaft *Kommunizierbarkeit* einer Infrastruktur wird in diesem Zusammenhang auf die Frage ausgedehnt, wie sich Arbeiten an der Infrastruktur, die sich in Ausfällen äußern, an die Benutzer kommunizieren lassen. In diesem Zusammenhang meint die Eigenschaft *Wartbarkeit* auch die Möglichkeit, systematisch für Benutzer ein konsistentes und informatives Bild der Infrastruktur zu liefern, auch wenn Teile davon nicht verfügbar sind.

## **4.5 Eigenschaften und Phänomene zusammengefasst**

Die Projektbeschreibungen und insbesondere die Vorfälle haben verdeutlicht, dass Informations-Infrastrukturen in ganz unterschiedlichen Milieus auftreten, und es wurde eine Fülle von Eigenschaften und Phänomenen offenkundig. Obwohl alle Projekte ein großes Spektrum im Infrastruktur-Bereich abdecken, steht jedes Projekt für einen Schwerpunkt. Die Vorfälle des *ifu*-Projektes betonen räumliche und kulturelle Distanz. Bei *hamburg.de* werden andere Nutzungszusammenhänge und verschiedene Praktiken in den Vordergrund gestellt. Schließlich liegt der Schwerpunkt beim *CommSy*-Projekt in der Überschreitung von Organisationsgrenzen.

Am Ende jedes Vorfalles (und in den Anhängen) habe ich in einer Re-

flexion konkrete Eigenschaften und Phänomene benannt, die die Beobachtungen auf den Punkt bringen. Diese Eigenschaften sind wegen des empirischen Ansatzes nicht vollständig, denn lediglich vor dem Hintergrund der jeweiligen Projekte konnten Erfahrungen gesammelt werden. Aufgrund der gewählten Forschungsmethodik kann ich diesen Anspruch nicht formulieren. Durch den Zugang über Projekte sind darüber hinaus die Eigenschaften nicht disjunkt. Vielmehr motivieren verschiedene Perspektiven ähnliche Charakteristika.

Aus meiner Sicht ist nur ein Teil der Eigenschaften interessant. Die in den Vorfällen und in den Projektbeschreibungen genannten Eigenschaften Skalierbarkeit, Wartbarkeit, Performanz, Redundanz und Ausfallsicherheit scheinen mir von keiner besonderen Bedeutung, da sie im Kontext von Infrastrukturen über das bereits in der Informatik etablierte Verständnis nicht hinausgehen. Mit diesen Eigenschaften werde ich mich im Folgenden nicht beschäftigen. Die übrigen Eigenschaften möchte ich zur einfacheren Handhabung bündeln. Deshalb fasse ich *Übertragbarkeit* mit *Reproduzierbarkeit* zusammen. *Kommunizierbarkeit*, *Erlernbarkeit* und *Selbstbeschreibungsfähigkeit* werden unter *Alltagstauglichkeit* subsummiert. Die Eigenschaft *Kulturalität* fällt meines Erachtens unter *Standardisiertheit*. Die *unerwarteten Anforderungen* werden von der *Entwicklungsfähigkeit* aufgenommen. *Konfigurierbarkeit* und *Aufbaubarkeit* bleiben für sich alleine stehen.

*Reproduzierbarkeit* bezeichnet die Möglichkeit, den Aufbau einer Infrastruktur mit gleichem oder ähnlichem Anwendungsfeld an einem anderen oder an demselben Ort zu wiederholen (z. B. Vorfall 4.3.1, 4.4.4). Diese Eigenschaft findet sich ebenfalls im Bereich der Übertragung großer technischer Systeme, wenn diese an anderen Standorten von z. B. demselben Unternehmen ebenfalls aufgebaut werden sollen (vergleiche Kapitel 2.1.2). Reproduzierbarkeit kann als *Übertragbarkeit* ausdifferenziert werden, wenn im Zielkontext bereits eine Infrastruktur existiert, der die Eigenschaften der ersten hinzugefügt werden sollen.

*Alltagstauglichkeit* fordert von einer Infrastruktur, dass deren Gebrauch für einen Benutzer oder Entwickler im alltäglichen Ausüben von Tätigkeiten sinnvoll in den Alltag eingepasst werden kann. Sie setzt auf den Kriterien *Selbstbeschreibungsfähigkeit*, *Transparenz* (Maaß 1994b, Maaß 1994a) und *Lernförderlichkeit* aus den ergonomischen Kriterien für Bildschirmarbeitsplätze auf (ISO 9241–10 1997). Die Eigenschaft geht aber über diese hinaus, indem sie sie zusammenfasst und den Alltag, nicht einen

Arbeitsplatz, als Maßstab anlegt. Damit verbunden ist das Bedürfnis des Benutzers, in seiner alltäglichen Arbeit in den durch die Infrastruktur angebotenen Dienstleistungen ein für sich nutzbares Angebot wiederzufinden. Diese Eigenschaft fußt auf *Erlernbarkeit* und *Kommunizierbarkeit*, die beim Austausch der Benutzer und Betreiber über die Infrastruktur essenziell werden. Sie sind Eigenschaften, die an der Grenze von *Innensicht* und *Außensicht* vorzufinden sind und auf beide Seiten gespiegelt werden (z. B. Vorfall 4.2.1, 4.4.5).

*Standardisiertheit* umfasst sowohl technische Standards als auch sozial herausgebildete im Rahmen von Kulturen (Bowker und Star 1999). Diese *Kulturalität* ist sowohl im Bereich der Benutzer (z. B. Vorfall 4.2.4) als auch der Systembetreiber (Orr 1996) erkennbar und kann durch das Begriffspaar *Außensicht* und *Innensicht* ausdifferenziert werden. Im Kontext von Protokollen und formalen Standards (z. B. Vorfall 4.3.1, 4.3.2) ist diese Eigenschaft ebenso für Benutzer wie für Entwickler von Bedeutung.

*Entwicklungsfähigkeit* bezeichnet die Flexibilität, mit der sich eine Software-Infrastruktur aus Sicht der Anwender und Entwickler im Rahmen von Anpassung auf neu erwachsende Anforderungen verhält. Auslöser dafür können im Sinne der LTS z. B. Rückschläge, grundlegende Veränderungen oder Innovationen gesehen werden. Entwicklungsfähigkeit kann in die Phasen der Entwicklung, Innovation, Wachstum der LTS-Diskussion eingeordnet werden. Aufbauend auf historischer Entwicklung kennzeichnet Entwicklungsfähigkeit die Aufnahmefähigkeit für Veränderungen und ist damit eine Basisforderung aus dem Bereich des Continuous Engineering. Nur durch ausreichende Entwicklungsfähigkeit kann man bei *unerwarteten Anforderungen* reagieren, die sowohl gegenläufige Entwicklungen von Infrastrukturen als auch von Bestandteilen innerhalb einer Infrastruktur bezeichnen (z. B. Vorfall 4.2.4, 4.3.1, 4.3.4, 4.4.4).

*Konfigurierbarkeit* zielt auf die Möglichkeit ab, die Software-Infrastruktur in den für den Anwendungskontext notwendigen Grenzen anzupassen. Dieser Vorgang lässt sich von der Entwicklung einer Infrastruktur nicht klar abgrenzen. Sein Schwerpunkt liegt in Änderungen, die ohne Programmierung und ohne strukturändernde Eingriffe durchgeführt werden können (z. B. Vorfall 4.3.4, 4.4.1). Der Vorgang des Konfigurierens kann sich dabei sowohl für den Benutzer als auch für den Betreiber einer Infrastruktur stellen. Ziel einer Infrastruktur-Entwicklung kann es sein, möglichst große Teile für den Benutzer ohne Hilfe des Betreibers konfigurierbar zu machen.

Im Gegensatz zu den übrigen Eigenschaften tritt *Aufbaubarkeit* hervor. Sie bezeichnet sowohl das Umfeld, in dem eine Infrastruktur aufgebaut werden soll (welche Hindernisse, welche Unterstützungen sind vorhanden?), als auch die Eignung einer Infrastruktur, in einem bestimmten Umfeld eingesetzt zu werden. Diese Eigenschaft steht nicht auf derselben Stufe wie die vorhergehenden. Alle anderen Eigenschaften sind davon betroffen. Der Kontext wird erkennbar, indem die Infrastruktur und deren Entwicklung auf *Gegebenheiten* einzugehen haben. Einen Rahmen bietet die Infrastruktur, da sie für die Aufgaben der Benutzer selbst als gegeben angenommen wird. Für eine softwaretechnische Fundierung ist diese breite Einbeziehung von Gegebenheiten (im Sinne einer "installed base") von grundlegender Relevanz. Dies wird unter dem Stichwort „Kontextuelle Gestaltung“ (Contextual Design) diskutiert (Wixon et al. 1990, Beyer und Holtzblatt 1997, Beyer und Holtzblatt 1999). Prozess und Produkt können nicht das Ziel verfolgen, eine gegebene Basis abzulösen. Kontextuelle Gestaltung propagiert Entwicklung im Kontext und unter Berücksichtigung des Kontextes. Die Projekte haben gezeigt, dass es notwendig ist, mit den Gegebenheiten umzugehen, sich auf sie einzustellen und deren Änderung über die Zeit einzuflechten (z. B. Vorfall 4.3.2, 4.4.1). Dazu müssen verschiedene Ebenen der Gegebenheiten unterschieden werden: organisatorische Praxis, existierende Ausstattung (z. B. an Räumen und Technik), beteiligte Personen, durchlebte Historie und andere.

Neben den sechs bis hierhin aufgeführten Kerneigenschaften ist ein Phänomen wiederholt in Erscheinung getreten, das sich nicht als Eigenschaft formulieren lässt. Charakteristikum von Infrastruktur-Entwicklung ist eine Vielzahl von unabhängigen Entwicklungsvorhaben an einer Infrastruktur. Daraus resultieren zwangsläufig Wechselwirkungen. Für das beobachtete Phänomen, dass sich die *Wechselwirkungen* zu *Störungen* in der Entwicklung verstärken, möchte ich den Begriff der *Interferenz* einführen. Die Interferenz wird im Folgenden eine Schlüsselrolle einnehmen.

Eine *Interferenz* ist ein übergreifendes Phänomen zwischen Entwicklungsvorhaben; es bezeichnet dabei zwei gleichzeitig auftretende Entwicklungen an einer Infrastruktur, die sich überlagern und damit im Allgemeinen unbeabsichtigt Einfluss aufeinander nehmen. In großen technischen Systemen wird davon ausgegangen, dass technische Rückschläge den Prozess behindern können und dass dies durch weiter ausgefeilte Ingenieursprinzipien adressiert werden kann. Interferenzen entstehen bei Infrastruk-



tur-Entwicklung dadurch, dass an derselben Infrastruktur unterschiedliche Organisationen oder Praxisgemeinschaften unabhängige Entwicklungsvorhaben durchführen, die sich zeitlich und/oder bezüglich des zu entwickelnden Infrastruktur-Bestandteils überlappen. Das Phänomen ist von besonderer Wichtigkeit, da es eine neue Qualität in der Software-Entwicklung darstellt (z. B. Vorfall 4.2.2, 4.2.3, 4.3.3, 4.4.2, 4.4.3).

Diese gebündelten Eigenschaften habe ich in der Tabelle 1 noch einmal aufgeführt. Die Eigenschaften sind aus meiner Sicht eine direkte Erweiterung zu den Infrastruktur-Eigenschaften von Star und Ruhleder aus der Perspektive der Software-Entwicklung (vergleiche Kapitel 3.3.2).

Tabelle 1: Die Kerneigenschaften von Software-Infrastrukturen

| <b>Kerneigenschaft</b> | <b>Damit verbundene Fragestellung</b>   |
|------------------------|---|
| Entwicklungsfähigkeit  | Kann die Software-Infrastruktur über einen längeren Zeitraum an unerwartet auftretende Anforderungen angepasst werden? Treten keine oder nur geringe Mengen von Merkmalen auf, die eine Anpassung an neue Anforderungen behindern bzw. die sich mit der aktuellen Entwicklung stören? Unerwartete Anforderungen unterstreichen somit das Problem, dass bei der Entwicklung vorher nicht antizipierte Anforderungen erwachsen. |
| Konfigurierbarkeit     | Ist die Software-Infrastruktur im Spielraum Einzelner in den für den Anwendungskontext notwendigen Grenzen anpassbar, ohne durch strukturändernde Eingriffe verändert werden zu müssen?   |
| Reproduzierbarkeit     | Kann eine gegebene Zusammenstellung als Software-Infrastruktur an einem anderen Ort oder in einem anderen Kontext unter ähnlichen Voraussetzungen erneut aufgebaut werden? Lässt sich eine Software-Infrastruktur in einem anderen Verwendungszusammenhang unter ähnlichen Voraussetzungen wieder verwenden?  |
| Alltagstauglichkeit    | Ist für Benutzer der Software-Infrastruktur deren Struktur und Nutzarmachung aus sich selbst erschießbar? Sind die Fähigkeiten, die Konfigurierbarkeit und die Grenzen der Software-Infrastruktur erlernbar und somit für eigene Zwecke übertragbar? Lassen sich die Fähigkeiten, Eigenschaften und Grenzen der Software-Infrastruktur vermitteln und als schlüssig darstellen?   |

(Fortsetzung auf der nächsten Seite)

(Fortsetzung)

| <b>Kerneigenschaft</b> | <b>Damit verbundene Fragestellung</b>   |
|------------------------|---|
| Standardisiertheit     | Werden in der verwendeten Software-Infrastruktur ausreichend etablierte Standards verwendet, die den Anschluss bzw. die Zusammenarbeit mit anderer (fremder) Software(-Infrastruktur) in der Zukunft ermöglichen? Welche kulturellen Annahmen sind durch die Software-Infrastruktur manifestiert und welche kulturellen Anforderungen der Anwender müssen erfüllt werden? |
| Aufbaubarkeit          | Erlauben es die (zu akzeptierenden) Gegebenheiten ("installed base"), die geforderte Software-Infrastruktur zu errichten und zu betreiben? Inwieweit gibt es Wechselwirkungen mit den vorgefundenen Gegebenheiten und anderen Software-Infrastrukturen, die die Aufgabenerledigung behindern?   |

## **Eigenschaften von Software-Infrastrukturen**

Real generosity toward the future lies in giving all to the present. — Albert Camus

Um auf die Frage hinzuleiten, wie Software-Entwicklung für Infrastrukturen entsprechend angepasst werden kann, möchte ich die Eigenschaften von Software-Infrastrukturen vor dem Hintergrund der relevanten Diskurse genauer charakterisieren. Die im vorangegangenen Kapitel erarbeiteten Kerneigenschaften werden systematisch gegen große technische Systeme, Software als Artefakt, Infrastruktur-Akteure und einen softwaretechnischen Prozess gehalten und diskutiert.

### **5.1 Ein Bewertungsrahmen für die Verdichtung und Einordnung der Empirie**

Als ersten Schritt zu einer konstruktiven Orientierung für die Software-Entwicklung werde ich die durch die eingehende Analyse gewonnenen Kerneigenschaften (vergleiche Kapitel 4) vor dem Hintergrund der relevanten Infrastruktur-Forschung und softwaretechnischen Landschaft diskutieren. Zur Infrastruktur-Forschung möchte ich beginnend bei großen technischen Systemen (vergleiche Kapitel 2) die Bemühungen um “information infrastructures” (vergleiche Kapitel 3.2), Kriterien für Infrastruktur und das Verstehen von Akteur-Netzen (vergleiche Kapitel 3.3) zählen. In die softwaretechnische Landschaft gehören das Verständnis von Software als Artefakt und Software als Prozess. Mich interessieren die betrachteten Infrastrukturen als Software-Infrastruktur im Sinne der Definition 3 auf Seite 44.

Anhand dieser Diskurse arbeite ich jeweils die maßgeblichen Annahmen und Eigenschaften heraus. Die Konsequenzen für Software-Infrastruktur stehen am Ende jedes Abschnittes. Meine Kerneigenschaften ordne ich in diese Konsequenzen ein. Dazu gehe ich in jedem Abschnitt die entschei-

denden Charakteristika des Diskurses durch, diskutiere und illustriere sie an Erfahrungen aus den drei empirischen Projekten exemplarisch.

Zur Herausarbeitung der unterschiedlichen für die Softwaretechnik relevanten Sichten, der damit verbundenen Entwicklungshaltung und dem jeweiligen Potenzial sollen im Folgenden die Kerneigenschaften und das Phänomen *Interferenz* im Kontext der empirischen Projekte vor dem Hintergrund der Sichten „Software-Infrastruktur als großes technisches System“, „Software-Infrastruktur als Artefakt“, „Software-Infrastruktur-Akteure“ und „Software-Infrastruktur-Entwicklung als Prozess“ betrachtet werden.

## **5.2 Software-Infrastruktur als großes technisches System**

Mit der Sicht „Software-Infrastruktur als großes technisches System“ wird der Bezug zu einem etablierten sozialwissenschaftlichen Diskurs hergestellt (vergleiche Kapitel 2). Dieser Diskurs stammt aus der Technikforschung und fokussiert primär auf nicht-informatische Betrachtungsgegenstände. Ich werde die Phasen im Folgenden jeweils auf eine Software-Infrastruktur als großes technisches System beziehen. Grenze des Ansatzes ist, dass er lediglich deskriptive Mittel zur Verfügung stellt, die nur retrospektiv eingesetzt werden können. Die folgenden Abschnitte sind entsprechend der herausgearbeiteten Phasen großer technischer Systeme gegliedert. Die drei Phasen können nicht nur nacheinander, sondern auch gleichzeitig, aber auf verschiedenen Ebenen stattfinden oder sich teilweise überschneiden.

### **5.2.1 Erfindung, Entwicklung und Innovation**

In den ersten Phasen (vergleiche Kapitel 2.2.2) werden als mögliche Intention für die Entwicklung einer Infrastruktur und deren Betrieb vorwiegend ökonomische Gründe genannt:

“Specifically, since ‘economic organization, including the firm, must reflect the fact that knowledge is costly to produce, maintain and use’ (Demsetz 1993), infrastructure is a means to lower such costs by allowing its efficient processing, transfer and accumulation.”

(Ciborra und Hanseth 1998, S. 266)

Das Ereignis der Erfindung kann in Bezug zu Infrastrukturen auf mehreren Ebenen gesehen werden. Eine Kernerfindung wie z. B. E-Mail kann im Mittelpunkt stehen und zum Ausbau einer Infrastruktur führen. Die Entwicklung bestünde dann in der Ausarbeitung der zu Grunde liegenden Standards und dem Anfertigen von Referenzimplementierungen. Es kann aber im Bereich der Informations-Infrastrukturen die Zusammenstellung der technischen Infrastrukturelemente den Charakter und damit die Infrastruktur begründen. Damit ist die Auswahl adäquater Implementierungen Teil der Entwicklung. Mit Entwicklung kann der Ausbau einer Erfindung zu einer den äußeren Umständen angepassten Infrastruktur bezeichnet werden. Die jeweils notwendigen Impulse sind als Anschub dieser Entwicklungen die Innovationen.

Das betrachtete *CommSy*-Projekt kann als ein gutes Beispiel für *Erfindung*, *Entwicklung* und *Innovation* angesehen werden, da es alle drei Elemente anschaulich wiedergibt (vergleiche hierzu insbesondere die Darstellung des Projektes im Anhang C). Das Projekt *hamburg.de* stützt sich nicht auf eine bahnbrechende *Erfindung*. Es beruht hauptsächlich auf der *Übertragung* (vergleiche Kapitel 5.2.2) von Konzepten, die andere städtische Websites bereits entwickelt hatten. Dennoch ist das Konzept in seinem Kontext neu und erfordert auf verschiedenen Ebenen eine Vielzahl von Erfindungen. Diese sind regelmäßig durch *Entwicklungen* und notwendige *Innovationen* gefolgt. Das *ifu*-Projekt bedarf der wiederholten Anwendung der drei Phasen auf einer Detailebene. Obwohl in dem Projekt das Konzept der *Übertragung* (vergleiche Kapitel 5.2.2) im Mittelpunkt steht, müssen für die unterschiedlichen Umgebungen, in die übertragen werden soll, wiederholt kleine *Erfindungen*, *Entwicklungen* und *Innovationen* stattfinden.

## 5.2.2 Übertragung

Der Begriff der *Übertragung* kennzeichnet sowohl die Möglichkeit, eine bestehende Informations-Infrastruktur in einen neuen, ähnlichen, Einsatzkontext zu übernehmen als auch eine bereits existierende Infrastruktur in einem geänderten Kontext (z. B. politisch) an die neuen Rahmenbedingungen anzupassen. Weber weist in diesem Zusammenhang insbesondere auf die wirtschaftliche Notwendigkeit der Übertragung hin und stellt Infrastruktur selbst als einen möglichen Weg zur Lösung dar:

„Erkennbar ist jedoch, daß nicht jede neue Anwendung eine für sie spezifi-

sche Software-Infrastruktur erfordert, sondern jeweils ganze Klassen von Anwendungen durch eine Art von Software-Infrastruktur vereinheitlicht werden können. [...] Langfristentwicklungen [sind] insbesondere dadurch zu unterstützen, daß Evolutionsprinzipien zur Anwendung kommen.“

(Weber 1999, S. 24)

Im Kontext der beschriebenen empirischen Projekte ist die *ifu* ein Beispiel dafür, wie eine etablierte Infrastruktur (am Arbeitsbereich SWT) auf einen neuen Kontext (am Hauptcampus) übertragen wurde. Ebenso übertragen wurden Techniken für die Bereitstellung der Infrastruktur. So ist z. B. die Installation in den verschiedenen Organisationen durch Übertragung effizienter Konzepte verbessert worden. Das *CommSy* ist ein Beispiel dafür, wie wegen ähnlicher Lehr-Lern-Situationen die technische Unterstützung auf verschiedene Anwendungssituationen übertragen werden konnte. Die Übertragung fand dabei zuerst in Lehrveranstaltungen persönlich bekannter Akteure statt und wurde durch den öffentlichen Server in Veranstaltungen fremder Akteure übertragen. Das Projekt *hamburg.de* fußt auf einer konzeptionellen Übertragung. Erfahrungen aus anderen städtischen Website-Projekten wurden aufgegriffen. Allerdings wurden auf einer technischen Ebene bei *hamburg.de* keine Übertragungen genutzt, sodass Fehlschläge gerade hierauf zurückzuführen sein können (“not invented here”-Problem).

### 5.2.3 Wachstum, Wettstreit und Konsolidierung

*Wachstum* findet statt, wenn ein erfolgreiches Konzept neue Akteure anzieht. Es ist zu bedenken, dass im Wachstum einer Infrastruktur die Anzahl der Wechselbeziehungen stärker als linear steigt, denn die neu hinzugefügten Bestandteile stehen vielfach mit mehr als einem existierenden in Verbindung:

“When these components are brought together into a larger unit, they become interdependent. When one of them is changed, for whatever reason, the others will often need to be changed as well.” (Hanseth 2000, S. 59)

Im *Wettstreit* stehen sich verschiedene Infrastrukturen gegenüber. Standards (s. u.) können für den Wettstreit entscheidend sein (vergleiche Spurweite bei Eisenbahnen, Kapitel 2.1.1, Kommunikationsprotokolle Abbate 1994, S. 197). Erreicht die Anzahl der Akteure, die sich einem Standard anschließen, eine kritische Masse, wenden sich zunehmend Akteure von alternativen Standards ab (Kahin und Abbate 1995).

Die Anzahl der verwendeten Bestandteile und Standards erschwert aber gerade die *Konsolidierung*, da hier bei der Umgestaltung eines Bestandteils alle in Wechselbeziehung stehenden berücksichtigt werden müssen und deren erwartetes Verhalten in Betracht gezogen werden muss.

Im Kontext des *hamburg.de*-Projektes, das im untersuchten Zeitraum selbst kein direktes Beispiel für die Punkte Wachstum, Wettstreit und Konsolidierung bietet, können andere Free-E-Mail-Anbieter genannt werden. Sie haben alle die Phase des Wachstums durchschritten und mit den technischen Problemen und der Skalierbarkeit gekämpft. Der zwangsläufige Wettstreit um Anwender und damit Werbeeinnahmen wurde gefolgt von einer Phase der Konsolidierung (Konzentration durch Kauf, Standardisierung des Leistungsspektrums etc.). *hamburg.de* läuft hier außer Konkurrenz, da die Existenz als Free-E-Mail-Anbieter politisch motiviert ist. Allerdings misst sich das angebotene Leistungsspektrum sehr wohl an den anderen Anbietern. Das *CommSy* durchläuft die Phasen Wachstum, Wettstreit und Konsolidierung auf den verschiedenen Ebenen. Das Wachstum hat im Bereich der Funktionalität stattgefunden und die Anzahl der Anwendungskontexte ist angewachsen. Ein Wettstreit findet zwischen konkurrierenden Lehr-Lern-Plattformen statt. Ebenso ist das Streben nach Konsolidierung im technischen Bereich und bei der Benutzerberatung zu erkennen. Im Projekt *ifu* konnten diese LTS-Phasen nicht beobachtet werden. Dies liegt an der Kürze des Projektes.

#### 5.2.4 Beharrungsvermögen und radikale Änderungen

Einmal installierte Infrastrukturen oder Bestandteile derselben besitzen ein inhärentes *Beharrungsvermögen*. Dies fällt unter den von Hughes gewählten Begriff *Triebkraft* (momentum). Durch die reine Existenz eines Bestandteils wird eine bestimmte Richtung in der Entwicklung vorgegeben, die nur durch massiven Eingriff, eine *radikale Änderung*, nachträglich geändert werden kann. Im Kontext großer technischer Systeme heißt dies:

“Only a historic event of large proportions could deflect or break the momentum.”  
(Hughes 1994)

Aus dem Blickwinkel der Informations-Infrastrukturen lässt sich dies wie folgt formulieren:

“[...] the accumulating resistance against change, the tight inter-connection between different parts of an II including the entangled relationships among

the standards and the dynamic and contingent alternation between stabilising and changing a standard [...]” (Monteiro et al. 1996, S. 419)

Mit steigender Größe einer Infrastruktur wächst deren Unumkehrbarkeit an. Ein Beispiel hierfür ist das Internet, dessen Grundlagen nur noch schwer geändert werden können. Teilweise werden diese Änderungsbedürfnisse paradoxerweise durch die Größe evoziert (Hanseth und Monteiro 1998a, S. 29). Andererseits sind größere Veränderungen notwendig, um das Beharrungsvermögen zu brechen und um gerade Änderungen durchzusetzen (Monteiro et al. 1996, S. 421).

Im Projekt *ifu* stellten die Räume und die technische Ausstattung, die für das Vorhaben ausgeliehen wurden, einen Bestandteil der Infrastruktur dar, der ein starkes *Beharrungsvermögen* besaß. In der Konsequenz musste man mit diesen Umständen umgehen (z. B. deutsche Tastaturen, Öffnungszeiten, Lage) und Nachteile akzeptieren. Bei der Entwicklung des *Comm-Sy* entstand ein Beharrungsvermögen durch die Entwickler selbst, indem sie einerseits ihre Interessen verfolgten und andere Entwicklungsvorhaben mit wenig Priorität betrieben. Andererseits entstand Beharrungsvermögen durch die technische Konfiguration in den Zeiträumen, in denen nur wenige Entwickler zur Verfügung standen, da damit der Status quo erhalten blieb. Das Beharrungsvermögen hat sich im Projekt *hamburg.de* am Beispiel der Webmail-Realisierung gezeigt. Hier besaß die für die E-Mail vorgenommene Basis-Installation das *Beharrungsvermögen*, sodass andere Anforderungen daran ausgerichtet werden mussten.

### 5.2.5 Konsequenzen für Software-Infrastrukturen

Große technische Systeme besitzen eine ähnliche Komplexität wie Infrastrukturen in Bezug auf die Vielzahl der beteiligten Akteure, der relevanten Technik, Standards usw. Der direkte Vergleich der Infrastruktur-Projekte mit großen technischen Systemen zeigt eine hohe Ähnlichkeit im Verhalten und in den Eigenschaften und unterstreicht die Nützlichkeit von Konzepten, die zur Benennung der Phasen im Rückblick verwendet werden können. Die Phasen Wachstum, Wettstreit und Konsolidierung lassen sich z. B. der Kerneigenschaft *Entwicklungsfähigkeit* zuordnen, wohingegen Beharrungsvermögen der Gegenspieler von *Entwicklungsfähigkeit* ist. Radikale Änderungen und Wettstreit können durch störende Wechselwirkungen, In-



terferenzen (Kapitel 5.5.2), ausgelöst werden. Übertragung kann als eine Ausprägung der *Reproduzierbarkeit* angesehen werden.

Aus diesen Kongruenzen lässt sich umgekehrt ableiten, dass ein Software-Entwicklungsprozess für Infrastrukturen mit den Eigenschaften und Effekten aus dem Bereich großer technischer Systeme umgehen können muss. Es ist deshalb nützlich, deren Konzepte in den Infrastruktur-Diskurs zu übernehmen. Ungeachtet dessen verbleibt als Defizit, dass damit kein anleitendes Entwicklungsverständnis vorliegt. Alle LTS-Konzepte können lediglich helfen, den aktuellen Stand zu beschreiben und damit besser zu verstehen.

### 5.3 Software-Infrastruktur als Artefakt

Mit „Software-Infrastruktur als Artefakt“ wird eine Sicht eingenommen, bei der die Grundannahme die Trennung von Entwickler, Artefakt und Benutzer ist. Damit wird der allein stehende Gegenstand betont und die Beziehung zu anderen tritt in den Hintergrund. Im Artefakt werden Grundannahmen der Entwickler vergegenständlicht, die nur innerhalb des gleichen Kulturkreises Gültigkeit haben und problemlos genutzt werden können. Das Artefakt kann in andere Kulturkreise eingebracht werden („immutable mobile“, s. S. 55), muss dort aber nicht verstanden werden. Eigenschaft des Artefaktes ist es, dass die kontextfreie Produktion eines identischen Gegenstandes beliebig wiederholt werden kann. Software-Infrastruktur als Artefakt zu sehen unterstellt die Produktion einer Infrastruktur ohne Beteiligung der Benutzer.

Aus dieser Perspektive lässt sich eine Vernetztheit zwischen Entwicklern, Betreibern und Benutzern, die sich aus den Projekterfahrungen im Kontext von Infrastrukturen als essenziell herausgestellt hat, nur schwer beschreiben. Andererseits bietet die Sicht Einblicke in den Versuch, Unabhängigkeit zwischen den beteiligten Gruppen herzustellen, und kann insofern hilfreiche Hinweise darauf liefern, wie in einem Entwicklungsprozess notwendige Unabhängigkeit hergestellt werden kann. Sie ist wichtig, weil Artefakte grundlegend für das alltägliche Leben und Handeln sind.

Zur Artefakt-Sicht gehören die Eigenschaften Standardisiertheit, Kopplungsfähigkeit und Geschichtlichkeit, die in den folgenden Abschnitten betrachtet werden.

### 5.3.1 Standardisiertheit

*Standards* vergegenständlichen Wissen über Zusammenhänge und erlauben die Zusammenarbeit auf der Grundlage geringer Aushandlung zwischen den Beteiligten. Standards spiegeln eine Kultur wider. Das hier zu Grunde gelegte Verständnis von Kultur stützt sich auf einen semiotischen Kulturbegriff ab. Kultur wird als ineinander greifende Systeme auslegbarer Zeichen verstanden (Geertz 1983). Standards definieren einen Teil der Beziehung zwischen den an der Infrastruktur beteiligten Aktanten. Dies wird unterstrichen durch die Feststellung, dass Infrastruktur als eine *Relation* aufzufassen sei. Verbunden mit Standards und Relationen ist die Frage nach dem Wissen um diese Dinge. Es geht darum, Standards und Relationen zu kennen, sie einsetzen zu können und den Umgang mit ihnen zu vermitteln. Dies soll durch den Begriff *Kompetenz* zusammengefasst werden.

Standards können durch Definitionen, Verträge o. ä. etabliert werden. Im Bereich der Informations-Infrastrukturen sind dafür Normen (z. B. DIN, ISO) und Kommentaranfragen (so genannte RFCs) üblich. Standards werden nicht *per se* durch den Willen einer Organisation zum Standard<sup>1</sup> definiert, sondern werden durch die Anzahl relevanter Benutzer begründet. Gleichzeitig steigt das Beharrungsvermögen:

“As the number of users as well as the types of users grows, reaching agreement on changes becomes more difficult (Steinberg 1995).”

(Monteiro et al. 1996, S. 420)

Die Etablierung von Standards ist ein separates Forschungsfeld in den Wirtschaftswissenschaften (Kahin und Abbate 1995). Die Definition von Standards steht immer im Spannungsfeld zwischen Einfachheit und der Aufnahme von Interessen aller Beteiligten (“conflicting needs”, vergleiche Bowker und Star 1999). Damit verbunden ist die Frage von Macht und Machtausübung.

“The control of standards is a central, often underanalyzed feature of economic life [...]. It is key to knowledge production as well.”

(Bowker und Star 1999, S. 15)

---

1 “Some institutions, like standardization bodies, are set up in order to try to coordinate such processes. But these institutions have hardly any authority or power to enforce any kind of behaviour on the individual actors.” (Hanseth 2000, S. 60)

Ein Standard wird somit sowohl zum Vor- als auch zum Nachteil, da einerseits die Möglichkeiten zur Zusammenarbeit steigen und andererseits die Möglichkeit zur Änderung geringer wird (vergleiche Beharrungsvermögen). Es existiert das Dilemma, dass Standards gebraucht werden, um die notwendige Systemkomplexität handhaben zu können:

“[...] the complex, geographically dispersed and strongly inter-connected character of an II generates a strong need for standardisation and accumulates resistance against further modifications. At the same time one has to prepare for changes.” (Monteiro et al. 1996, S. 408)

Ein Versuch, die Anzahl der zu betrachtenden Standards zu reduzieren, besteht darin, wiederholt die Möglichkeit des Kapselns zu nutzen. Dieses Vorgehen wird z. B. bei Netzwerken praktiziert (Monteiro et al. 1996, S. 412) und bei Infrastrukturen angewendet, die sich auf andere Netze abstützen.

Das *hamburg.de*-Projekt konnte durch die Einhaltung von (technischen) Standards (z. B. HTTP, SMTP, POP3, XML) Dienstleistungen zur Verfügung stellen, die sofort von einer Vielzahl von Anwendern verwendet werden konnten und untereinander kombinierbar waren. Das *CommSy* dient als Beispiel für Konventionen des Handelns in pädagogischer Praxis (z. B. projektorientierte Lehre), die es ermöglichten, das *CommSy*-Produkt in verschiedenen Kontexten einzusetzen. Auf der technischen Ebene wurde das Ziel verfolgt, möglichst wenige und schmale Standards zu verlangen (Reduzierung auf HTML) und andererseits vielen Standards gegenüber offen zu sein (vergleiche Attachments). Im *ifu*-Projekt ist das Dilemma kultureller Gegensätze aufgetreten (z. B. unterschiedliche und teilweise gegenläufige Papierformate, Schriften, Arbeitsgewohnheiten). Diesen verschiedenen Anforderungen konnte man in vollem Umfang nicht gerecht werden. Auf der anderen Seite haben z. B. Dokumentformate (Microsoft Office, PDF, PS) eine breite Basis für die Zusammenarbeit geliefert.

Neben den Standards selbst und den *Relationen*, die mit ihrer Hilfe definiert werden können, üben die beteiligten Personen *Kompetenzen* aus. Vereinheitlichungen (Standards) werden geschaffen, um Kompetenzen darin zu vergegenständlichen und die Kompetenz zu besitzen, diese Standards einzusetzen. Einher damit geht das Problem, die etablierten Standards und deren Möglichkeiten zu kommunizieren (Kubicek und Schmid 1996, S. 34). Dies tangiert Aspekte des Lernens und die Unterscheidung zwischen Innen- und Außensicht (s. u.).

Weber fordert u. a. ein „ganzes Bündel von Kompetenzen“, welches „die Kompetenz des Technologie- und Systemintegrators zur Zusammenführung von Einzeltechnologien zu einer angepassten Gesamttechnologie“ umfasst (Weber 1999, S. 19-20). Und Floyd weist in diesem Zusammenhang darauf hin, dass bei der Entwicklung von Software-Infrastrukturen so komplexe Gebilde entstehen, dass nicht mehr jedermann die Kompetenz besitzt, diese voll zu verstehen (Floyd 2002, S. 5-28).

Im Projekt *hamburg.de* steht als Problem die Frage nach Kompetenzen im Vordergrund. Da das Projekt zur Blütezeit des Internets begann, war es schwierig, ausreichend qualifizierte und kostengünstige Spezialisten zu bekommen, die mit der verwendeten Technik und den Standards umgehen konnten. Das *ifu*-Projekt besaß das Problem, die entsprechenden Kompetenzen zur Nutzung der bereitgestellten Infrastruktur den Personen aus unterschiedlichen Kulturen in sehr kurzer Zeit beizubringen. Gleichzeitig brachten sie sehr unterschiedliche kulturell geprägte Kompetenzen mit und mussten diese in Beziehung zur existierenden Basis setzen. Im Gegensatz dazu setzt das *CommSy* absichtlich auf wenigen technischen Standards auf Seiten des Benutzers auf. Dies ist Teil eines Konzepts, bei dem Lerninhalte und der Umgang mit Technik im Vordergrund stehen sollen. Pädagogische Konventionen im Bereich der projektorientierten Lehre werden vorausgesetzt, deren Etablierung nicht in dem Maße betrieben wird, wie es bei technischen Standards der Fall ist. Technische Standards auf Seiten des Servers (Open-Source-Technologien) stellen die Flexibilität sicher.

### 5.3.2 Kopplungsfähigkeit

*Gateways* dienen auf technischer Ebene dazu, Infrastrukturen zwischen Organisationen zu verknüpfen. Mithilfe der Gateway-Technik können Verbindungen zwischen zwei technischen Komponenten mediiert werden, d. h. von einer Ausgangssprache in eine Zielsprache übersetzt werden. Damit können auf der organisatorischen Ebene z. B. Anforderungen der zweiten Organisation zum Großteil durch das Gateway realisiert werden, ohne dass dies die erste betrifft und womöglich Rückwirkungen auf weitere mit der ersten verbundene haben könnte. Gateways eignen sich ebenfalls, die durch Organisationen eingebrachte Komplexität zu reduzieren. Diese durch Gateways erreichte Entkopplung wird z. B. im Internet als Konzept praktiziert (Abbate 1994, S. 205-206). Dort werden Gateways vielfach in Tran-

sitionsprozessen eingesetzt (Wechsel von einer Protokollversion zur nächsten). Hanseth und Monteiro weisen auf weitere Vorteile von Gateways hin: Verbindung inkompatibler Netze, Entscheidungen hinausschieben, Flexibilität erhöhen (Hanseth und Monteiro 1998a, S. 31).

Die Schnittstelle einer Infrastruktur kennzeichnet den nach außen sichtbaren Teil der Infrastruktur. Dies sind im Bereich Wasserversorgung z. B. der Querschnitt von Rohren und die definierten Wasserdruckbereiche, im Bereich von Straßennetzen z. B. die Fahrbahnbreite, Verkehrszeichen und die Verkehrsgeschwindigkeit. Die Schnittstelle ist gleichzeitig durch ihre Interpretation kulturell geprägt: Während z. B. in dem einen Kulturkreis eine rote Ampel zum unbedingten Halten zwingt, wird diese in anderen nicht so absolut interpretiert und erlaubt durchaus das vorsichtige Weiterfahren.

Aus technischer Sicht wird die Schnittstelle über die formalen Parameter begrenzt. In diesem Zusammenhang wird deshalb der Protokollbegriff eingeführt, um den zeitlichen Aspekt zusätzlich zu berücksichtigen. Ein Protokoll baut auf einer Schnittstelle auf und definiert die zeitlichen Zusammenhänge von Aktionen (den Kontext).

“If networks are to function together as an internet they must have some common procedures or protocols to handle addressing, routing, and end-to-end reliability. [...] The choice between having a uniform network or internet protocol represents a trade-off between the simplicity and efficiency of having uniform networks and the flexibility of allowing diverse ones. [...] (In their view, interfacing heterogeneous networks was not just a stopgap measure until networks could be made uniform but permanent necessity).”

(Abbate 1994, S. 199-200)

Gateways arbeiten auf definierten Schnittstellen und Protokollen und liefern selbst ebensolche nach außen. Dadurch können grundsätzlich konkurrierende Dinge miteinander verbunden werden. Machtgefüge, die darin bestehen, dass ein Standard angeboten wird, können durch den Einsatz von Gateways verändert werden. Dadurch bietet das Konzept eines Gateways ein Potenzial bei der Überwindung von Beharrungsvermögen.

Im Projekt *hamburg.de* wurde ein Gateway eingesetzt, um den existierenden E-Mail-Server für die Implementation von Webmail zu benutzen. Das Produkt *Courier IMAP-Server* arbeitet auf demselben Postfachformat wie die bestehende Realisierung. Dadurch konnte die Webmail-Applikation auf der Basis des IMAP-Protokolls angebunden werden, ohne dass die bestehende E-Mail-Einrichtung geändert werden musste. Insgesamt erhält

man in der Infrastruktur einen höheren Grad an Freiheit, da sowohl der E-Mail-Server, der IMAP-Server und die Webmail-Applikation ausgetauscht werden können, ohne die jeweils anderen Bestandteile zu tangieren.

Bei der *ifu* wurde eine Kopplungsfähigkeit erwartet, die nicht antizipiert worden war. Dies betraf die Kopplung der *ifu*-Infrastruktur mit der Infrastruktur der jeweiligen Studierenden bzw. Lehrenden. Einige gingen von einer geringen Kopplung aus und hatten ihren notwendigen Ausschnitt der eigenen jeweiligen Infrastruktur mitgebracht (Laptop, Dokumente, Datei usw.). Andere erwarteten die Möglichkeit, auf ihre entfernte Infrastruktur mittels der lokalen zugreifen zu können. Je nach Unterschiedlichkeit der involvierten Bestandteile ging dies besser oder schlechter.

Das *CommSy* überlässt den Benutzern die Aushandlung der Kopplung zu ihrer eigenen Infrastruktur, indem es als Kopplungsmedium HTML und Dateianhänge anbietet. Die Verwendung beider muss in der Benutzergruppe jeweils neu etabliert werden.

### 5.3.3 Geschichtlichkeit

Eine Eigenschaft von Infrastruktur ist, dass sie auf einer existierenden Basis aufsetzt (“installed base”, s. S. 53). Weiterentwicklungen finden ausgehend von der derzeit bestehenden Basis statt und setzen auf dieser auf. Sie legen sich vergleichbar mit Sedimentschichten übereinander und dokumentieren einen Teil der Entwicklungsgeschichte. Diese darüber hinausgehende Eigenschaft bezeichne ich mit *Geschichtlichkeit*. Die existierende Basis beeinflusst, wie das Neue entwickelt werden kann. Unter der Annahme von Infrastruktur als eine “installed base” folgt, dass Infrastrukturen immer bereits existieren, wenn man sie als solche betrachtet. Eine Infrastruktur kann nicht aus dem Nichts bzw. nicht sofort als Infrastruktur entworfen werden (Hanseth 2000, S. 60).

Mit der Weiterentwicklung verbunden treten Probleme auf, die beim Übergang von Bestehendem zu Neuem offenkundig werden. Das Neue muss eine Form des Übergangs von der alten Infrastruktur anbieten (Hanseth 2000, S. 67). Man kann zwei Weiterentwicklungsstrategien für den Umgang mit Geschichtlichkeit angeben, die es vermeiden in einem Stillstand zu bleiben: Evolution oder Revolution. Bei der evolutionären Strategie wird Kompatibilität zum Bestehenden gewährleistet, indem Einbußen in der Entwicklungsgeschwindigkeit hingenommen werden. Durch Warten auf die

jeweils individuellen Umstiegszeitpunkte werden Aufwendungen für den Umstieg minimiert. Die revolutionäre Strategie erlaubt das zügige Einführen von Neuerungen unter Vernachlässigung der bestehenden Benutzer. Das birgt das Risiko, dass der Umstieg vollständig fehlschlägt. Nur mächtige Institutionen und einflussreiche Verbündete können zum Gelingen dieses Ansatzes beitragen.

Die am meisten praktizierte, die evolutionäre, Strategie kann in langsame und zügige Weiterentwicklung unterschieden werden. Während die langsame Weiterentwicklung Kompatibilität zu Altem aufrechterhält und damit ein immer komplexer werdendes Netz entwickelt, zielt die zügige Weiterentwicklung auf die Etablierung einer neuen Infrastruktur, die mit Gateways an die alte – größtenteils nicht mehr gepflegte – Infrastruktur angekoppelt wird. Durch die Ankopplung existieren für einen begrenzten Zeitraum zwei Infrastrukturen mit verwandter Leistung. Das Risiko einer Fehlentwicklung wird hierdurch minimiert und gleichzeitig der Umstiegsdruck auf Anwender durch Fortschritt motiviert.

Bei *hamburg.de* ist die Geschichtlichkeit durch die bereits existierenden Lösungen im Bereich öffentliche Webseiten und E-Mail-Adressen von Behörden erkennbar (Summe vieler kleiner existierender Speziallösungen). Bestehende Nutzungsmuster und Möglichkeiten mussten ebenso unterstützt werden wie verbreitete Werkzeuge (verschiedene Web-Browser-Versionen). Im Projekt *ifu* wurde durch die gemieteten Räume eine Geschichtlichkeit in das Projekt übernommen, deren Veränderung nicht im eigenen Einflussbereich lag. Die Vorgeschichte z. B. der technischen Ausstattung, der Räume und der betreuenden Personen wirkt auf die neu aufzubauende Infrastruktur. Entscheidungen und Entwicklungsspielräume wurden daran ausgerichtet. *CommSy* beruht auf etablierten Mustern, wie projektorientierte Lehre durchzuführen ist. Deren Vorgeschichte liegt in der Arbeit von zwei Arbeitsbereichen. Auf der technischen Seite stützt sich *CommSy* auf die Vorgeschichte der Ausstattung eines Arbeitsbereichs ab. Das *CommSy* stößt an Grenzen bei der Weiterentwicklung, wenn es darum geht, bestehende Projekträume in neue Versionen zu überführen. Wiederholt müssen alte Datenbestände in neue Formate überführt werden und verlangsamen so den Weiterentwicklungsprozess.

### 5.3.4 Konsequenzen für Software-Infrastrukturen

Die Artefakt-Sicht auf Infrastruktur betont die Kulturalität. Schnittstellen, Protokolle und Standards sind ein Ausdruck von Kultur und erlauben es, eine Grundlage für die Beziehung verschiedener Aktanten zu legen. Das Abstützen auf Standards erfordert von den Beteiligten eine ausreichende Kompetenz im Umgang untereinander, um die relevanten Standards, Schnittstellen und Protokolle zu kennen und konform zu diesen zu interagieren. Kulturelle Unterschiede können elegant z. B. durch Gateways übersetzt werden, indem das Gateway den in einem Kulturkreis verbreiteten Standard (z. B. TCP/IP) in den eines anderen (z. B. X.25) transformiert.

Die explizite Etablierung einer kulturellen Basis erlaubt die getrennte Entwicklung von Produkten, die potenziell miteinander verknüpft werden können. Die Protokolle sind damit für die Prozess-Sicht (vergleiche Kapitel 5.5) und für den Verlauf eines Software-Entwicklungsprozesses wichtig. Kulturelle Standards und der Umgang damit sind von mir unter dem Begriff *Standardisiertheit* zusammengefasst. Die Einhaltung von Standards und deren konsequente Anwendung erhöht die *Alltagstauglichkeit* einer Infrastruktur, indem die verlässliche Basis für Entwickler und Benutzer vergrößert wird.

Ein Entwicklungsprozess für Software-Infrastrukturen muss mit der Geschichtlichkeit zurechtkommen und diese anerkennen. Damit wird von der existierenden Basis und der aufzubauenden Infrastruktur verlangt, dass sie der Eigenschaft *Aufbaubarkeit* genügt. Gateways stellen ein adäquates Mittel dar, um mit Geschichtlichkeit umzugehen. Mithilfe von Gateways können durch Geschichtlichkeit erwachsene Beschränkungen ausgeräumt und die *Entwicklungsfähigkeit* einer Infrastruktur gestärkt werden. Darüber hinaus erhöhen Gateways die *Konfigurierbarkeit* einer Infrastruktur, ohne dass Eingriffe in den Kern notwendig werden.

Die Artefakt-Sichtweise lässt eine Orientierung an den Akteuren vermissen. Damit blendet sie verschiedene Entwickler- und verschiedene Benutzer-Perspektiven aus und unterscheidet insbesondere deren verschiedene Perspektiven (Innen- und Außensicht) nicht (vergleiche Kapitel 5.4.3). Der nächste Abschnitt arbeitet die Wichtigkeit von Akteuren und deren Diversität heraus.



## 5.4 Software-Infrastruktur-Akteure

Mit dieser Sicht wird der Schwerpunkt auf handelnde Personen gelegt. Die Akteurs-Sicht unterstreicht die Perspektivität bei der Infrastruktur-Entwicklung. Hiermit tritt einerseits das Subjekt – die an der Entwicklung und Nutzung beteiligten Personen – in den Vordergrund. Andererseits bringt die Akteur-Netzwerk-Theorie die Betrachtung von nicht-menschlichen Dingen als Aktant in das Konzept hinein, welches durch Begriffe wie z. B. *unveränderliches Bewegliches* und *Beharrungsvermögen* aus der Diskussion um große technische Systeme unterstrichen wird. Damit lassen sich soziale Akteure und Dinge auf gleicher Ebene behandeln. Dies war nicht wichtig, solange Software als Produkt gesehen wurde, der Infrastruktur-Charakter von Software erfordert es aber, den Kontext reichhaltiger zu erschließen (Kling 1992, S. 4) und keine künstliche Trennung von eng miteinander verwobenen Aktanten zu erzeugen (Latour 1996b, S. 302).

### 5.4.1 Akteure und Politik

Im Kontext von Infrastrukturen sind Akteure relevant, die bereits aus der Software-Entwicklung bekannt sind (beispielsweise Software-Entwickler, Auftraggeber, Benutzer, Hardware-Betreiber – meist aus demselben Unternehmen). Hinzu kommen aufgrund der organisationsübergreifenden Anlage des Infrastruktur-Konzeptes weitere Akteure (unterschiedliche Auftraggeber, Gruppen von Software-Entwicklern, anonyme Benutzer, verschiedene Hardware-Betreiber, Dienstleister – aus unterschiedlichen Organisationen). Die empirischen Fälle haben gezeigt, dass deren Rolle in der Infrastruktur-Entwicklung wichtig ist. Insgesamt ist die Anzahl der beteiligten Akteure in Infrastruktur-Projekten signifikant höher als bei klassischer Anwendungsentwicklung. Die Akteur-Netzwerk-Theorie legt nahe, alle beteiligten Menschen, Organisationen und Dinge als Aktanten zu betrachten (vergleiche Kapitel 3.3.3). Sie bietet Mittel, um in dem daraus entstehenden Netz zu gruppieren und vergrößern. Damit werden Personen, Organisationen, Geräte, Installationen, Standards, Klassifikationsschemata usw. zu Aktanten.

Ciborra und Hanseth verweisen darauf, dass die Steuerung von Infrastruktur-Entwicklung ein signifikantes Problem ist, gerade weil eine Vielzahl von Interessengruppen mit konfligierenden Absichten beteiligt ist (Ciborra und Hanseth 1998, S. 265). Dies kann zu einer expandierenden In-

frastruktur führen, die in verschiedene Richtungen wächst, die größtenteils außerhalb der Kontrolle der verschiedenen Beteiligten ist. Dabei darf nicht vernachlässigt werden, dass die Infrastruktur selbst als Akteur auftritt, der sein Fortbestehen in Ausbreitung, Umfang und Form bestimmt (Beharrungsvermögen).

Genau diesen existierenden Entwicklungsprozess gilt es aus dem Blickwinkel der Software-Entwicklung zu betreiben. Hierbei sind Interessenkonflikte und unterschiedliche Wahrnehmungen zu erwarten (Abbate 1994, S. 194). Diese zu erkennen, damit umzugehen und sie konstruktiv zu wenden ist Kriterium für einen guten Software-Entwicklungsprozess.

Der Begriff „Politik“ verweist im Zusammenhang mit Akteuren sowohl auf lenkendes und steuerndes Eingreifen von politischen Organisationen (Regierungen, Ministerien usw.) wie auch von Nicht-Regierungs-Organisationen (z. B. Standardisierungsgremien). Er verdeutlicht gleichzeitig, dass neben den rein sachlichen Interessen ebenso Machtinteressen ausgeübt werden. Im politischen Bereich hat sich z. B. die Regierung der USA öffentlichkeitswirksam engagiert:

“The White House formed the Information Infrastructure Task Force (IITF) in 1993 to articulate and implement the Administration’s vision for the National Information Infrastructure (NII). The task force consisted of high-level representatives of the Federal agencies that play a major role in the development and application of information and telecommunications technologies.”  
(White House 2000)

Die Etablierung von Standards hat beispielsweise im Telekommunikationssektor gezeigt, dass es hierbei ebenfalls um einen politischen Prozess geht. Im Falle der Protokolle X.25 und TCP/IP haben diese jeweils in den Regionen, in denen sie verwendet wurden, Ansprüche angemeldet: Durch die Verwendung des Standards wurde signifikanter Druck auf die Zulieferer ausgeübt, entsprechende standardkonforme Hardware herzustellen (Abbate 1994, S. 198-199). Eingeführte Standards können somit maßgeblich auf wirtschaftliche und gesellschaftliche Entwicklungen Einfluss nehmen.

Im Projekt *hamburg.de* brachte die hohe Anzahl von beteiligten Akteuren (vergleiche Anhang B) verschiedene Interessen, Ziele und Vorgehensweisen in den Entwicklungsprozess ein. Es war problematisch, diese auszurichten und miteinander in Beziehung zu setzen. Im Projekt *ifu* wurden Teile der Infrastruktur durch das Unterprojekt *vifu* bereitgestellt, das nicht allein Ziele verfolgte, die dem Projektbereich Information dienten (verglei-

che Anhang A). Konfligierende Entscheidungen und Zeitpläne führten zu der Überlegung, sich von dem Unterprojekt unabhängig zu machen. Diese potenziell aufwendige Trennung wurde durch Gespräche und Interessenausgleich vermieden. Die Entwicklung des *CommSy*-Projekts wurde durch die Interessen der Entwickler gelenkt. In Abhängigkeit der jeweils im Fokus einzelner Entwickler stehenden Einsatzkontexte wurde das System passend weiterentwickelt. Das führte zeitweise zu gegenläufigen oder widersprüchlichen Entwicklungen.

### 5.4.2 Grenzüberschreitung zwischen Organisationen

Der Aspekt der organisationellen Grenzüberschreitung ist für mein Verständnis von Infrastruktur wesentlich. Sei es, dass Teile einer innerhalb einer Organisation etablierten Infrastruktur durch Zusammenarbeit mit anderen Organisationen in diese hineinragen (vergleiche hierzu Projekträume im *CommSy*-Kontext) oder dass es um den Aufbau einer Infrastruktur über Organisationsgrenzen hinweg geht (vergleiche *ifu*-Projekt).

Die Entwicklung einer Informations-Infrastruktur erfolgt eher anhand von Praxisgemeinschaften und ohne Rücksicht auf Organisationsgrenzen. Entsprechende Prozessmodelle können sich also nicht nur an einer Organisation orientieren. Damit treten Probleme in den Bereichen Zuständigkeiten, Entscheidungsspielräume, Politik u. a. hervor.

Kling benennt die organisationellen Veränderungen explizit, die durch Einführung von Infrastrukturen hervorgerufen werden. Er konstatiert, dass man als Entwickler typischerweise mit institutionell etablierten Praktiken, Mustern organisationeller Kontrolle und durch Arbeitskultur gebildeten Bedeutungszusammenhängen konfrontiert wird (Kling 1992, S. 35).

Die vielfach eingesetzte methodische Herangehensweise in diesem Zusammenhang ist die Akteur-Netzwerk-Theorie. Hanseth und Monteiro weisen auf die Vorteile und gleichzeitig auf die unbeantworteten Fragen hin:

“ANT helps us understand how and why everything in principle might change and how changes takes [sic] place at a micro level. It does not, however, help us much in understanding why planned change[s] so often fail and the fact that existing structures appears [sic] to be irresistible to any planned change efforts. [...] The new institutionalism is based upon social constructivism as well as ethnomethodology. [...] The key insight is the way stability of institutions are accounted for.”

(Hanseth und Monteiro 1998a, S. 27)

Das *CommSy*-Projekt und das *ifu*-Projekt haben anschaulich gezeigt, wie eine Infrastruktur über Organisationsgrenzen hinweg entwickelt und betrieben werden kann und welche Herausforderungen damit verbunden sind. Beim *CommSy* musste in jedem neuen Einsatzkontext sichergestellt werden, dass die Grundannahmen über projektorientierte Lehre passten und dass die minimalen technischen Kenntnisse vorhanden waren. Jedes neue Projekt zeigte klar die Differenzen zwischen den Praxisgemeinschaften. In der *ifu* wurden auf einer Ebene Studentinnen und Lehrende aus verschiedenen Praxisgemeinschaften zusammengebracht und bildeten orientiert an den Lehreprojekten neue Praxisgemeinschaften. In diese trugen sie ihre etablierten Muster hinein und bildeten gemeinsam neue heraus. Auf einer anderen Ebene wurde die Organisation *ifu* mit den verschiedenen Organisationen des Hauptcampus der Universität Hamburg in Verbindung gebracht. Diese beiden Ebenen haben sich bei der Infrastruktur-Entwicklung überlagert. Im *hamburg.de*-Projekt wurden verschiedene Praxisgemeinschaften durch die beteiligten Projektpartner und die Gesellschafter miteinander verbunden. Das Beispiel E-Mail-Verschlüsselung ist ein Indiz dafür, dass die in einer Praxisgemeinschaft (*hamburg.de*-Steuerungsgruppe) etablierte Nutzung (Verwendung der Verschlüsselung) jeweils in die anderen Kontexte der Beteiligten übertragen wird. Zusätzlich musste sich die von *hamburg.de* zur Verfügung gestellte Infrastruktur in Beziehung zu einer Vielzahl von Praxisgemeinschaften der Benutzer setzen.

### 5.4.3 Innensicht und Außensicht

Das Konzept der *Innensicht* und das der *Außensicht* ist von elementarer Bedeutung im Rahmen von Infrastrukturen. Bisher wurden die unterschiedlichen Standpunkte mit Entwickler-Sicht und Benutzer-Sicht bezeichnet. Während bei partizipativer Software-Entwicklung Wert darauf gelegt wird, dass Entwickler und Anwender in einem gegenseitigen Lern- und Kommunikationsprozess das jeweils Notwendige voneinander lernen (Floyd et al. 1989, Schuler und Namioka 1993), ist dies bei Infrastrukturen so nicht mehr möglich und auch nicht gewünscht. Die Komplexität der verwendeten Technik, deren Anzahl, das organisatorische Geflecht usw. machen es unmöglich, dass alle Beteiligten die Infrastruktur verstehen und ihr Verständnis untereinander abgleichen, da verschiedene Nutzungskontexte vorliegen (im Gegensatz von z. B. Arbeitsplatztypen). Hinzu kommt, dass das

Interesse der Akteure grundlegend verschieden sein kann. Zum Beispiel ist die Administration einer Software bei der Entwicklung kein Schwerpunktthema. Entscheidend ist, dass sich Entwickler ein vollkommen anderes Modell über die Wechselwirkungen in und von der Infrastruktur aufbauen, als dies Anwender tun. Ich möchte den fundamentalen Unterschied zwischen der Sicht von Entwicklern gegenüber der von Benutzern, der eine neue Qualität in die Software-Entwicklung bringt, über das Konzept von *Innensicht* und *Außensicht* ausdrücken.

Die *Innensicht* stellt die Wahrnehmung der Infrastruktur aus Sicht der Erfinder, Systemerbauer und Betreiber dar. Sie stellen sich die Frage: „Was muss ich tun, um eine Infrastruktur für einen bestimmten Kontext aufzubauen?“ Ihr Interesse liegt neben dem Betrieb in der Effizienzsteigerung, dem Vermeiden von Fehlern und nicht zuletzt in der Ästhetik des von ihnen Geschaffenen (vergleiche die von Hughes eingeführten Personen). Sie bauen sich ein Modell von der Infrastruktur über die ihnen bekannten Wechselwirkungen der Bestandteile. Handlungen aus der Perspektive der Innensicht umfassen alle Weiterentwicklungen, um die o. g. Ziele zu verfolgen. Dabei sind die Nutzungskontexte der Benutzer vielfach nicht bekannt und werden nur indirekt über die Reaktionen der Infrastruktur erschlossen. Die Akteure der Innensicht, z. B. Administratoren und Operateure, haben ein Detailverständnis der Infrastruktur, das von keinem anderen erreicht werden kann. Die dafür gebildete Fachsprache orientiert sich an technischen Standards. Zum Aufbau und Betrieb der Infrastruktur benutzen die Akteure der Innensicht andere Infrastrukturen, die sie aus der Außensicht wahrnehmen.

Die *Außensicht* nehmen diejenigen ein, die mit der Infrastruktur arbeiten, Dienstleistung von ihr benötigen und auf indirektem Wege mit ihr verbunden sind (mittelbare Nutzung). Sie stellen sich die Frage: „Welche Infrastruktur brauche ich zum Erledigen meiner Aufgabe?“ Ihr Interesse liegt gerade im *Nicht-Kennen* der Realisierung, des Betriebs und der Dinge, die es braucht, um ein fehlerfreies Funktionieren zu garantieren. Ihr Modell basiert nur auf den erkennbaren und notwendigen Bestandteilen der Infrastruktur. Für Personen mit Außensicht ist die Innensicht irrelevant. Sie haben kein Verständnis für die Ästhetik des Inneren und verfolgen ein von der Infrastruktur unabhängiges Ziel, für dessen Erreichen sie lediglich die Infrastruktur verwenden wollen. Infrastruktur gewinnt an Wert, wenn sie von außen betrachtet wird und indem man ihr Nutzungsmöglichkeiten zuschreibt. Die Haltung lässt sich unter dem Begriff Alltagstauglichkeit

zusammenfassen (Keil-Slawik 2000, S. 212). Eine alltagstaugliche Infrastruktur lässt sich an vier Aspekten erkennen: Ressourcenneutralität, Übertragbarkeit, Offenheit und Nachhaltigkeit. Es sind zur alltäglichen Nutzung keine weiteren Ressourcen notwendig (Ressourcenneutralität), die Nutzung kann von verschiedenen Akteuren in der gleichen Art durchgeführt werden (Übertragbarkeit), die Infrastruktur-Bestandteile können nach freier Wahl miteinander kombiniert werden (Offenheit) und alle Bestandteile stützen sich auf etablierte Standards ab, um in neuen Kontexten weiterverwendet werden zu können (Nachhaltigkeit).

Die analytische Trennung von Innensicht und Außensicht eröffnet neue Erkenntnisse. Sowohl für die Analyse als auch für die Konstruktion ist es notwendig, die Innen- von der Außensicht zu trennen. Gerade an der Grenze zwischen den beiden Sichten manifestieren sich zwei Grundprobleme im Bereich Infrastrukturen: *Bedürfnisantizipation* und *Modellinkongruenz*. Bedürfnisantizipation steht für Aktionen, in denen Personen aus dem Kreis der Systemerbauer sich vorstellen, was die Anwender brauchen (Jeenicke 2001). Durch die Umsetzung werden Veränderungen in die Infrastruktur eingeführt (s. u. Interferenzen), die teilweise etablierte Arbeit mit der Infrastruktur behindern. Modellinkongruenz kann zwischen Beteiligten beider Seiten auftreten und stellt eine besondere Schwierigkeit in der Kommunikation dar (s. u. Kubicek und Schmid). Jede Seite nimmt an, dass die jeweils andere mit demselben Modell über die Infrastruktur arbeiten würde. Eine Strukturähnlichkeit zwischen dem Innen, der inneren Realisation, und dem Außen, der äußeren Wahrnehmung, gibt es aber im Allgemeinen nicht.

Die Innensicht der Infrastruktur im *hamburg.de*-Projekt ist geprägt durch die technische Realisierung, wogegen die Außensicht im Vergleich mit anderen Anbietern mit ähnlicher Leistung entsteht. Im Mittelpunkt der Aktivitäten der Steuerungsgruppe stand die Planung und Entwicklung einer Infrastruktur. Dabei wurden nicht nur die Interessen der allgemeinen Benutzer (z. B. Bürger der Stadt Hamburg) wenig einbezogen, sondern auch die der speziellen Benutzer (z. B. Behördenvertreter). Im letzten Fall scheiterte die Bedürfnisantizipation. Im *CommSy*-Projekt wird mit der Außensicht die pädagogische Nutzung betont, während die Innensicht einen Schwerpunkt auf die Entwicklung und den Betrieb legt. Um sowohl in Bezug auf Bedürfnisantizipation als auch Modellinkongruenz wenig Risiko einzugehen, wurden regelmäßige Rückkopplungen mit den Benutzern durchgeführt. Spezielle Kommunikationskanäle helfen dabei, Bedürfnisse der Benutzer

wahrzunehmen. Die einfache Gestaltung des Systems soll Inkongruenzen so gering wie möglich halten. Die *ifu* betont die Außensicht von Studierenden und Lehrenden. Schwierigkeiten offenbarten sich bei der Bedürfnisantizipation und in der Kommunikation mit den Infrastruktur-Betreibern. Für die Kommunikation war zu wenig Raum eingeplant und es wurden keine expliziten Medien bereitgestellt. Damit stand kein systematischer Kanal zur Verfügung. In der Antizipation lag das Problem, dass Unsicherheiten nicht durch entsprechende Kommunikation ausgeglichen werden konnten.

#### 5.4.4 Konsequenzen für Software-Infrastrukturen

Die Existenz einer großen Anzahl von Akteuren im Entwicklungsprozess einer Software-Infrastruktur erfordert es, dass dieser Prozess die Möglichkeit explizit eröffnet, wiederholt neue Akteure in den Prozess einzugliedern. In Abhängigkeit von der Organisationszugehörigkeit der neu auftretenden Akteure ist ein unterschiedliches Vorgehen notwendig. In der Nutzung einer Infrastruktur sind Organisationen nicht die einzig relevante Größe, um Personen Gruppen zuzuordnen. In den Vordergrund treten Praxisgemeinschaften, die sich durch verwandte Bedürfnisse, ähnliches Handeln beziehungsweise Infrastrukturnutzung konstituieren. Erkennt man an, dass es grundsätzlich verschiedene Sichten auf die Infrastruktur gibt (wesentlich sind hier Innensicht und Außensicht), kann man wiederholt die unterschiedlichen Interessen, Vorgehensweisen und Bedürfnisse einordnen, die von den verschiedenen Akteuren artikuliert werden. Die verschiedenen Sichten auf die Infrastruktur stellen eine Schwierigkeit für die Eigenschaft *Alltagstauglichkeit* dar, da jede Sicht andere Anforderungen daran hat. Mit Blick auf *Standardisiertheit* können Interessenkonflikte entstehen, wenn unterschiedliche Praxisgemeinschaften nach verschiedenen Standards verlangen. Die Vielzahl der beteiligten Akteure und deren unterschiedliche Modelle der Infrastruktur erfordern einen kontinuierlichen Kommunikationsprozess, damit ein gegenseitiges Verständnis aufgebaut werden kann. Hierzu gibt die Akteurs-Sicht keinen weiteren Hinweis.

### 5.5 Software-Infrastruktur-Entwicklung als Prozess

Mit der Prozess-Sicht von Software-Infrastruktur werden die Grenzen der anderen Sichten aufgegriffen. Die Grenze der großen technischen Systeme

me liegt in ihrer Beschränkung, nur über geschehene Entwicklungen urteilen zu können und nur aus einer distanzierten Perspektive zu schauen, die Grenze der Artefakt-Sicht liegt in der Abgrenzung des Artefakts von seinem Entstehungsprozess und die Grenze der Akteurs-Sicht in der Vernachlässigung eines Prozesses.

Die Prozess-Sicht unterstellt nicht, ein fertiges Produkt zu erstellen, welches nach seiner Fertigstellung in unveränderter Form genutzt wird. Vielmehr geht es um die kontinuierliche Anpassung eines Netzes aus Technik, damit verbundenen Dienstleistungen, Nutzungskonventionen usw. an die sich ändernden Bedürfnisse der Benutzer.

Möglichkeiten der Prozess-Sicht liegen im situativen Reagieren auf neue Umstände, im Aushandeln von Konflikten, im Adaptieren angemessener Vorgehensweisen, in einer erhöhten Flexibilität, die das Sich-Zurechtfinden in einem organisch-komplexen Netz erlaubt. Ein Prozess erlaubt es, auf unterschiedlichen Granularitäten zu operieren. Somit können sowohl lokale als auch globale Belange adäquat adressiert werden. Konsequenzen dieser Perspektive sind das Fehlen eines definierten Endzustands genauso wie das Fehlen eines geordneten Beginns. Darüber hinaus liegt der Sicht eine kontinuierliche Veränderung zu Grunde. Damit wird der Sachverhalt akzeptiert, dass es nur lokale bzw. partielle Enden gibt, es wird ein langes Fortbestehen gewürdigt und die Notwendigkeit für fortwährende Änderungen anerkannt.

### **5.5.1 Ein vernetztes Prozessverständnis**

Der Prozess der (informationstechnischen) Infrastruktur-Entwicklung ist gleichzeitig ein informatischer und ein sozialer Prozess, da neben der reinen Verarbeitung von Informationen Arbeitszusammenhänge expliziert werden (Star und Bowker 1995, S. 41).

Im Mittelpunkt steht hier die Frage, welche Impulse softwaretechnische Entwicklungsprozesse aufnehmen können, um im Kontext von Infrastrukturen adäquat zu reagieren. Es existieren vier Vorschläge für Prozessmodelle als Ausgangspunkt: Hughes' Prozessmodell für große technische Systeme, Damsgaards und Scheepers' Modell für die Einführung von Intranets, Banslers Modell für die Einführung von Intranets in großen Organisationen und Kubiceks und Schmidts Modell für die Kommunikation über Infrastrukturen. Das von Hughes angebotene Prozessmodell für die Entwicklung großer technischer Systeme besteht aus den dargelegten Phasen (ver-



gleiche Kapitel 5.2.1–5.2.3) (Joerges 1988, S. 12). Während sich Hughes auf große technische Systeme bezieht, bieten Damsgaard und Scheepers ein Modell für die Einführung von Intranets an (Damsgaard und Scheepers 1999, Scheepers 1999). Dieses thematisiert allerdings nur die Einführung, nimmt dabei aber stärkeren Bezug auf die sozialen Belange. Bansler und andere gehen auf die Einführung von Intranets in großen Organisationen ein (Bansler et al. 2000).

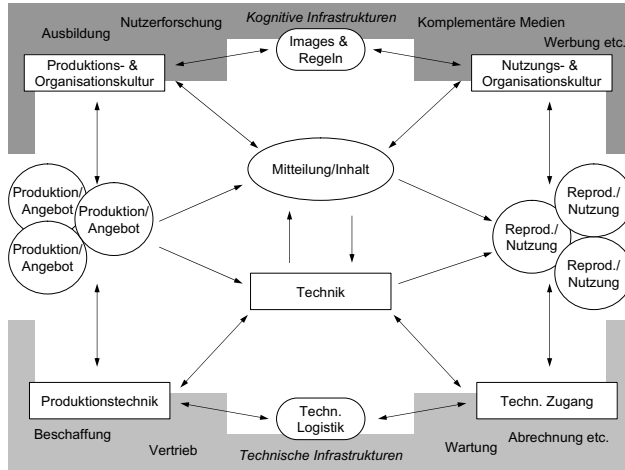
Im Zentrum insbesondere der letzteren Prozesse steht die Kommunikation. Kubicek und Schmid unterstreichen die Wichtigkeit von Kommunikation für den Entwicklungs- und Einführungsprozess und geben deshalb bei der Etablierung von Infrastrukturen am Beispiel elektronischer Medien zu bedenken:

„[...] der erhöhte technische und inhaltliche Abstimmungs- und Regulierungsbedarf sozialer Massenmedien ist nur zu bewältigen, wenn sich flankierende – gewissermaßen metakommunikative – Informations- und Interaktionsbeziehungen zwischen und innerhalb der Anbieter- und der Nutzergruppen entwickeln, wodurch die Selektion und Planung, die Kontrolle und Bewertung von Medienangeboten koordiniert werden kann. Dies wird z. B. durch komplementäre Medien ermöglicht, in denen auf Angebote und Verwendungsweisen hingewiesen, diese kommentiert und erklärt werden usw. (z. B. Zeitschriftenartikel, Werbung, Bedienungsanleitungen, Handbücher etc.). [...]

Darüber hinaus werden all die Aufgaben zunehmen, die im weitesten Sinn mit der technischen Distribution zusammenhängen. D.h. Vertriebsstrukturen und neue technische Infrastrukturen müssen geschaffen oder verändert werden, Kooperationen mit anderen Unternehmen und Institutionen müssen hergestellt und reguliert werden.“ (Kubicek und Schmid 1996, S. 34)

Das von Kubicek und Schmid (vergleiche Abbildung 1 auf der nächsten Seite) angebotene Modell unterscheidet zwischen technischer Infrastruktur (unten) und kognitiver Infrastruktur (oben). In diesem Spannungsfeld zwischen Technik und Inhalten werden angebotene Inhalte reproduziert und genutzt. Produktion und Nutzung der Inhalte sind in Kulturen eingebettet, die sich durch die jeweiligen Nutzergruppen definieren. Das gegebene Modell reflektiert die mit Innensicht und Außensicht bezeichneten Akteursperspektiven (vergleiche Kapitel 5.4.3).

Das Modell legt nahe, einen Software-Entwicklungsprozess in Bezug auf Kommunikation entsprechend reichhaltig auszustatten, um die unterschiedlichen Kommunikationsbedürfnisse zu befriedigen. Dabei stellt das



(Kubicek und Schmid 1996, S. 35)

Abbildung 1: Kommunikation über Infrastruktur

Modell zwei Dinge in den Mittelpunkt: die Technik und die Mitteilung (über Technik). Beide liegen kongruent zu den als Innen- und Außensicht bezeichneten Perspektiven. Die Technik ist stellvertretend für die Innensicht zu sehen. In der Art wie Akteure der Innensicht über Technik kommunizieren, ist die Mitteilung vergleichbar mit der Kommunikation, die Benutzer als Vertreter der Außensicht über die Infrastruktur pflegen. Das Modell unterstreicht damit, dass Akteure beider Sichten mit Informationen übereinander bedient werden müssen. Entsprechende Kommunikationskanäle und Möglichkeiten müssen zur Verfügung gestellt werden. Damit wird klar, welche kommunikativen Anstrengungen mit Infrastruktur-Entwicklung verbunden sind.

Der Aufbau einer Informations-Infrastruktur braucht Zeit (Hanseth 2000, S. 59). Eine Vielzahl von Dingen muss berücksichtigt werden, u. a. folgende Fragen: Welche Arbeitszusammenhänge und Ziele sollen unterstützt werden (Alltagssituationen)? Welche bereits existierende Basis gibt es ("installed base")? Welche organisatorischen Einheiten sind beteiligt (Akteure)? Gibt es parallel verlaufende Entwicklungen an derselben Infrastruktur (Entwicklungsvorhaben)? Welche existierenden Standards werden bereits verwendet, welche sind neu zu definieren? Ein Prozess zur Entwicklung ei-

ner Infrastruktur muss die Möglichkeit bieten, sich hierüber ein Verständnis zu bilden.

Parallel dazu bedeutet ein Entwicklungsprozess der Infrastruktur bei den Akteuren zusätzlich einen Lernprozess. Zusammenhänge der betreffenden Organisationen, Kenntnisse über verfügbare und potenziell relevante Standards, Wissen über existierende Infrastrukturen oder Infrastrukturbestandteile (s. u.) usw. müssen sich angeeignet werden.

Das Fehlen eines expliziten Prozesses im Projekt *ifu* und die nicht etablierte Kommunikation wurden besonders offenkundig in den Problemfällen. Da es kein etabliertes Medium für die Kommunikation zwischen Infrastruktur-Betreibern und Benutzerinnen gab, war die Kommunikation sporadisch, unvollständig und durch Zufall geprägt. Entsprechend bildete sich bei den Benutzerinnen ein inkonsistentes Bild. Eine ähnliche Problemsituation kann in der Kommunikation mit den anderen Infrastruktur-Akteuren gesehen werden. Da die *ifu*-Beteiligten weder untereinander noch mit den anderen Akteuren koordiniert kommunizierten, ergaben sich Widersprüche und Inkonsistenzen bei den Weiterentwicklungen. Ein ähnliches Problem bestand im *hamburg.de*-Kontext, da hier ebenfalls die Kommunikation mit den Benutzern der Infrastruktur wenig beachtet wurde (Bleek et al. 2002). Das Projekt *CommSy* zeichnet aus, dass hier die Kommunikation über Fehler das Fehlen einer begleitenden Kommunikations-Infrastruktur offen gelegt hat. Je stärker das *CommSy* für einen Personenkreis zur Infrastruktur wurde, desto wichtiger war es, darüber zu kommunizieren.

### 5.5.2 Interferenzen

Mit *Interferenz* möchte ich ein zentrales Phänomen bei der Entwicklung von Infrastrukturen bezeichnen. Grundlage für Interferenzen sind ständig ablaufende partielle Erneuerungen, Erweiterungen und Ersatz von Bestandteilen der Infrastruktur. Weber spricht in diesem Zusammenhang von einem Grundkonzept (Weber 1999, S. 19). Er stellt fest, dass es gerade eine Infrastruktur ausmacht, dass sie einer kontinuierlichen Weiterentwicklung während des Betriebs unterliegt. Durch die große Ausdehnung der Infrastruktur können die o. g. Entwicklungsvorhaben personell und organisationell unabhängig voneinander sein. Durch die Verbindung über die Infrastruktur selbst besteht aber die Möglichkeit zur Wechselwirkung.

Eine Interferenz entsteht, wenn störende Wechselwirkungen in einer Infrastruktur auftreten. Ich definiere sie wie folgt:

*Definition 4 (Interferenz)*

Eine Interferenz ist eine Störung bei der Infrastruktur-Entwicklung, die zwischen gleichzeitig stattfindenden Entwicklungsvorhaben an (Bestand-) Teilen derselben Infrastruktur, die organisationell unabhängig sind, aber infrastrukturell zusammenhängen, auftritt.

Mit „organisationell unabhängig“ werden jeweils die Teile der Infrastruktur bezeichnet, die von Akteuren betrieben oder genutzt werden, die nicht an einem gemeinsamen Entwicklungsvorhaben arbeiten.

Mit „infrastrukturell zusammenhängend“ werden die Teile der Infrastruktur bezeichnet, die über Standards miteinander gekoppelt sind.

Die organisationelle Unabhängigkeit der Entwicklungsvorhaben ist für Infrastrukturen eine kennzeichnende Eigenschaft. Durch die zeitliche und räumliche Ausdehnung der Infrastruktur und den größtenteils lokalen Bezug der Entwicklungsvorhaben findet eine Vielzahl von Entwicklungen unabhängig voneinander statt. Unabhängig bezeichnet dabei, dass kein Akteur in beide Entwicklungsvorhaben involviert ist.

Die Gesamtheit der Infrastruktur definiert sich gerade über das infrastrukturelle Zusammenhängen. Damit sind physisch verbundene Artefakte, aufeinander bezogene Dienstleistungen, gemeinsam verwendete Standards usw. gemeint. Entscheidend ist, dass es einen ununterbrochenen Pfad zwischen zwei Bestandteilen der Infrastruktur gibt.

Durch ein Entwicklungsvorhaben wird z. B. an einer Stelle der Infrastruktur ein Artefakt geändert, welches über viele Pfade mit anderen Teilen der Infrastruktur zusammenhängt. Entlang dieser Pfade pflanzen sich die geänderten Bedingungen fort. Auf diesem Weg werden sie gegebenenfalls abgeschwächt, sodass keine Interferenz entsteht. Bleiben sie erhalten oder verstärken sie sich, wirken sie auf zeitlich oder räumlich entfernte Bestandteile der Infrastruktur. Durch die Wechselwirkung ist eine Interferenz entstanden.

Es ist zu berücksichtigen, dass nicht nur technische Artefakte während der Änderung interferieren, sondern die beteiligten Organisationen sowohl in ihrem Einflussbereich als auch im Verhältnis zur Technik Anlass zu Interferenzen geben können:

“When looking more closely at infrastructures and institutions, they share the structural properties making them hard to change separately as well as when they are considered a single unit. This makes it conceivable that the change strategies share something as well.”

(Hanseth und Monteiro 1998a, S. 29)

Parallel laufende Entwicklungsprozesse an Bestandteilen einer Infrastruktur existieren. Dadurch wird die Wichtigkeit des Konzeptes *Interferenz* unterstrichen. Einmal in Betrieb genommene Infrastrukturen können nur während des Betriebs weiterentwickelt werden. Aufgrund der Größe einer Infrastruktur und der Vielzahl von Aktanten finden an einer Infrastruktur immer mehrere gleichzeitig ablaufende Entwicklungsvorhaben statt. Ein Teil der Entwicklungsvorhaben steht in Abhängigkeit zu anderen, ein Teil wird unabhängig betrieben.

Dies ist die entscheidend neue Situation bei der Entwicklung von Software-Infrastrukturen im Gegensatz zu klassischer Software, da hier nicht mehr alle Entwicklungsprozesse in einer Hand liegen. Jedem Initiator eines Entwicklungsvorhabens muss bewusst sein, dass andere Vorhaben existieren und dass keine Möglichkeit besteht, alle zu kennen oder zu koordinieren. Zwischen den Entwicklungsvorhaben einer Infrastruktur können Interferenzen auftreten. Die Interferenz erschwert es deshalb besonders, Infrastruktur-Entwicklungsprozesse angemessen zu betreiben.

Entscheidende Interferenzen traten im *ifu*-Projekt bei verschiedenen Vorfällen auf. Insbesondere zu nennen sind die Sicherheitsbemühungen des RRZ und die Auswirkungen der neuen Internet-Explorer-Version. Im Falle der Sicherheitsbemühungen gab es an derselben Infrastruktur, dem Netz der Universität Hamburg, zwei gleichzeitig ablaufende Entwicklungsvorhaben. Die Projekte *ifu* und *vifu* hatten es sich zum Ziel gesetzt, allen Studentinnen und Lehrenden eine E-Mail-Adresse zur Verfügung zu stellen. Dafür wurde der Server in Berlin betrieben. Das lokale Rechenzentrum, das RRZ, wollte die technischen Möglichkeiten für Spam-E-Mail reduzieren. Die Grundlage des einen Entwicklungsvorhabens, E-Mails über den Berliner Server zu verschicken, interferierte mit den Aktivitäten des RRZ, den E-Mail-Versand über entfernte Server zu unterbinden.

Im Projekt *hamburg.de* konnten Interferenzen im Zusammenspiel der verwendeten Software-Bestandteile beobachtet werden. Eine Vielzahl bereits implementierter Schablonen für Webseiten-Generierung stützte sich auf das Caching des Content-Management-Systems ab. Zeitlich parallel

und unabhängig davon wurden Applikationen für das Web entwickelt, die sich auf einen Applikations-Server abstützten. Als es darum ging, diese beiden Entwicklungsvorhaben nach außen hin zusammenzuführen, musste man sich für ein Session-Management entscheiden. Die Entscheidung für das Session-Management des Applikations-Servers interferierte mit dem Caching des Content-Management-Systems.

Das *CommSy*-Projekt offenbarte Interferenzen bei der Entwicklung des Systems für unterschiedliche Einsatzkontexte. Während die Anpassung des *CommSys* für den Studiengang Wirtschaftsinformatik die Verwaltung der Benutzerinformationen in der Datenbank implementierte, erforderte die Anpassung für die *ifu* die Verwaltung der Benutzerinformationen in einem Directory-Server. Die daraus resultierende interne Repräsentation von Gruppen war nicht kompatibel, sodass beim Zusammenführen beider Projekte alle Module betroffen waren.

## 5.6 Zusammenfassung

Die sechs Kerneigenschaften und das Interferenz-Phänomen habe ich in diesem Kapitel vor dem Hintergrund der Diskussion um große technische Systeme und Software-Infrastrukturen in vier Sichtweisen beleuchtet. In Bezug zu den großen technischen Systemen konnte gezeigt werden, dass das dort angebotene Begriffsrepertoire (z. B. Systemerbauer, Beharrungsvermögen) eine gute Ausdruckskraft für Infrastrukturen darstellt und hilft, dort beobachtete Phänomene zu benennen. Die beschriebenen Phasen für Entwicklungsprozesse großer technischer Systeme können in Teilen auf Infrastrukturen übertragen werden und liefern Aufschluss darüber, wie das Entstehen von Infrastrukturen stattfindet. So deckt sich z. B. der Umstand, dass nach einer für die Infrastruktur grundlegenden Erfindung nur noch kleinere Innovationen möglich sind, mit den Beobachtungen aus der Empirie.

Die Artefakt-Sicht hat die Wichtigkeit von Kulturalität für Infrastruktur auf drei Ebenen offenbart. Erst über ausgehandelte Standards können Infrastrukturen entstehen und die heutigen Größen annehmen. Verschiedene zu gleichen Zwecken errichtete Infrastrukturen müssen miteinander verbunden werden und das Gateway-Konzept bietet eine Möglichkeit, diese Kopplung durchzuführen. Durch den lang andauernden Entstehungsprozess von Infrastrukturen und deren Abstützung auf anderen Infrastrukturen

tragen sie eine Geschichtlichkeit mit sich, die bei der Weiterentwicklung berücksichtigt werden muss. Alle diese drei Facetten von Kultur treten bei Infrastrukturen zu Tage.

Im Gegensatz zu klassischen Software-Entwicklungsprojekten spielt eine viele größere Zahl unterschiedlicher Akteure bei Infrastrukturprojekten eine Rolle. Die Akteure kommen aus unterschiedlichen Organisationen und sind Mitglieder in verschiedenen Praxisgemeinschaften. Dadurch benutzen sie unterschiedliche Infrastrukturen und tragen gewohnte Nutzungsmöglichkeiten als Anforderungen an andere Infrastrukturen heran. Neben der Vielzahl von Akteuren stellt die Unterscheidung in Innensicht und Außensicht eine neue Qualität bei der Software-Entwicklung dar. Durch diese Unterscheidung werden widersprüchliche Ziele beim Umgang mit der Infrastruktur und Differenzen in der Kommunikation über Infrastruktur deutlich.

Der Aspekt Kommunikation ist Teil des Übergangs zu einer Prozess-Sicht, die einen kontinuierlichen Aushandlungsprozess über die Infrastruktur auf verschiedenen Ebenen anerkennt. Beginnend bei der Bewertung des Geschichtlichen ermöglicht die Kommunikation, dass die Betreiber die Nutzungsperspektive der Benutzer kennen lernen. Rückkopplung zeigt den Benutzern die stattfindenden Entwicklungsprozesse an der Infrastruktur. Die Prozess-Sicht erkennt gleichzeitig an, dass mehrere parallel ablaufende Prozesse die Entwicklung einer Infrastruktur vorantreiben. Diese Prozesse können störend aufeinander wirken. Der Begriff Interferenz wird dafür im Folgenden verwendet.

Die Eigenschaften von Software-Infrastruktur und Interferenzen bei der Software-Infrastruktur-Entwicklung gehen über das hinaus, was derzeit mit softwaretechnischen Methoden handhabbar ist. Im Sinne von Weber muss man erkennen, dass die Software-Entwicklung an diesem Punkt mit den bisherigen Entwicklungsmethoden einem Großteil der oben benannten Eigenschaften und Effekte nicht ausreichend Rechnung trägt:

„Die mit der Einführung von Infrastrukturen notwendig werdenden Techniken lassen sich wie folgt charakterisieren:

- Systeme müssen sich leicht weiterentwickeln lassen, um neuen Benutzer- oder Betreiberanforderungen ohne großen Aufwand gerecht werden zu können.
- Existierende Systeme müssen zu neuen Systemen zusammengefügt werden können, um zunächst isolierte Lösungen zu neuen integrierten Lösungen weiterzuentwickeln.

- Der Austausch von Komponenten existierender Systeme durch neue verbesserte Komponenten soll einfach möglich sein, so daß die Neuentwicklung eines Systems schrittweise erfolgen kann und die Migration von einem alten System zu seinem neuen Äquivalent kontrolliert stattfinden kann. [...]
- Durch die Veränderung einer Software-Infrastruktur in einigen Komponenten oder Teilsystemen sollen die verbleibenden Teile nicht beeinträchtigt werden, d. h. sie sollen in bezug auf die zu ändernden Komponenten änderungsunabhängig bleiben. [...]"

(Weber 1999, S. 22-23)

Weber spricht zu Recht das Problem der Weiterentwicklung an: Aufbauend auf bereits existierende Software-Arrangements oder Infrastrukturen ("installed base", s. S. 53, Geschichtlichkeit) muss deren Beharrungsvermögen überwunden werden und eine kontinuierliche Weiterentwicklung möglich sein.

Ein Ergebnis der Auseinandersetzung mit den Infrastruktur-Eigenschaften, der Interferenz und den herausgearbeiteten Sichten ist, dass Software-Infrastrukturen eine neue Qualität an Komplexität in die Softwaretechnik bringen. Infrastruktur vereint alle aufgeführten Sichten! Es ist nicht dienlich, nur eine der Sichten zu sehen. Infrastruktur muss aus unterschiedlichen Perspektiven gleichzeitig betrachtet werden, um alle Verhaltensweisen und Charakteristika zu erkennen. Software-Infrastrukturen erfordern den Umgang mit existierenden Installationen, ein Aspekt, der in wenigen softwaretechnischen Methoden berücksichtigt wird. Sie erfordern den Umgang mit einem komplexen Netz aus sozialen Akteuren, deren berechnete Teilnahme am Entwicklungsvorhaben sukzessive etabliert werden muss. Sie erfordern den Umgang mit einer Vielzahl von Standards, deren Relevanz durch die Vernetzung mit anderen Aktanten bestimmt wird. Und Infrastrukturen besitzen eine räumliche und zeitliche Ausdehnung, die aus Entwicklungssicht nur partielle Entwicklungsvorhaben zulässt. Damit muss man anerkennen, dass am selben Gegenstand gleichzeitig andere Entwicklungen stattfinden und diese mit der eigenen in Wechselwirkung stehen können.

Im kommenden Kapitel werde ich einen exemplarischen Software-Entwicklungsprozess ausbreiten, der die oben genannten Eigenschaften und Umstände berücksichtigt. Damit möchte ich zeigen, mit welchen unterstützenden Elementen Software-Infrastruktur-Entwicklung möglich ist.



### **III**

## **Infrastruktur als Orientierung für die Software-Entwicklung**



## Software-Infrastruktur-Entwicklung

O  
tell me  
all about  
Anna Livia!  
I want to hear  
all about Anna Livia.  
Well, you know Anna Livia?  
Yes, of course, we all know Anna Livia.  
Tell me all. Tell me now. You'll die when you hear.  
Anna was, Livia is, Plurabelle's to be.  
O Pluhurabelle  
— James Joyce, Finnegans Wake

Veränderung ist die Nicht-Eigenschaft. — Martin Walser

Als Ausgangspunkt für die weitere Arbeit möchte ich den Begriff „Software-Infrastruktur“ in einen Kontext einordnen, in dem „Infrastruktur“ als Fortsetzung von Interpretationspfaden für Software gesehen werden kann. Daran anschließend werde ich eine methodische Einbettung in die Softwaretechnik vornehmen, um Entwicklungsmethoden und Vorgehensweisen zur Infrastruktur-Entwicklung in Beziehung zu setzen. Dies führt zu einer Infrastruktur-Orientierung, die mit Herausforderungen für die Software-Entwicklung verbunden ist. Als zentrales Element werden Wechselwirkungen benannt, die eine neue Qualität von Komplexität erzeugen und mithilfe von zwei formulierten Prinzipien adressiert werden.

Ich zeige daraufhin, wie Software-Entwicklung als Infrastruktur-Entwicklung stattfinden kann. Dafür wird ein exemplarischer Entwicklungsprozess vor dem Hintergrund der Feststellungen aus den vorangegangenen Kapiteln prototypisch beschrieben. Insbesondere werden die verschiedenen Praxisgemeinschaften thematisiert und ihre Zuordnung zu Arenen, die notwendige Kommunikation sowie der Umgang mit Interferenzen herausgearbeitet.

## 6.1 Software-Infrastruktur – eine Einordnung

Dieser Abschnitt stellt eine Beziehung zu Kapitel 5 her, indem er die vorliegende Diskussion um verschiedene Sichten auf Software systematisiert. Software kann auf unterschiedliche Weise wahrgenommen werden. Um dies zu beleuchten, möchte ich zwei weit verbreitete Interpretationspfade verfolgen und Infrastrukturen in diese einordnen. Die jeweilige Wahrnehmung prägt den Umgang mit der Software, die darüber erschlossenen Möglichkeiten und die Art und Weise des Entwicklungsvorgehens (Lehman 1980, S. 1061-1076).

Ein möglicher Pfad beginnt damit, Software als *Programm* wahrzunehmen. Damit wird der formale Charakter betont, ihr Determinismus und die Kopier- und Transportiermöglichkeit. Programme sind kontextfrei und laufen auf Maschinen ab. Programme können bewiesen werden, enthalten Fehler und sind austauschbar. In Fortführung des Gedankens wird Software als *System* betrachtet. Diese Betrachtung vergrößert den Geltungsbereich. Gleichzeitig unterstützt die System-Metapher die Wahrnehmung einer zunehmenden Komplexität von Software, indem Ordnungsprinzipien und Hierarchien eingeführt werden. Ein Programm ist fertiggestellt, wenn es seiner Spezifikation genügt. Weiterführend wird Software als *Werkzeug* interpretiert. Die *Werkzeug*-Metapher bietet eine Einbettung von Software in den Anwendungskontext. Dadurch erlaubt die Metapher es, Software vor dem jeweiligen fachlichen Hintergrund zu bewerten. Aufgabenangemessenheit verdrängt oder ergänzt formale Korrektheit als Entscheidungskriterium für den Einsatz. Ein Software-Werkzeug ist fertig, wenn es im Verwendungszusammenhang des Anwenders die Bearbeitung seiner Aufgaben ausreichend unterstützt. Schließlich eröffnet Software als *Medium* zu sehen neue Nutzungsmodelle und gibt gleichzeitig einen Interpretationsrahmen, der medientheoretische Kriterien hinzuzieht. Zu diesem Pfad möchte ich Software als *Infrastruktur* ergänzen. Die neue Qualität besteht darin, dass Software erstens nicht mehr im Mittelpunkt steht, sondern in den Hintergrund tritt, sie wird „Mittel zum Zweck“ in Fortführung der Werkzeug- und der Medien-Metapher. Zweitens ist die Nutzung von Software in einem stärkeren Maße als beim Werkzeug durch die Handlung und Zusammenstellung des Anwenders, Alltagstauglichkeit, bestimmt. Und drittens stellt das Perspektivenpaar von Innensicht und Außensicht ein entscheidendes kommunikatives Problem auf den Ebenen der Nutzung und Wei-

terentwicklung heraus. Die Nutzung ergibt sich in unterschiedlichen sich überlagernden Praxisgemeinschaften. Software als Infrastruktur ist darüber hinaus niemals fertig, sondern wird situativ aus der Perspektive des Anwenders abgeschlossen.

Ein zweiter Interpretationspfad charakterisiert Software erst als *Produkt*. Damit verbunden ist die Eigenschaft, das Produkt zu standardisieren, wiederholt zu produzieren, es kontextfrei herzustellen und in verschiedenen Zusammenhängen einzusetzen. Im Zentrum steht die Herstellung. Die Qualität eines Produktes lässt sich unabhängig vom Verwendungskontext prüfen. Ein Produkt kann gehandelt werden und entzieht sich damit der Kontrolle des Herstellers. Die Eigenschaften des Produktes sind gleichsam seine Vor- und Nachteile. *Produkt-Familien* als Metapher versucht einen Teil der Begrenztheit zu überwinden. Kombinierbarkeit und Konfigurierbarkeit werden hinzugewonnen, wodurch ein Kontextbezug hergestellt werden kann. Die Beschränkungen der Produkt-Sicht brechen die Vorstellung von Software als *Prozess* auf, bei der die Entwicklung und der Einsatz als aufeinander abgestimmte Prozesse wahrgenommen werden. Der Annahme liegt zu Grunde, dass als Ziel der Entwicklungsprozess schrittweise in den Hintergrund tritt und durch den Nutzungsprozess abgelöst wird. Das unterstreicht die Individualität der Herstellung. Es geht schwerpunktmäßig darum, Unikate anzufertigen. In der Fortsetzung des Pfades bedeutet Software als *Infrastruktur* wahrzunehmen, anzuerkennen, dass Entwicklung und Nutzung als kontinuierlich ablaufende Prozesse miteinander verwoben bleiben. Es gibt nicht mehr einen Entwicklungsprozess, sondern die Situation gliedert sich in getrennte Entwicklungsprozesse auf, die über die gemeinsame Infrastruktur in Abhängigkeit zueinander stehen, aber von nahezu unabhängigen Akteuren betrieben werden. Diese Vielfältigkeit ist kennzeichnend für Infrastrukturen.

Im Sinne der beiden Pfade wende ich mich nun der Software-Infrastruktur als Ausprägung von Infrastrukturen zu. Eine *Software-Infrastruktur* ist der anwendungsbezogene Software-Bestandteil einer informationstechnischen Infrastruktur (vergleiche Definition 3 auf Seite 44). Eine Software-Infrastruktur entsteht durch die *perspektivische* und *aufgabenbezogene Zusammenstellung* dieser Bestandteile im Alltagshandeln eines Akteurs. Software-Infrastruktur kann mindestens aus den zwei Sichten *Innensicht* und *Außensicht* wahrgenommen werden. Die unterschiedliche Perspektive bestimmt z. B., ob sie als *Arbeitsgegenstand* oder als *Arbeitsmittel* verstanden

wird. Der Vorstellung einer Software-Infrastruktur liegt zu Grunde, dass an ihr *gleichzeitig verschiedene Nutzungs- und Entwicklungsprozesse* in unterschiedlichen Stadien ablaufen. Diese Prozesse können miteinander *interferieren*.

Die Metapher *Software-Infrastruktur* unterstreicht durch das Wort „Infrastruktur“, dass Software mit der Erwartung auf hohe Verlässlichkeit und freie Kombinierbarkeit verwendet wird. Für Software-Infrastruktur ist kennzeichnend, dass sie einem kontinuierlichen Weiterentwicklungsprozess unterliegt, der aus unterschiedlichen und voneinander unabhängigen Entwicklungsvorhaben besteht. Der Begriff Software-Infrastruktur illustriert, dass wenig Wissen über den Einsatzkontext im Vergleich zu Software-Werkzeugen bei den Entwicklern und Betreibern existiert. Auf der Seite der Anwender besteht geringes Wissen über die Realisierung und die damit verbundenen Zusammenhänge.

## 6.2 Methodische Einbettung in die Softwaretechnik

Das Wiedersehen:

Ein Mann, der Herrn K. lange nicht gesehen hatte, begrüßte ihn mit den Worten: „Sie haben sich gar nicht verändert.“

„Oh!“ sagte Herr K. und erlebte. — Bertolt Brecht, Herr Keuner

Software-Infrastrukturen sind schwer erschließbare Gebilde, da sie als Teil einer informationstechnischen Infrastruktur in einen Rahmen aus einer Vielzahl verschiedenartiger Bestandteile eingebettet sind. Ihre Ausdehnung ist nicht vollständig erkennbar, Zuständigkeiten sind schwierig zu ermitteln, Wechselwirkungen nicht absehbar und die Reichweite der eigenen Entscheidungen ist begrenzt. Umso wichtiger ist es, bei der eigenen Software-Entwicklung einen Prozess einzusetzen, der mit dieser unsicheren Situation umgehen kann. Bisherige Entwicklungen von Software-Infrastruktur verlaufen keineswegs systematisch. Vielfach kann beobachtet werden, dass Infrastrukturen in ihrer Entwicklung treiben (Ciborra 2000b) oder durch den Wechsel von Planung und Improvisation (Orlikowski 1996) erstellt werden. Es besteht deshalb ein Bedarf an einer methodischen Unterstützung der Entwicklung von Software-Infrastrukturen.

In den bekannten softwaretechnischen Vorgehensweisen werden die in Kapitel 4 und 5 beschriebenen Eigenschaften von Software-Infrastrukturen allerdings nicht ausreichend diskutiert. Ein Großteil bekannter softwaretechnischer Prozessmodelle geht von einer Neuentwicklung aus (McDermid 1991, Denert 1991, Balzert 1996, Sommerville 2001), und Interferenzen finden gar keine Beachtung. Darüber hinaus laufen im Kontext von Infrastrukturen unabhängige Entwicklungen ab, bei denen die jeweils Beteiligten die Gestalt der zugehörigen Entwicklungsprozesse frei wählen. Es kann deshalb nicht Ziel der Bemühung sein, für alle Entwicklungen einen einheitlichen Prozess vorzuschreiben oder ein spezielles Prozessportefeuille bereitzustellen.

Die existierenden Prozessmodelle eignen sich unterschiedlich gut für die oben charakterisierten Sichten auf Software. So passt z. B. das Wasserfallmodell gut für Programme, die einer Spezifikation zu genügen haben (Boehm 1976). Das Spiralmodell eignet sich für Software, die für einen klar umrissenen, fest definierten Anwendungsfall zu gestalten ist (Boehm 1988). Evolutionäre und partizipative Prozesse können für die Entwicklung von anwendungsorientierter Software eingesetzt werden (Floyd et al. 1989, Schuler und Namioka 1993). Der Unified Process versucht, einen Methodenrahmen für verschiedenartige anwendungsorientierte Prozesse bereitzustellen (Jacobson et al. 1999). Im Bereich vernetzter Software (Tanenbaum 1996) werden andere Ebenen mit den Entwicklungsmethoden adressiert als bei der o. g. Anwendungsentwicklung. Es stehen zusätzlich technische Fragen wie Robustheit, Skalierbarkeit, Replizierbarkeit, Effizienz, Speicherverbrauch, Logging etc. im Vordergrund. Darüber hinaus bekommen dort die hier nicht weiter behandelten Eigenschaften Ausfallsicherheit und Wartbarkeit eine schwer wiegende Bedeutung.

Schwerpunkt der Software-Entwicklung sind also jeweils sehr unterschiedliche Gestaltungsziele. Keines der Prozessmodelle ist explizit auf die Entwicklung von Infrastrukturen abgestimmt. Entscheidend ist es, einen angemessenen Prozess für den jeweiligen Ausschnitt zu finden. In der Konsequenz müssen die verschiedenen gewählten Prozesse miteinander verbunden werden.

Innerhalb einer Software-Infrastruktur wird demnach zu jedem einzelnen Software-Entwicklungsvorhaben ein potenziell anderer Software-Entwicklungsprozess gehören. In einigen Fällen wird die Auswahl ohne die besondere Berücksichtigung des Infrastruktur-Gedankens getroffen. Und

selbst diejenigen, die ihre Software-Entwicklung als Entwicklung einer Software-Infrastruktur begreifen, werden wahrscheinlich auf bekannte Prozesse zurückgreifen. Es scheint mir deshalb sinnvoll, ergänzende Prozesselemente bereitzustellen, die dem gewählten Prozess hinzugefügt werden können, um in der neuen Situation zu bestehen.

### **6.2.1 Evolutionäre Software-Entwicklung**

Da ich bei meiner Definition von Software-Infrastruktur auf anwendungsbezogene Software fokussiere, liefern Methoden für die Entwicklung von Anwendungs- oder Arbeitsplatzsoftware (Floyd und Züllighoven 1998, Züllighoven 1998) einen möglichen Ausgangspunkt. Hier scheinen sich evolutionäre Modelle besonders für die Entwicklung von eingebetteter Software zu bewähren, da sie explizit Mittel zur Verfügung stellen, um den Kontext zu berücksichtigen. Als Vertreter dieser Gattung möchte ich STEPS (Floyd et al. 1989) heranziehen. Ich werde nach einer knappen Skizzierung des Modells die Notwendigkeiten herausarbeiten und in den folgenden Abschnitten die Modifikationen und Ergänzungen an diesem Modell exemplarisch einführen und diskutieren.

Der Methodenrahmen STEPS bietet ein Vorgehensmodell, welches zur evolutionären und partizipativen Entwicklung beiträgt. Im Mittelpunkt des Prozesses steht gleichberechtigt neben der Software-Entwicklung das Ermitteln der Aufgaben der relevanten Benutzer, um für ihre Tätigkeiten angemessene Software zu konstruieren. Zu diesem Zweck wird zwischen Herstellung und Einsatz unterschieden. Initiiert wird ein Entwicklungsprojekt durch eine Projektetablierung, in der Entwickler und Benutzer ihre Vorstellungen und Möglichkeiten darlegen. Aus der Herstellung entsteht so eine Systemversion, die von den Benutzern eingesetzt wird. Erfahrungen mit diesem Einsatz fließen in eine Revisionsetablierung, die einen erneuten Evolutionszyklus einläutet. Eine entscheidende Grundannahme ist hierbei, dass die Nutzergruppe bekannt, eine Partizipation möglich und von allen Beteiligten gewünscht ist. Das Modell ist in dieser Arbeit der Ausgangspunkt für einen Software-Infrastruktur-Entwicklungsprozess.

### **6.2.2 *Design for Opportunities* und *Continuous Engineering***

Klassische Software-Entwicklungsprozesse und STEPS unterstellen, dass sich die zu unterstützenden Aufgaben zeitlich vor der Entwicklung der Soft-



ware festlegen lassen. Diese Idee wird zunehmend aufgegeben und durch das Schaffen eines Möglichkeitsraums ersetzt, in dem die Benutzer sich selbst orientieren. Dies lässt sich in eine softwaretechnische Richtung einordnen, die von den Autoren mit “Design for Opportunities” bezeichnet wird (Robinson 1993, Dittrich und Lindeberg 2003).

Bei der Software-Infrastrukturentwicklung wird angestrebt, durch die Infrastruktur einen Möglichkeitsraum zu eröffnen, der vom Anwender interpretiert, dessen Elemente kombiniert und im eigenen Sinne genutzt werden soll (Gebrauchstauglichkeit und Alltagstauglichkeit). Insofern liefern Ansätze aus diesem Bereich hilfreiche Impulse für die Infrastruktur-Entwicklung.

Eine weitere Anforderung an einen angemessenen Entwicklungsprozess für Software-Infrastrukturen ist es, mit bestehender Software umgehen zu können (Weber 1999, S. 31). Diese Forderung wird im Continuous Engineering vertreten. Sie ist aus meiner Sicht zu unterstützen, denn wie bereits oben herausgearbeitet, wird man mit einer “installed base” (s. S. 53) umgehen müssen.

### **6.2.3 Agile Software-Entwicklungsprozesse**

Unter den Stichworten “Adaptive Software Development” (Highsmith III 1999) und „Agile Prozesse“ bzw. “Agile Software Development” (Cockburn 2001, Highsmith 2002) werden Software-Entwicklungsprozesse diskutiert, bei denen die Prozessvorschriften während der Entwicklung revidiert und fallweise erneut entschieden werden. Merkmale agiler Prozesse sind Leichtgewichtigkeit und Flexibilität. Mit agilen Software-Entwicklungsprozessen unternimmt man den Versuch, die Anzahl der im Prozess zu erstellenden Artefakte weitestgehend zu reduzieren. Jedes Dokument muss gerechtfertigt werden und auf die Erstellung und Pflege von Nebenprodukten jenseits vom Programmcode wird verzichtet, wenn das Risiko dafür gering genug ist.

Leichtgewichtigkeit unterstreicht, dass Software-Entwicklung im Mittelpunkt steht und die Prozess-Steuerung nur als zusätzliche Aufgabe mit geringem Aufwand verstanden wird. Mit Flexibilität wird ausgedrückt, dass die gewählten Steuerungselemente wiederholt zur Diskussion stehen, um sie auf ihre Angemessenheit in der aktuellen Situation zu überprüfen. Darüber hinaus unterstreicht das Wort „agil“, dass es im Prozess leicht fällt,

Änderungen im Vorgehen und an dem Entwicklungsgegenstand kurzfristig durchzuführen und Geplantes zu revidieren, weil nur wenige Entwicklungsdokumente betroffen sind.

Agile Prozesse sind für Software-Infrastrukturen relevant. Sie verwenden die nötige „Kleinschrittigkeit“, um auf Wechselwirkungen flexibel zu reagieren, und beinhalten Prinzipien, die Änderungsaufwände möglichst klein halten. Software-Infrastruktur-Entwicklung kann agil konzipiert werden. Es ist lediglich notwendig, dass der gewählte Prozess Merkmale bzw. Ergänzungen besitzt, die auf die Anforderungen der Infrastruktur-Entwicklung eingehen. Diese werde ich im Folgenden eingehend beschreiben.

## 6.3 Software aus einer Infrastruktur-Perspektive

Mich erstaunen Leute,  
die das Universum begreifen wollen,  
wo es schwierig genug ist,  
in Chinatown zurechtzukommen. — Woody Allen

Der Charakter von Software-Infrastrukturen ist nicht wohlgeformt, nicht stark strukturiert und nicht geplant. Vielmehr wechseln sich Inseln strukturierter Elemente mit ungeordneten Bereichen und Verbindungen ab. Selbst geplante Bestandteile weisen keine hundertprozentig klare Trennung auf. Software-Infrastrukturen sind grundsätzlich über einen längeren Zeitraum gewachsen (Geschichtlichkeit) und daher eher vergleichbar mit sich überlagernden Sedimentschichten. In gleichem Maße sind auch keine klaren Grenzen zur Außenwelt zu ziehen. Je nach Sichtweise unterscheiden sich diese signifikant (Hanseth und Monteiro 1998a, S. 24).

### 6.3.1 Komplexität von Software-Infrastruktur

Im *Continuous Engineering* (Weber 1999) und in der Debatte zu *Information Infrastructure* (White House 2000, FBI 2000, OECD 1996, Bange mann 1997) wird die Einbettung der Software-Infrastruktur in Praxisgemeinschaften nur wenig berücksichtigt. Clement und Parsons unterstreichen, dass die ursprüngliche Annahme bei der Gestaltung von Software ein abgeschlossener Einsatzkontext war:

“Desktop Computers were envisioned as individual tools, standard packaged software as complete solutions for specific tasks.”

(Clement und Parsons 1990, S. 86)

Sie nennen drei Probleme, die beim Einsatz von vernetzten Systemen zusätzlich auftreten: *The Conceptual Model Problem*, *The Tacit Memory Problem* und *The Systems Design Problem*.<sup>1</sup> Diese verdeutlichen die Ebenen, auf denen Benutzer Verständnisproblemen ausgesetzt sind. Im Kontext von Software-Infrastrukturen habe ich diese mit Erlernbarkeit, Kommunizierbarkeit und Selbstbeschreibungsfähigkeit oder subsummierend mit Alltags-tauglichkeit bezeichnet. Meine Prozesserweiterungen sollen diese Probleme adressieren. Auf der Ebene der Außensicht müssen sich Benutzer von dem verwendeten und relevanten Ausschnitt der Infrastruktur ein Modell bilden können:

“The computer users must interface with and understand multiple layers of technology: *local hardware and operating systems interface, network interface, integration of package and data, basic application package interface, and advance package features*. A thorough knowledge of the semantic role of these layers of technology must be applied to design the mapping. This is not a sequential process, but rather requires reiteration and redesign as the interaction of each layer is considered.”

(Clement und Parsons 1990, S. 87, Hervorhebung im Original)

Die neue Qualität der Komplexität in der Software-Entwicklung kommt bei der Infrastruktur dadurch zustande, dass ein offenes Netz aus Akteuren unterschiedlichen Charakters mit verschiedenen Beziehungsarten den

---

1 “The Conceptual Model Problem: Networked systems cause a particularly persistent knowledge problem. The fundamental difficulty relates to the need for adequately complete mental models of the system[']s topology and the multiple layers of software interfacing. [...] The users lack a mental model that encompasses the roles of the different components, local and remote, involved in the transfer and the details of the interaction between them. [...]”

The Tacit Memory Problem: The second aspect of the office computer knowledge problem relates to the difficulty of making tacit the diverse package command syntax. Knowledge may shift to and from the tacit domain. Skills and knowledge may be tacit to the expert but very problematic to the novice. [...]”

The Systems Design Problem: The third aspect of computer related office knowledge involves the complex nature of office technology and, in particular, the extensive amount of systems designing that goes into utilizing the technology.”

(Clement und Parsons 1990, S. 86-87, Hervorhebung im Original)

Rahmen für ein Entwicklungsvorhaben bildet. In diesem Netz entstehen vielfältige Wechselwirkungen durch eigenes und fremdes Handeln.

Infrastruktur-Entwicklung kann mit der steigenden Komplexität in der Art umgehen, dass ausreichend Kommunikationskanäle bereitgestellt werden, um das Informationsbedürfnis aller Beteiligten zu befriedigen. Außerdem kann der gewählte Entwicklungsprozess helfen, die Komplexität kurzzeitig zu reduzieren, um Entscheidungen leichter treffen zu können.

### **6.3.2 Verschiedene Praxisgemeinschaften und ihre Infrastruktur-Wahrnehmung**

Indem ich die beiden Perspektiven *Innensicht* und *Außersicht* herausgearbeitet habe (vergleiche Kapitel 5.4.3), sind zwei Gruppen mit sehr unterschiedlichen Interessen hervorgetreten: die Betreiber und die Entwickler (Innensicht) sowie die Benutzer (Außersicht). Ein Entwicklungsprozess für Software-Infrastrukturen muss diese beiden Gruppen berücksichtigen. Die Interessen der Betreiber sind dabei anders und womöglich gegenläufig zu denen der Benutzer. Da die Anwendungsgebiete der Software und die Aufgabengebiete der Benutzer nicht bekannt sein können und die Benutzer wegen der Ausdehnung der Infrastruktur schlecht erreichbar sind, ist es besonders schwer, die Benutzer in den Entwicklungsprozess einzubeziehen. Darüber hinaus haben die Fallstudien gezeigt, dass sich die Benutzer in verschiedene Praxisgemeinschaften auffächern.

Die Beziehung der Anwender zu einer Infrastruktur definiert sich über ein Modell, das sie sich über die von ihnen wahrgenommenen Bestandteile der Infrastruktur bilden (vergleiche Abbildung 1 auf Seite 126). Sie sehen Dienstleistungen, Zuständigkeiten und eingegangene Verpflichtungen. Sie wollen von der Infrastruktur nicht alles wissen oder verstehen (Knowledge Interruptions, vergleiche Clement und Parsons 1990). Ihre Sicht auf die Infrastruktur lässt sich zusammenfassend als Außersicht bezeichnen.

Die Beziehung der Infrastruktur-Betreibenden ist von den Anwendern grundlegend verschieden. Sie haben ein Konstrukteursmodell der Infrastruktur. Dieses Modell steht vielfach im Widerspruch zum Benutzungsmodell der Anwender und manifestiert so einen Bruch der beiden Modelle. Die Betreiber haben eine Innensicht der Infrastruktur.

Die neue Qualität für Software-Entwicklung besteht darin, dass die Modelle der Innensicht-Mitglieder nicht mit denen der Außersicht in Überein-

stimmung gebracht werden können. Bedingt durch ihre unterschiedlichen Aufgaben und die damit verbundenen Tätigkeiten an der Infrastruktur werden sie nicht kongruent. Die Differenz von Außensicht und Innensicht wird im Kommunikationshandeln offenkundig, wenn Anwender der Infrastruktur mit infrastrukture gestaltenden Personen in Kontakt treten. Dies findet z. B. im Fehlerfalle statt. Orr greift denselben Sachverhalt, den Clement und Parsons als Knowledge Interruptions bezeichnen, auf und fasst dafür drei Konzepte zusammen:

“In my interpretation, the three concepts – Levi-Strauss’s *bricolage*, Schon’s reflective practice, and Suchman’s situated practice – all center on the interactive construction of an understanding and a basis for action in the context of the problematic situation.”

(Orr 1996, S. 11-12, Hervorhebung im Original)

Es geht bei der Infrastruktur-Entwicklung also um das Handeln in einer permanent „problematischen Situation“. Die wichtige Rolle, die im Fehlerfalle Personen an der Peripherie zwischen Außensicht und Innensicht spielen, wird von Orr ebenfalls hervorgehoben:

“Within the triangular relationship of service, technicians focus on maintaining the relations between their customer and the machines, and this is accomplished through technicians’ relationship with both. One of the technicians’ basic premises is that the machines will fail and need repairs; another is that the customers’ understanding of the machines will differ from those of the technicians.”

(Orr 1996, S. 79)

Die Entwicklung von Software-Infrastrukturen bedeutet zwangsläufig auch die Entwicklung von Software in unterschiedlichen Praxisgemeinschaften. Diese separaten Software-Entwicklungen werden in der Infrastruktur zusammengefasst. Infrastruktur zu untersuchen bedeutet deshalb in der Konsequenz, nebenläufige Entwicklungen zu betrachten. Zwischen diesen nebenläufigen Entwicklungen kann es zu Interferenzen kommen, d. h. Entwicklungen in einem Bereich können die verlässliche Basis eines anderen zerstören und damit die Entwicklung erschweren.

Dadurch, dass jeder einzelne Benutzer in mehreren Praxisgemeinschaften Mitglied sein kann, erleben die Benutzer jeweils unterschiedliche Nutzungskulturen von informationstechnischer Infrastruktur. Diese Kultur tragen sie in andere Praxisgemeinschaften hinein und an andere Infrastrukturen heran. Durch das Hineintragen von anderen Nutzungsmustern und

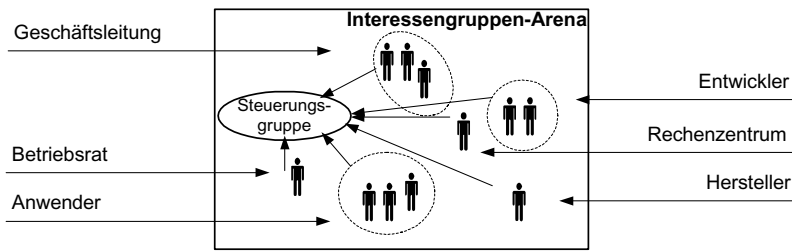
Anwendungsfällen beeinflussen Benutzer die Entwicklung einer Software-Infrastruktur.

Infrastruktur wird durch eine Gesamtheit persönlicher Nutzungsperspektiven definiert. Sie richtet sich nach einem Zweck und wird durch das verfolgte Ziel geprägt. Die Perspektiven können von mehreren Akteuren geteilt werden, wodurch sich Praxisgemeinschaften konstituieren.

Eine entscheidende Größe bei der Entwicklung von Infrastrukturen ist die Zeit. Eine Infrastruktur ist dauerhaft in Benutzung und somit kann sie nur während der Benutzung weiterentwickelt werden. Hierbei treten aus Sicht der Benutzer und Entwickler unterschiedliche Wahrnehmungen der Entwicklung der Infrastruktur auf. Mitglieder einer Praxisgemeinschaft erleben die Veränderungen an der Software-Infrastruktur unvermittelt, wenn sie nicht in den Änderungsprozess eingebunden sind oder über ihn informiert werden. Für sie entsteht der Eindruck, die Infrastruktur würde sich „selbsttätig“ verändern.

Die Gesamtheit der Software-Infrastruktur kann nicht mehr unter der Kontrolle und Wahrnehmung einzelner Personen oder Praxisgemeinschaften liegen, sondern ist auf verschiedene Praxisgemeinschaften verteilt und ragt damit z. B. in jeweils andere Organisationen hinein. Damit haben – beabsichtigt oder nicht – Änderungen an Teilen der Infrastruktur Auswirkungen in fremden Praxisgemeinschaften. Diese werden gar nicht bzw. unterschiedlich wahrgenommen. In diesem Zusammenhang konstatiert bereits Kling den unberechenbaren Charakter von Infrastruktur (Kling 1992, S. 17). Es wird deshalb eine Vorgehensweise benötigt, die mit dieser Eigenschaft umgehen kann und ihr entgegenwirkt.

Im Sinne von Kommunizierbarkeit ist die in Software-Entwicklungsprozessen (Floyd et al. 1989, Züllighoven 1998) angenommene gemeinsame Fachsprache der (zukünftigen) Anwender des Systems nicht vorzufinden. Dies liegt erstens daran, dass Anwender aus verschiedenen Praxisgemeinschaften stammen können, die unterschiedliche Kulturen besitzen. Zweitens wiederholt sich hier das Problem der Außen- und Innensicht in der Beziehung von Anwendern und Betreibern, die mit unterschiedlichen Modellen und Sprachen über „dieselbe“ Infrastruktur reden. Orr thematisiert ein ganz ähnliches Problem bei der Wartung von Kopiergeräten, bei dem sich Mechaniker darüber beklagen, dass Anwender der Kopierer nicht die richtigen Worte benutzen, wenn sie im Fehlerfalle mit ihnen über die Maschinen sprechen (Orr 1996).



(in Anlehnung an Bleek et al. 2002)

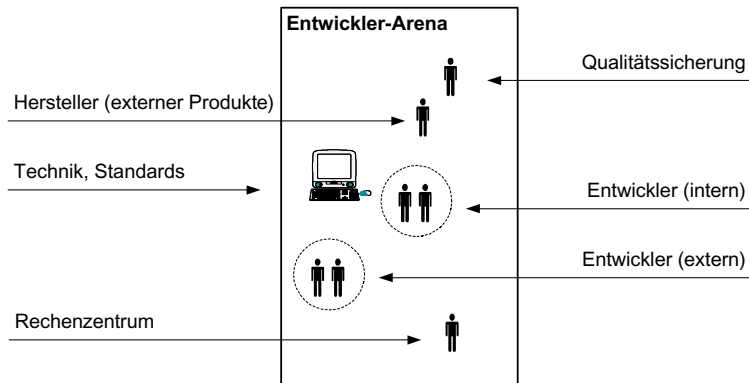
Abbildung 2: Software-Entwicklungsprozess: Beispielhafte Zusammensetzung der Interessengruppen-Arena

Keine der bekannten Methoden oder Vorgehensmodelle nimmt direkten Bezug auf die Entwicklung von Infrastrukturen oder Software, die Organisationsgrenzen übergreifend von mehreren beteiligten Praxisgemeinschaften gleichzeitig mit gegebenenfalls konkurrierenden Zielen in kontinuierlichem Fortschreiten (weiter-)entwickelt wird. Den Methoden fehlen Elemente, welche es erlauben, mit dieser heterogenen Situation umzugehen.

### 6.3.3 Arenen im Software-Entwicklungsprozess

In traditionellen Software-Entwicklungsprozessen sind bereits verschiedene Personengruppen involviert. Um die Komplexität der im Kontext von Software-Infrastrukturen relevanten Personengruppen darzustellen, möchte ich dies für einen evolutionären und partizipativen Software-Entwicklungsprozess leisten. In Verbindung zu einem Software-Entwicklungsprozess lassen sich drei Arenen beobachten, in denen beteiligte Personengruppen versuchen, ihre Meinungen zu artikulieren und ihren Einfluss auszuüben: die *Interessengruppen-Arena*, die *Entwickler-Arena* und die *Anwender-Arena* (Bleek et al. 2001, Bleek et al. 2002).

Arenen sind Orte des öffentlichen Handelns (Rolf 1998). Das Arenen-Konzept möchte ich neben Praxisgemeinschaften und Rollen stellen, da mit Praxisgemeinschaften nur soziale Zusammenhänge betrachtet werden und Rollen nur den Status in einer Gruppe widerspiegeln. Die Arena erlaubt es mir, im Prozess situationsabhängig Praxisgemeinschaften, Rollenträger, Artefakte und unveränderliche Bewegliche zusammenzuführen und in einem durch das Infrastruktur-Entwicklungsvorhaben definierten Zusammen-



(in Anlehnung an Bleek et al. 2002)

Abbildung 3: Software-Entwicklungsprozess: Beispielhafte Zusammensetzung der Entwickler-Arena

hang zu betrachten. Eine Arena stellt für die Personengruppen einen gemeinsamen Bezugspunkt, nämlich das Entwicklungsvorhaben im Rahmen der Infrastruktur, Synchronizität im gemeinsamen Handeln und einen Ort dar. Gleichzeitig definieren die Akteure Regeln des Zusammenwirkens, wie z. B. die Kommunikation mit Außenstehenden.

Es ist hilfreich, die Unterscheidung in Arenen zu treffen, da sie verdeutlicht, welche Beteiligten zur Innensicht (Entwickler), welche zur Außensicht (Anwender) gehören und welche Beteiligten beide oder andere Sichten (Interessenvertreter) einnehmen. Zusätzlich ermöglichen die Arenen später konzeptionell die Zugehörigkeit der Beteiligten zu verschiedenen Infrastrukturen und die Überlappung verschiedener Entwicklungsprozesse darzustellen.

Für eine erfolgreiche Entwicklung ist es essenziell, die relevanten Akteure in die Entwicklung angemessen einzubeziehen. In der Interessengruppen-Arena werden diese unterschiedlich intensiv repräsentiert. Für die klassische Anwendungsentwicklung sind es bereits Auftraggeber (z. B. Geschäftsleitung), Anwender (z. B. Angestellte), Entwickler (z. B. beauftragtes Unternehmen), Betreiber (z. B. Rechenzentrum, DV-Abteilung), Fremdprodukthersteller (z. B. Datenbank). Diese Diversität der beteiligten Gruppen ist schon für Software-Entwicklungsprojekte im Internet-Bereich fest-

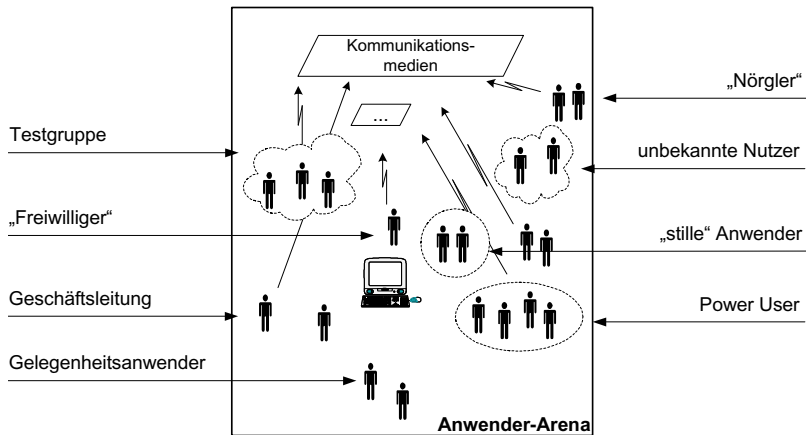


gestellt worden (Bleek et al. 2002). Die Abbildung 2 auf Seite 147 zeigt eine beispielhafte Interessengruppen-Arena. Die Arena umfasst alle relevanten Akteure klassischer Software-Entwicklungsprojekte. Weitere sind nach Bedarf einzubeziehen (s. u. Kapitel 6.5). Es ist aber keine Aussage darüber getroffen, welche Akteure aus den Interessengruppen zum Entscheidungsprozess im Software-Projekt beitragen (dürfen). Je nach Bedürfnis kann eine einzelne Gruppe detaillierter oder gröber betrachtet werden (z. B. die Gruppe der Anwender s. u.). Die Rolle der Beteiligten ist dabei die von Interessenvertretern. Sie artikulieren ihre eigenen bzw. die ihrer Gruppe innewohnenden Interessen. Hier stehen sich Betreiber und Entwickler auf der einen Seite und Anwender auf der anderen Seite gegenüber. Ein Aushandlungsprozess in der Interessengruppen-Arena wird helfen, ein angemessenes Gleichgewicht zwischen den Beteiligten herzustellen.

Bezüglich des Software-Entwicklungsprozesses ist die Entwickler-Arena von weiterer Wichtigkeit. In ihr sind Entwickler, Produktvertreter, Qualitätsbeauftragte, Fremdfirmen für Entwicklungsleistungen sowie Fremdfirmen für Dienstleistungen (z. B. Rechnerbetrieb) usw. anzutreffen. Die Abbildung 3 auf der vorherigen Seite zeigt die Entwickler-Arena. Die dargestellten Aktanten sind typische Vertreter heutiger Entwicklungsprojekte. Hier spielen Technologie (existierende Rechner mit z. B. Datenbankservern, Netzwerkkomponenten, Lesegeräten, Großrechnern) und Standards (z. B. Datenbankabfragesprache, Kommunikationsprotokoll) bereits in klassischen Projekten eine entscheidende Rolle.

Die Rolle der Beteiligten ist die Entwicklung und der Betrieb der Infrastruktur. Dies findet im Spannungsfeld zwischen Anforderungen aus der eigenen Arena und denen aus der Anwender-Arena statt. Die Entwickler sind vielfach an neuen Technologien interessiert, stehen im Wettstreit mit anderen Infrastrukturen und streben eine ästhetische Gestalt der Infrastruktur an. Im Gegensatz dazu ist es ein Interesse der Betreiber (z. B. des Rechenzentrums), möglichst wenig Veränderung an der Infrastruktur durchzuführen, diese in ihrem Funktionsumfang minimal und stabil zu halten und geringe Kosten aufzuwenden.

Die dritte Arena bilden die Anwender. Hier finden sich u. a. Benutzer (z. B. Angestellte), Schulungspersonal (intern, Fremdfirmen) und der Support (ebenfalls intern oder Fremdfirmen). Je stärker ein Entwicklungsvorhaben einen Infrastruktur-Charakter hat, umso schwieriger ist es, frühzeitig



(in Anlehnung an Bleek et al. 2002)

Abbildung 4: Software-Entwicklungsprozess: Beispielhafte Zusammensetzung der Anwender-Arena

die dieser Arena zuzuordnenden Personenkreise zu identifizieren. Die Abbildung 4 zeigt das Beispiel einer Anwender-Arena.

## 6.4 Die lokale und die globale Ebene bei der Infrastruktur-Entwicklung

Es hat sich gezeigt, dass Software als Infrastruktur eine neue Komplexität auf einer Vielzahl von Ebenen (Anwendungszusammenhänge, Nutzungskultur, Benutzer, Betreiber, Entwickler, Technologie usw.) darstellt. Mit dieser Komplexität muss man bei der Software-Entwicklung umgehen. Meiner Einschätzung nach kann es deshalb aber nicht bedeuten, einen neuen Entwicklungsprozess anzubieten, der mit dieser Komplexität zurechtkommt, da dieser wahrscheinlich noch komplexer wäre als es bestehende Prozesse bereits sind.

Hier zeigt sich eine interessante Parallele zur philosophischen Diskussion um die Postmoderne. Dort wird ebenfalls konstatiert, dass gegensätzliche Dinge widersprüchlich nebeneinander stehen, gesellschaftliche Zusammenhänge komplexer werden und es in der Konsequenz notwendig ist, Vielfalt anzuerkennen und den Gestaltungsanspruch im Ganzen auf-

zugeben (Welsch 1997). Diese Haltung möchte ich für die Entwicklung von Software-Infrastrukturen übernehmen. Komplexität von Software-Infrastrukturen ist anzuerkennen. Um mit ihr umzugehen, möchte ich wiederholt lokale Abgeschlossenheit herstellen, damit man bei der Software-Entwicklung handlungsfähig bleibt bzw. wird.

Um Software-Infrastruktur mit gewohnten Methoden zu entwickeln, ist es notwendig, ein teilweise abgeschlossenes Verständnis lokal zu erzeugen, um in dessen Rahmen mit bekannten Methoden vorzugehen. Ich möchte Prozesselemente vorstellen, die es ermöglichen, für einen begrenzten Zeitraum ein lokal abgeschlossenes Verständnis zu gewinnen und zwischen den Perspektiven „lokal“ und „global“ wiederholt zu wechseln. Bereits Star weist unter Verweis auf Hewitt darauf hin, dass Systeme der realen Welt, die sowohl aus Menschen als auch aus Maschinen bestehen und sich über Raum und Zeit erstrecken, zu komplex sind, um sie vollständig zu verstehen (Star 1997, Hewitt 1985). Diese Systeme werden asynchron verändert, Beziehungen der Bestandteile reichen nicht beliebig weit, die Relation der Bestandteile untereinander ist einem kontinuierlichen Aushandlungsprozess unterworfen und es gibt keine zentrale Kontrollinstanz. Menschen gehen nach Star mit dieser Situation in der Art um, dass sie sich lokal abgeschlossene Vorstellungen verschaffen, in denen die Komplexität ausreichend reduziert ist und die Zusammenhänge durchschaubar sind (Star 1997). Dadurch wird lokales Handeln für die Beteiligten ermöglicht. Ein Prozess zur Entwicklung einer Software-Infrastruktur sollte deshalb folgenden zwei Prinzipien gehorchen:

*Prinzip 1 (Lokale Abgeschlossenheit herstellen)*

Um bewährte Methoden aus der Softwaretechnik anwenden zu können, wird bei der Software-Infrastruktur-Entwicklung für ein Entwicklungsvorhaben eine lokale Abgeschlossenheit hergestellt.

*Prinzip 2 (Globalen Bezug sicherstellen)*

Um die lokale Entwicklung im Rahmen der Software-Infrastruktur einordnen zu können, wird regelmäßig ein Bezug zu den äußeren Bestandteilen der global erschlossenen Infrastruktur sichergestellt.

Entwicklung von Infrastruktur verfolgt kleine lokale Ziele, d. h. es geht nicht um die Entwicklung einer kompletten Infrastruktur, sondern es werden Weiterentwicklungen und Verbesserungen in lokalen Bereichen ins Au-

ge gefasst (“Is fixed in modular increments, not all at once or globally”, s. S. 53). Das Zusammenspiel der Veränderung mit dem bestehenden System wird erst nach deren Einführung erfahrbar. Die Auswirkungen zu erfahren und der regelmäßige Wechsel sind Teile der Entwicklung.

Das vielfältige Umfeld, in dem Software-Entwicklung stattfindet, mit seinen Zusammenhängen ist das eigentlich Neue. Software-Entwicklung findet in einem komplexeren organisatorischen Umfeld statt (mehrere Organisationen und Praxisgemeinschaften sind in den Arenen beteiligt), der zeitliche Horizont ist nicht nur in die Zukunft gerichtet (siehe unten; geschichtliche Dimension) und der vormals Orientierung stiftende Produktcharakter ist nicht mehr identifizierbar (Fertigstellung und festes Eigenschaftenbündel sind nur noch in zeitlicher Beschränkung definierbar und erkennbar).

Die Öffnung der Software-Entwicklung hin zur Infrastruktur erschließt die zeitliche Dimension von Software: Es gibt nun eine Vorgeschichte (“installed base”, s. S. 53) und man hinterlässt für weitere Entwicklungen wiederum eine Vorgeschichte. Beides gilt es systematisch im Software-Entwicklungsprozess zu berücksichtigen. Die zweite Hälfte dieses Kapitels wird eine Antwort aus einer softwaretechnischen Perspektive formulieren, indem die für diese Situation passende Prozesselemente und Prozesseigenschaften skizziert werden.

Die Entwicklung von Software-Infrastrukturen geschieht auf verschiedenen Ebenen. Wie bereits festgestellt, laufen gleichzeitig eine Vielzahl von Entwicklungsvorhaben an derselben Infrastruktur parallel ab. Das bedeutet, dass auf einer globalen Ebene diese Entwicklungen miteinander in Beziehung gesetzt und gegebenenfalls aufeinander abgestimmt werden sollten, um eine in sich konsistente Infrastruktur zu erhalten. Auf einer detaillierten Ebene sind für eine konkrete Entwicklung lokale Räume zu erschließen. Der globale Überblick hilft bei dem eigenen Software-Entwicklungsprozess die eigenen Aktivitäten einzuordnen. Der lokal erschlossene Raum gibt ausreichend Sicherheit für das Entwicklungsvorhaben. Im Folgenden werde ich beschreiben, wie lokale Abgeschlossenheit und globaler Bezug hergestellt werden können.

#### 6.4.1 Lokale Abgeschlossenheit herstellen

Die lokale Ebene der Infrastruktur-Entwicklung betrachtet im Detail das eigene Entwicklungsvorhaben. Sie nimmt mit Absicht eine autistische Perspektive ein. Lokale Abgeschlossenheit ist wichtig, um Komplexität zu reduzieren und klassische softwaretechnische Verfahren einsetzen zu können. Sie ermöglicht es, kleine Entwicklungsvorhaben durchzuführen und in kurzen Entwicklungszeiträumen zu planen. Dafür ist es notwendig, ein kleines und lokales Ziel zu definieren. Der Anlass für ein Ziel kann auf verschiedene Art gegeben sein. Es ist wichtig, die Wahl des Ziels zu explizieren und die an dieser Entscheidung beteiligte Gruppe von Personen ebenfalls explizit herauszuarbeiten.

Vor dem Hintergrund eines lokalen Ziels können relevante Infrastruktur-Bestandteile ermittelt werden. Dies bedeutet, wie von Kling beschrieben, sukzessive ausgehend von dem Anlass gebenden Infrastruktur-Bestandteil die Umgebung zu erschließen. Diese Umgebung sollte dabei klein gehalten werden, da im Zweifelsfall ansonsten die gesamte Infrastruktur damit zusammenhängt.

Die Bestimmung der lokalen Umgebung und des Ziels für das Entwicklungsvorhaben ermöglicht es, die relevanten Entwickler zu rekrutieren. Diese ergeben sich gerade daraus, welche Bestandteile der Infrastruktur im Rahmen des Entwicklungsvorhabens bearbeitet werden müssen. Sie können nun in den Entscheidungsprozess für das Entwicklungsziel erneut einbezogen werden. Gegebenenfalls sind die Zielvorstellung und die Umgebung entsprechend anzupassen. Dieser Prozess ist so lange zu wiederholen, bis eine gewisse Stabilität erreicht wurde.

Der Prozess kann als das Aufspannen einer *Hülle* verstanden werden. Denn das Konstrukt, bestehend aus Entwicklungsziel, relevanten Entwicklern und Personen und Infrastruktur-Bestandteilen ist über einen Keimkern induziert. Anders als bei formaler Konstruktion einer Hülle, bei der die Größe exakt bestimmt ist, ist sie bei der Infrastruktur-Entwicklung von den Entscheidungen der Beteiligten abhängig. Beim Herstellen lokaler Abgeschlossenheit ist zusätzlich eine lokale Begrenztheit sicherzustellen, d. h. möglichst wenig in das Entwicklungsvorhaben einzubeziehen. Hierfür kann keine Regel angegeben werden, sondern es kommt auf die Mitarbeit der Beteiligten an, in diesem Prozess Disziplin walten zu lassen. Praktiken der agi-

len Software-Entwicklung geben verschiedene Maße für kleine Entwicklungsziele an.

Ein mögliches Vorgehen zum Bilden der lokalen Hülle kann von den offensichtlichen Artefakten und Akteuren ausgehen, die an der Entwicklung beteiligt sind. Passend hierzu werden die damit verbundenen Aktanten rekursiv erschlossen. Der Erschließungsprozess pausiert bzw. bricht ab, wenn die an der Entwicklung beteiligten Akteure keine Relevanz in der Einbeziehung sehen. Werden komplexe Gruppen von Aktanten erschlossen, kann überlegt werden, diese durch einen Delegierten zu vertreten, um die Komplexität im eigenen Entwicklungsprozess gering zu halten.

Ein Beispiel aus den in dieser Arbeit ausgebreiteten Fallstudien ist die Einbindung von Druckern am Hauptcampus in die *ifu*-Infrastruktur. Die Räume am Hauptcampus besaßen Drucker, die über die Arbeitsstationen, die mit dem *ifu*-Server verbunden waren, angesprochen werden sollten. Der Server stand in Stellingen und es war notwendig, eine technische Lösung zu finden, um die Drucker-Warteschlangen mit diesem Server zu betreiben. Ausgehend von den konkreten Druckern wurde deshalb wiederholt der lokale Kontext erschlossen: Mit welchen Geräten ist der Drucker verknüpft? Welche Personen sind für den Betrieb zuständig? Welche technischen Barrieren behindern die Nutzung des Druckers von entfernten Standorten? Diese Fragen haben beim Aufbau eines Netzes geholfen, das zur Durchführung des Entwicklungsvorhabens „Anbindung des Druckers an die Arbeitsstationen“ essenziell war.

In der Abbildung 5 auf der nächsten Seite ist schematisch die Hülle eines Entwicklungsvorhabens dargestellt. Das Computer-Symbol im Zentrum steht für die geplante Software-Entwicklung. Um diese herum sind Aktanten erschlossen, die für dieses Vorhaben relevant sind. Je weiter sich der Kreis nach außen zieht, um so weniger relevant für das Entwicklungsvorhaben sind die Aktanten. Ist eine gewisse Komplexität erreicht, kann ein zusammenhängender Bereich des Netzes durch einen Delegierten ersetzt werden. Der Erschließungsprozess ist subjektiv und situativ und muss wiederholt überprüft werden.

Der Prozess des Einbeziehens von Akteuren wird im Folgenden mehrfach wiederholt werden. Immer wieder erscheinen relevante Akteure während eines Entwicklungsvorhabens. Ebenso können sich bereits eingebundene Akteure aus dem Entwicklungsvorhaben entfernen. Es ist wichtig, dass ein Entwicklungsprozess trotz seiner lokalen Begrenztheit dafür of-

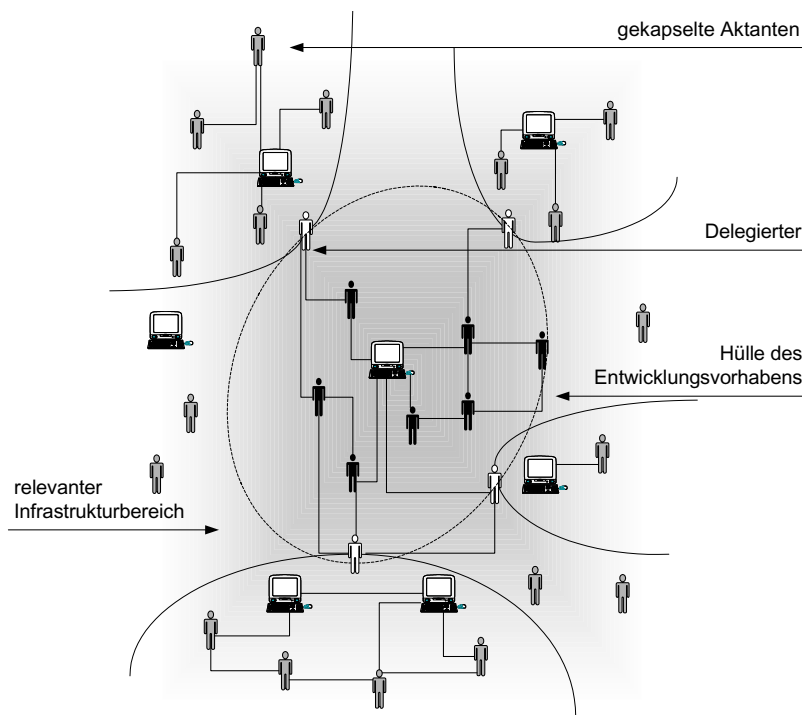


Abbildung 5: Die Hülle eines Infrastruktur-Entwicklungsvorhabens mit relevanten Grenzbereichen und Delegierten

fen ist, solche Akteure einzubeziehen bzw. bewusst Entscheidungen gegen deren Einbeziehung zu treffen.

Vor dem Hintergrund eines begrenzten und abgeschlossenen Entwicklungsvorhabens kann die Kommunikation mit den Benutzern der betroffenen Infrastruktur etabliert werden. Hier geht es darum, einen verlässlichen Kanal zu den Benutzern aufzubauen, über den sie ihre Erfahrungen und Bedürfnisse artikulieren können und der ihnen auf dem Rückweg Transparenz über die Entwicklung und Reaktion auf ihre Mitarbeit gibt. Näheres zur Kommunikation bei der Infrastruktur-Entwicklung findet sich in Kapitel 6.5.5.

Die Akteur-Netzwerk-Theorie kann beim Herausbilden lokaler Abgeschlossenheit in vielfacher Weise helfen. ANT kann verwendet werden, um

den Bereich der Infrastruktur außerhalb des Entwicklungsvorhabens möglichst grob zu kapseln. Andere Infrastruktur-Bestandteile werden so zu einzelnen Aktanten verdichtet (z. B. *der* Mailserver, *das* Rechenzentrum). Sie können in den Entwicklungsprozess mit reduzierter Komplexität einbezogen werden, indem Delegierte in den Entwicklungsprozess aufgenommen werden (z. B. der Mailserver-Administrator, die Rechenzentrumsleiterin). Es ist dabei wichtig, diese einzubeziehen, um das eigene Netz am Rand mit dem Außen zu verbinden und so Kontakt zur restlichen Infrastruktur zu halten.

Gleichmaßen leitet einen ANT an, das eigene Entwicklungsvorhaben nach außen zu kapseln. Das komplizierte Netz des eigenen Entwicklungsvorhabens aus verschiedenen Aktanten kann auf einen Punkt verdichtet werden. Mithilfe von Delegierten können eigene Interessen andernorts vertreten und als konsistent und für ein Ziel stehend dargestellt werden.

Um die eigenen Interessen bei der Infrastruktur-Entwicklung nicht nur zu artikulieren, sondern diese auch durchzusetzen ist es notwendig, sie anderen zu Eigen zu machen oder die eigenen Interessen anderen anzupassen. Dieser politische Prozess ist unbestreitbarer Teil der Infrastrukturentwicklung. Er ist schwer zu beschreiben und aus informatischer Sicht nicht weiter anleitbar. Teile des notwendigen Handelns werden im Zusammenhang mit Interferenzen in Kapitel 6.6 beschrieben.

Eine Möglichkeit, Interessen zu formulieren, diese zu wahren und durchzusetzen ist der Einsatz von unveränderlichen Beweglichen. Sie können zum Beispiel in der Form von Austauschformaten vorkommen. Diese vorgeben oder auszuwählen und andere dazu zu bringen, sie zu nutzen, verschafft bei der Infrastruktur-Entwicklung wertvolle Flexibilität und schreibt gleichzeitig anderen die eigenen Ziele ein. Damit wird nicht zuletzt eine Antwort auf die Frage nach Entwicklungsfähigkeit gegeben.

### 6.4.2 Globalen Bezug sicherstellen

Die zweite Ebene der Infrastruktur-Entwicklung verlagert den Fokus auf die Gesamtheit, um Relationen und Querbezüge zu verdeutlichen. Spätestens nach dem Ablauf eines Entwicklungszyklus ist es notwendig, den Bezug zur ganzen Infrastruktur wiederherzustellen. Damit kann eine notwendige Orientierung gewonnen werden, die hilft, das eigene Entwicklungsvorhaben genauer einzuordnen.



Der globale Bezug kann z. B. hergestellt werden, indem Pfade der eigenen Infrastruktur-Entwicklung zu anderen Teilen der Infrastruktur weiterverfolgt werden. Hierbei steht das Ziel im Mittelpunkt, nicht andere Organisationen bzw. Aktanten kennen zu lernen, sondern andere laufende Entwicklungsvorhaben aufzudecken. Damit gelingt ein prospektiver Blick in die Zukunft, wie sich die Infrastruktur später entwickeln wird. Vor dem Hintergrund der eigenen Entwicklung kann dies helfen, ein Urteil über weiteres Vorgehen zu bilden. Damit wird das Prinzip 2 auf Seite 151 eingelöst.

Die Erfahrungen des globalen Bezugs sind wertvolle Grundlage für weitere Entwicklungsschritte am selben Vorhaben. Sie fließen in die folgende Revisionsetablierung ein und können z. B. eine andere Besetzung der relevanten Personen oder Technologieauswahl zur Folge haben.

Beim Wechsel zwischen lokal und global kann ANT helfen, indem es mit den Konzepten von Netz und Kapselung sowohl die Möglichkeit und den Prozess zur Vergrößerung als auch den gegenteiligen Prozess zur Verfügung stellt. Gekapselte Netze können je nach Bedarf wieder geöffnet werden und deren Details werden sichtbar. Der von Kling angebotene Prozess ermöglicht es, diese Details so zu verfolgen, dass relevante Infrastruktur-Bestandteile offenkundig werden.

ANT erlaubt es, hier gleichzeitig Technologie (Konzepte) und Technik (Geräte) als Akteure einzubeziehen. Mit ANT können ebenfalls die verschiedenen Mitglieder einer Arena auf unterschiedlich granularem Niveau betrachtet werden. Ein Übergang zu Infrastruktur findet dann statt, wenn *aus der Außensicht* Elemente einvernehmlich gekapselt und benannt werden, die Dienstleistungen zur Verfügung stellen. Darüber hinaus ist es wahrscheinlich, dass mehrere Organisationen beteiligt sind.

## 6.5 Ein Entwicklungszyklus für Software-Infrastruktur

Was alle angeht, können nur alle lösen. Jeder Versuch,  
für sich zu lösen, was alle angeht, muss scheitern. —  
Friedrich Dürrenmatt, Die Physiker

Ich möchte nun einen exemplarischen Entwicklungsprozess für Software-Infrastrukturen vorstellen. An ihm werde ich zeigen, wie Software-Infrastruktur-Entwicklung aussehen kann und welche Prozesselemente ich für

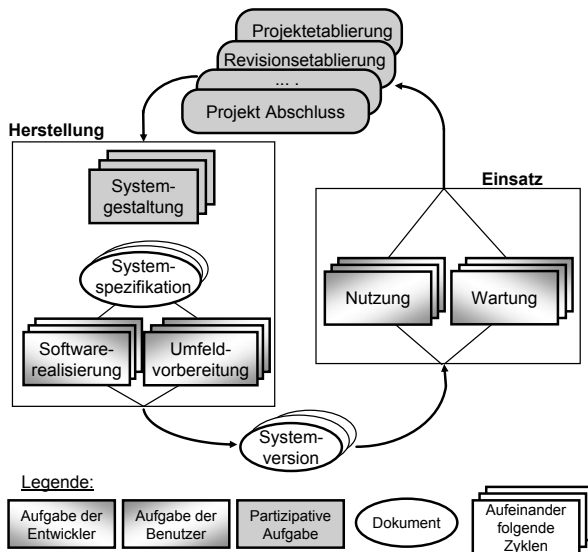
unbedingt nötig halte. Dabei stütze ich mich auf die zuvor ausgebreiteten Erfahrungen. Grundlage der Prozesselemente sind aus der Akteur-Netzwerk-Theorie entlehene Konzepte.

Die Etablierung eines Entwicklungsvorhabens steht am Anfang einer Infrastruktur-Entwicklung. Um die relevanten Personengruppen in das Entwicklungsvorhaben einzubeziehen, muss der lokale Kontext erschlossen werden, in dem die Entwicklung stattfinden soll. Dieser sollte wiederholt in Beziehung zum globalen Umfeld der Software-Infrastruktur gesetzt werden. Kommunikation ist essenziell für die Etablierung lokaler Kontexte. Welche Kommunikationskanäle eingesetzt werden können, steht im Mittelpunkt mehrerer Prozesselemente. Die an der Kommunikation beteiligten Personen lassen sich in Praxisgemeinschaften und Arenen einordnen, um deren Interessen am Entwicklungsprozess besser zu verstehen und einzubeziehen. Immer wieder während des Entwicklungsvorhabens auftretende Interferenzen mit anderen an der Infrastruktur stattfindenden Entwicklungen müssen offensiv behandelt werden. Dafür schlage ich im kommenden Kapitel ein Interferenz-Management vor.

Da Entwicklungsvorhaben an einer Infrastruktur unabhängig organisiert werden können, ist die Wahl der Entwicklungsmethode bzw. des Vorgehens nicht beeinflussbar und demnach als unterschiedlich anzusehen. Die angebotenen Prozesselemente müssen sich deshalb in verschiedene konkrete Software-Entwicklungsprozesse einpassen.

In dieser Arbeit ziehe ich einen evolutionären Entwicklungsprozess als Beispiel heran. Dieser eignet sich aus meiner Sicht besonders gut für Infrastruktur-Entwicklung. Das konkret verwendete STEPS-Modell stellt einen Methodenrahmen dar, der von bestimmten Grundannahmen ausgeht. Software-Entwicklung wird im Sinne situativer Vielfalt als Kommunikations- und Lernprozess verstanden. Dies unterstreicht die Eignung für Infrastruktur-Entwicklung. Die Herstellung und der Einsatz des späteren „Produktes Software“ sind miteinander verzahnt und verschränkt. Der Anspruch von STEPS ist daher nicht, Vorgaben für den Prozess zu machen, sondern der von STEPS bereitgestellte Rahmen ist vor dem Hintergrund „Infrastruktur“ zu interpretieren und auszufüllen.

Bisher gibt der Methodenrahmen STEPS Anleitung zur Verwendung von Methoden im Bereich aufgabenbezogene Anforderungsermittlung, zur Gestaltung der Mensch-Maschine-Interaktion und zum modularen Entwurf interaktiver Anwendungssysteme (Floyd et al. 1994). Ein Teil dieser Me-



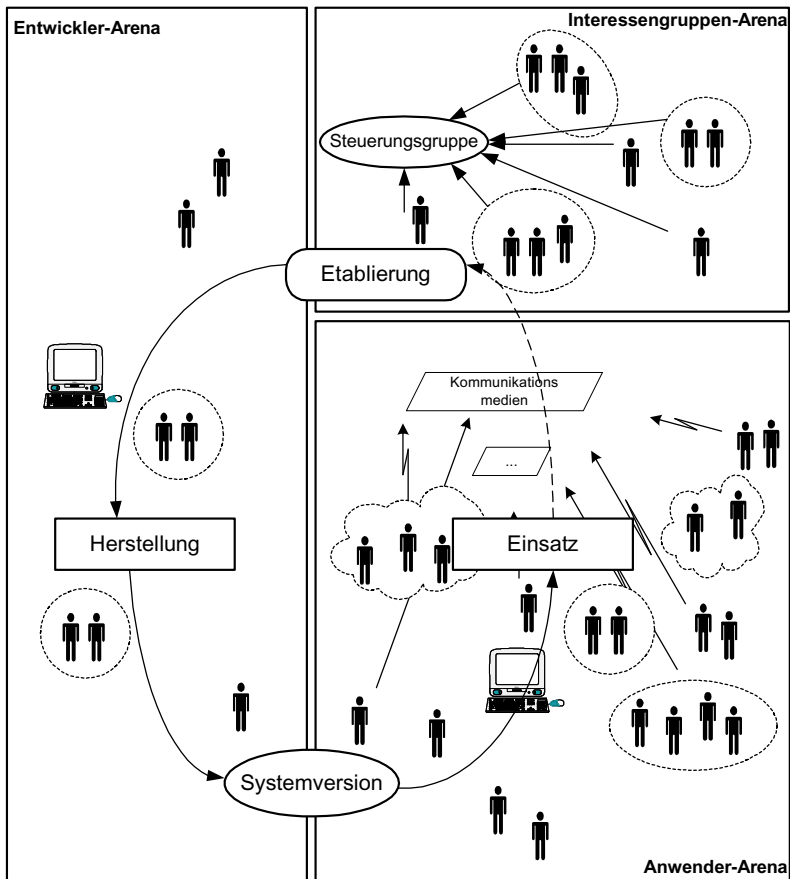
(Floyd et al. 1989)

Abbildung 6: Schematische Darstellung des STEPS-Prozesses

thoden kann im Kontext der Informations-Infrastrukturen eingesetzt werden, es bedarf allerdings der Adaption auf neue Rahmenbedingungen und einer konsequenten Ergänzung.

Das herangezogene STEPS-Modell unterscheidet vier zeitlich aufeinander folgende, ineinander verschränkte Abschnitte in einem Prozess: Etablierung, Herstellung, Versionierung und Einsatz. Die Abbildung 6 zeigt ein exemplarisches aus diesen Abschnitten bestehendes Modell.

STEPS unterscheidet zwischen Aufgaben der Entwickler, Aufgaben der Benutzer und partizipativen Aufgaben. Durch diese Aufgaben werden wiederholt gemeinsame Dokumente erstellt und in aufeinander folgenden Zyklen weiterentwickelt. Bereits im vorangegangenen Kapitel wurden ergänzend dazu drei Arenen benannt, in denen sich die beteiligten Personen aufhalten (vergleiche Abbildung 2 auf Seite 147, Abbildung 3 auf Seite 148 und Abbildung 4 auf Seite 150). Diese Unterscheidung erlaubt es, über den STEPS-Methodenrahmen hinausgehend Aussagen darüber zu treffen, welche Interessen die beteiligten Personen verfolgen könnten und welchen



(Bleek et al. 2002)

Abbildung 7: Der Software-Entwicklungsprozess eines Entwicklungsvorhabens

Einfluss sie auf das Projekt ausüben. Die Abbildung 7 stellt diese Art der Entwicklung dar, indem sie die in Kapitel 6.3 eingeführten Akteure und Arenen mit einem Entwicklungsprozess nach STEPS kombiniert. In der Abbildung ist die Schleife bestehend aus Etablierung, Herstellung, Systemversion und Einsatz über die drei Arenen gelegt. Im Folgenden wird auf die einzelnen Bestandteile im Kontext von Infrastrukturen eingegangen.

### 6.5.1 Etablierung des Entwicklungsvorhabens

Auslöser für ein Entwicklungsvorhaben an einer Infrastruktur ist vielfach das Bedürfnis, einen Missstand aufzuheben, eine gewohnte Arbeit in einem anderen Umfeld auf gleiche Weise zu erledigen oder eine Verbesserung in der bereits existierenden Infrastruktur herzustellen. Verschiedene Anlässe habe ich dazu in der Auswertung der Fallstudien benannt und charakterisiert.

Das hieraus resultierende Software-Entwicklungsvorhaben muss in geeigneter Weise etabliert werden, d. h. es müssen die relevanten Teile der Infrastruktur erschlossen, die tangierten Benutzer bzw. Praxisgemeinschaften identifiziert und die Akteure festgelegt werden, die eine Zielvorstellung entwickeln. Gegenüber klassischen Software-Entwicklungsvorhaben ist die Bestimmung dieser drei Bereiche bei der Infrastruktur-Entwicklung nicht abgeschlossen und kann nicht vollständig zu Beginn des Projektes durchgeführt werden. Ich schlage vor, diese als begleitenden Prozess zu konzipieren und werde das im Folgenden ausführen.

Ebenso ist es schwierig, zwischen Projektetablierung und Revisions-etablierung zu unterscheiden, da es hierbei auf die Perspektive ankommt. Für ein einzelnes Vorhaben mag es sich um die initiale Projektetablierung handeln, für die gesamte Infrastruktur kann eher von Revisionsetablierung gesprochen werden. Aus der Sicht eines Akteurs ist es ein neues Entwicklungsvorhaben, aus der Sicht eines anderen die Fortführung eines bereits existierenden Entwicklungsvorhabens. Die Unterscheidung zwischen diesen beiden Begriffen verschwimmt und verliert vor dem Hintergrund des Infrastruktur-Begriffs zunehmend an Bedeutung. Darüber hinaus muss man zwischen geplanter und ungeplanter Infrastruktur-Entwicklung unterscheiden. Veränderungen der Infrastruktur können zwar geplant sein und erlauben einen geordneten Etablierungsprozess, sie sind dies aber zumeist nicht. Infrastrukturen *gerinnen* häufig als solche und Entwicklungsvorhaben verweigern sich dadurch einer expliziten Etablierung. Ebenso werden Weiterentwicklungen vielfach zuerst von Einzelnen eingeführt und danach schleichend von anderen übernommen. Mit dieser Tatsache umzugehen kann bedeuten, beim Entdecken einer solchen Entwicklung daraus ein Vorhaben zu etablieren.

Ein Entwicklungsvorhaben im Infrastruktur-Kontext kann etabliert bzw. ausgelöst werden durch folgende Handlungen und Akteure:

- die geplante Entwicklung der Systemerbauer bzw. Betreiber;
- das geäußerte Bedürfnis einer Benutzergruppe;
- die Anpassung oder Initiative einer einzelnen Person;
- von einem Akteur beobachtete Entwicklungen an anderen Teilen der Infrastruktur.

Um den relevanten Teil der Infrastruktur zu erschließen, ist es parallel notwendig, die betroffenen Artefakte, die zielbestimmende Personengruppe, die relevanten Praxisgemeinschaften und die mit der relevanten Infrastruktur in Beziehung stehenden Organisationen zu ermitteln. Hierbei handelt es sich um verschränkte Prozesse. Dieser Erschließungsprozess ist notwendig, weil das eigene Entwicklungsvorhaben nicht alleine in der Infrastruktur stattfindet. Es gibt weitere Vorhaben und außen liegende Personengruppen, die es zu erkunden gilt.

In der Projektetablierung wird versucht, ein erstes skizzenhaftes Abbild des Entwicklungsvorhabens zu gewinnen, um es schrittweise im Prozess zu schärfen. Bezugspunkt für alle Entscheidungen ist dabei jeweils die im Entwicklungsvorhaben beschriebene Zielvorstellung. Wie dieser Prozess durchgeführt werden kann, ist Thema des Kapitels 6.4.1.

Damit wird das Prinzip 1 auf Seite 151 eingelöst, indem man sich zuerst für das Entwicklungsvorhaben einen lokal abgeschlossenen Raum herstellt. In diesem kann nach klassischem Verständnis Software-Entwicklung betrieben werden, wenn zwei Bedingungen erfüllt sind. Erstens muss die Kommunikation mit den oben genannten Benutzergruppen etabliert und zweitens kontinuierlicher Bezug zur gesamten Infrastruktur gehalten werden. Wie diesen beiden Bedingungen genügt werden kann, beschreibe ich weiter unten.

Die Etablierung kann mit und durch die Interessengruppen-Arena geschehen. Der angebotene Prozess (Abbildung 7 auf Seite 160) erlaubt es, eine Etablierung in der Steuerungsgruppe durchzuführen und durch die dort angesiedelten Vertreter weiterzutragen. In diesem Bereich tritt die Frage nach beteiligten Akteuren zuerst in den Vordergrund. Zur Etablierung gehört es, diese zu ermitteln und über ihre Beteiligung am Entwicklungsprozess geordnet zu entscheiden. Charakteristikum von Infrastrukturen ist es, dass immer wieder auf diese Entscheidung und evtl. neue Akteure zurückgekommen werden muss.

## 6.5.2 Herstellung in der Entwickler-Arena

Im Bereich der Systemgestaltung ist es zuerst wichtig, eine Systemspezifikation des zukünftigen Systems zu erstellen. Eine Schwierigkeit besteht darin, diese gemeinsam zwischen allen Beteiligten zu erarbeiten und ihnen zur Rückkopplung zur Verfügung zu stellen. Rückmeldung darüber zu erlangen gestaltet sich problematisch, denn nicht alle Belange der Infrastruktur betreffend sind von primärem Interesse z. B. der Anwender.

Für die Software-Realisierung helfen insbesondere Standards und das Streben nach Standardisiertheit sowie Entwicklungsfähigkeit weiter. Mit diesen teilweise gegenläufigen Eigenschaften wird das Spannungsfeld der Entwicklung aufgezeigt. Reproduzierbarkeit stellt einen zusätzlichen Faktor bei der Software-Realisierung dar.

Die Umfeldvorbereitung wird durch die Alltagstauglichkeit und beteiligte Akteure eingerahmt. Im Bereich der Alltagstauglichkeit geht es darum, die betroffenen Personen derart einzubeziehen, dass sie spätere Einsatzmöglichkeiten und gewünschte Eigenschaften der Infrastruktur antizipieren können. Dazu ist eine gemeinsame Sprache notwendig, über die man sich verständigen kann. Je kleiner der Personenkreis und je weniger Praxisgemeinschaften involviert sind, umso realistischer ist es, eine gemeinsame Sprache herauszubilden. Die beteiligten Akteure sind für die Umfeldvorbereitung zu ermitteln. Hier liegt ein Hauptproblem, denn man kann kein allgemein gültiges Verfahren angeben, wie diese ermittelt werden können. Es scheint vielmehr sinnvoll, den gesamten Entwicklungsprozess in kleine Vorhaben einzuteilen, um den Akteuren die Möglichkeit zu geben, sich schrittweise in den Entwicklungsprozess einzubringen.

Zur Gestaltung der Mensch-Maschine-Interaktion kann in Bezug auf Infrastrukturen nur wenig über das über Anwendungssysteme Hinausgehende gesagt werden. Elementar könnte im Sinne der Selbstbeschreibungsfähigkeit sein, dass die Anwendungssoftware mehr Auskunft über den Zustand der Infrastruktur gibt, indem sie keine Annahmen über z. B. Verbindungen trifft, sondern deren Zustand explizit visualisiert. Diese Forderung wird bereits für Anwendungs-Software gestellt, die zur Arbeit an technischen Automaten entwickelt wird (Bleek 1997).

Für den modularen Entwurf der Anwendungssysteme können z. B. Entwicklungsmethoden verwendet werden, die bereits für Anwendungssysteme und Client-Server-Systeme erprobt worden sind. Für deren Einsatz ist

es notwendig, die Kommunikationsmedien bereitzustellen, damit die voranstehend genannten Rückkopplungsmöglichkeiten realisiert werden können.

### 6.5.3 Systemversion

Die Systemversion, bestehend aus den schriftlichen Dokumenten der Software-Entwicklung, den lauffähigen Programmen, der dokumentierten Kommunikation mit den Benutzern und anderen Dokumenten, ist die zentrale Einheit im Software-Entwicklungsprozess. An ihr orientiert sich der Einsatz und die gesamte Kommunikation.

Im Kontext von Informations-Infrastrukturen ist es besonders schwierig, weiterhin von Systemversionen zu sprechen, weil man nicht von einem gemeinsamen Verständnis eines Gesamtsystems, einer „vollständigen“ Infrastruktur, ausgehen kann. Von einer Version der gesamten Infrastruktur zu sprechen ergibt keinen Sinn. Einerseits gibt es je nach Akteuren eine unterschiedliche Wahrnehmung, was zur Infrastruktur gehört, dementsprechend beeinflussen nicht alle Änderungen die existierende Systemversion. Andererseits laufen mehrere gleichzeitige Entwicklungsprozesse ab, sodass die Änderungen nicht disjunkt und sequentiell stattfinden. Für den Benutzer einer Infrastruktur stellen sich diese Unsicherheiten als störende Faktoren dar (vergleiche Interferenzen in Kapitel 6.6).

Für den Anwender und für den Entwickler ist eine Software-Infrastruktur nicht mehr vollständig erschließbar. Der Charakter als Gegenstand, Artefakt oder Produkt geht verloren. Dies liegt an zwei essenziellen Faktoren: Als Erstes ist zu nennen, dass die Komplexität der Infrastruktur weit über das hinausgeht, was innerhalb eines kurzen Zeitraums erschließbar ist. Dies bedeutet, dass alleine der Erschließungsprozess relevante Zeit in Anspruch nimmt, in der sich die Infrastruktur durch mögliche Entwicklungen verändert. Der zweite dagegen sprechende Faktor ist, dass sich nicht alle Teile der Infrastruktur in der Reichweite einer Organisation befinden, sodass andere Organisationen ihre Entwicklungen an der Infrastruktur durchführen und damit Inkonsistenzen in ein Verständnis von Version hineintragen.

Es ist deshalb zweckmäßig, die Dokumentation der Infrastruktur punktorientiert durchzuführen. Darunter verstehe ich, dass jeweils ein Artefakt dokumentiert und seine Beziehung zu anderen Aktanten offen gelegt wird. Benutzer und Entwickler können daran erkennen, ob das Verhalten eines



Artefakts noch dem beschriebenen entspricht und inwieweit sich sein Kontext geändert hat. Es entsteht eine Vernetzung von Beschreibungen, die den Hypertext-Dokumenten des Web gleicht. In diesem Sinne übernimmt man das Problem, auf Dinge zu verweisen, die gegebenenfalls nicht mehr in dieser Form existieren.

Die Kommunikation zwischen Mitgliedern der Außensicht und Mitgliedern der Innensicht über die Infrastruktur (z. B. Helpdesk) gehört zur Version einer Infrastruktur. Sie dokumentiert die Probleme, Einsatzkontexte und gegebenenfalls gemeinsam erarbeitete Lösungen im Umgang mit der Infrastruktur.

Der Versionsbegriff von Infrastrukturen ist ein partieller und dynamischer. Versioniert werden können nur einzelne Artefakte, Standards, Dienstleistungen usw. Das Zusammenspiel innerhalb der Infrastruktur ist ein komplexes Geflecht, in dem ein globaler und statischer Versionsbegriff wenig Sinn ergibt. Trotzdem ist es wichtig, die bewusst herbeigeführten Änderungen gegenüber anderen Akteuren zu kommunizieren. Lokale Versionsbegriffe mögen hier helfen, einen für eine Praxisgemeinschaft eigenen Referenzpunkt aufrechtzuerhalten. Entscheidend ist, dass gegenüber Akteuren erkennbar wird, dass eine Änderung stattgefunden hat. Nur unter dieser Voraussetzung können sie eine notwendige Überprüfung einleiten, ob ihre (bisher) konsolidierten Arbeitsmöglichkeiten noch bestehen und darüber Rückmeldung geben. Während also aus Entwicklerperspektive ein partieller Versionsbegriff für einzelne Bestandteile der Infrastruktur gilt, nehmen Benutzer die Versionen der Infrastruktur als fließend wahr und definieren Versionen für sich implizit durch stabile, konsolidierte Arbeitsmöglichkeiten der lokalen Umgebung.

### **6.5.4 Einsatz in der Anwender-Arena**

Im Bereich des Einsatzes und der Nutzung einer Software-Infrastruktur ergibt sich als erstes Problem, wie die neue Version den Anwendern zur Verfügung gestellt wird und diese darüber benachrichtigt werden. Die Infrastruktur wird als selbstverständlich von den Anwendern betrachtet. Ihnen ist also gar nicht bewusst oder für sie ist möglicherweise nicht von Interesse, dass ein Prozess der Änderung an der Infrastruktur vor der Fertigstellung steht. Möglicherweise können sie die Information auch nicht zuordnen, weil sie die Infrastruktur in der Art nicht verstehen oder nicht nutzen.

Das Einspielen einer neuen Version wird in jedem Fall zu Komplikationen in der Anwendung führen. Unverträglichkeiten werden offenkundig oder gewohnte Nutzungsmuster sind gegebenenfalls nicht mehr möglich. Der Rückfluss von Erfahrungen und Informationen über den Einsatz ist in der Abbildung 7 auf Seite 160 absichtlich gestrichelt dargestellt, da er nicht – wie in traditionellen Software-Entwicklungsprojekten – direkt stattfindet, sondern über verschiedene Medien zusammengetragen werden muss.

Der Begriff Wartung ist eng mit dem des Einsatzes verbunden. Leider ist der softwaretechnische Begriff Wartung problematisch. Wartung wird vielfach als Verbesserung von Fehlern während des Einsatzes verstanden. Eine Interpretation des Wartungsbegriffs als Pflege von Software durch Herausprogrammieren von Fehlern kann ich nicht unterstützen, da der Fehlerbegriff nur vor dem anwendungsfachlichen Hintergrund zu beantworten ist. Außerdem ist die Wartung aus Sicht des Anwenders nur schwer von der Weiterentwicklung der Infrastruktur zu unterscheiden: Ist das Einspielen einer neuen Version noch eine Wartung oder schon eine Weiterentwicklung? Im Bereich von Software-Infrastrukturen schlage ich deshalb zur Auflösung der Fehlinterpretation vor, den Begriff der Wartung in der Art zu definieren, dass darunter die Nutzung der Infrastruktur aus Sicht der Betreiber (Innensicht) verstanden wird. Für sie gilt in diesem Zusammenhang ein analoges Vorgehen im Entwicklungsprozess wie für Anwender.

## 6.5.5 Kommunikation im Entwicklungsprozess

I learned that system architects act as storytellers. —  
Alistair Cockburn

Kommunikation im Entwicklungsprozess betrifft die Kommunikation innerhalb der relevanten Interessengruppen, die Kommunikation innerhalb der Entwickler, die Kommunikation mit den den relevanten Infrastruktur-Bestandteil betreibenden Akteuren und mit den durch die Entwicklung betroffenen Benutzern und tangierten Organisationen. Diese verschiedenen Gruppen müssen durch ein Kommunikations-Management im Infrastruktur-Entwicklungsprozess eingebunden werden.

Kommunikation zwischen den Entwicklern und den Benutzern ist essenziell. Sie entscheidet über den Erfolg eines Entwicklungsvorhabens. Die Kommunikation ist besonders schwierig zu etablieren und aufrechtzuerhalten und deshalb sollte sie institutionalisiert und betreut werden. Hierfür bie-

ten sich verschiedene konventionelle Medien (Schwarze Bretter, Informationszettel, Meckerbriefkasten, Thekenbetrieb, Telefon usw.) genauso an wie ausgefeilte elektronische (Newsgroups, web-basierte Diskussionsforen, Bugtracking-Systeme, Chats usw.). Entscheidend ist erstens, dass verschiedene Kanäle gleichzeitig angeboten werden und wahlfrei zur Verfügung stehen. Zweitens müssen die Kanäle Rückkopplung herstellen, indem sie nach gewisser Bearbeitungszeit eine Antwort oder eine Lösung ankündigen. Keine Kommunikation darf dabei verloren gehen. Und drittens sollte ein Großteil der Kommunikation öffentlich erfolgen, um unbeteiligten Personen die Möglichkeit zu geben, die Medien, die Kommunikationskultur, das über den durchlaufenen Kommunikationsprozess gewonnene Wissen und das Potenzial kennen zu lernen.

ANT hilft, ein Netz derjenigen Aktanten aufzubauen, die mit dem Entwicklungsvorhaben verbunden sind. Greift man die sozialen Akteure und die hinter den Dingen stehenden Akteure heraus, ergeben sich die Linien, auf denen Kommunikation stattfinden sollte. Ob am anderen Ende jeweils Einzelne oder Gruppen anzutreffen sind, muss in der Situation entschieden werden. Danach richtet sich u. a. die Wahl des Mediums. Hält man die Kommunikation offen und signalisiert Bereitschaft zur Kommunikation, können gerade durch jedem zugängliche Medien neue relevante Akteure identifiziert und einbezogen werden.

## 6.6 Interferenz-Management

Unter dem Begriff Interferenz-Management möchte ich ein Bündel von Maßnahmen zusammenfassen, die es ermöglichen, Interferenzen frühzeitig zu erkennen und ihnen entgegenzuwirken. Interferenzen entstehen, wenn an organisational unabhängigen Teilen der Infrastruktur Entwicklungen von Akteuren durchgeführt werden, die infrastrukturell miteinander zusammenhängen und aufeinander wirken. Die zeitlich parallele Durchführung der Änderung führt zu gegenseitigen Störungen (vergleiche Abbildung 8 auf Seite 172).

Das angebotene Maßnahmenbündel besteht aus dem Herstellen und Aufrechterhalten eines kontinuierlichen Überblicks, der Pflege von Kommunikation mit Beteiligten, dem Identifizieren, Reduzieren, Behandeln, Ausgrenzen und Lindern von Interferenzen. Interferenzen zu vermeiden scheint nach den gegebenen Erfahrungen kein erreichbares Ziel zu sein.

Den kontinuierlichen Überblick über die Gesamt-Infrastruktur zu wahren (vergleiche Kapitel 6.4.2) ist Aufgabe eines allgemeinen Entwicklungsprozesses für Software-Infrastrukturen. Er dient somit ebenso der Identifizierung (s. u.) von Interferenzen, die das aktuelle Entwicklungsvorhaben stören könnten. Konflikte und gegenläufige Interessen sind Anhaltspunkte für Pole, zwischen denen Interferenzen entstehen können.

Wesentliche Grundlage für den kontinuierlichen Überblick ist, die Kommunikation mit den Benutzern zu pflegen (vergleiche Abschnitt „Kommunikation im Entwicklungsprozess“), um frühzeitig über Störungen, ungewünschtes Verhalten und andere Anomalien informiert zu sein. Die Benutzer helfen den Entwicklern, indem sie dies berichten, die Entwickler können den Benutzern helfen, indem sie bei von Interferenzen unabhängigen Problemen ebenso unterstützend wirken.

Interferenzen bewusst zu identifizieren bedeutet einerseits, Ursachen für Betriebsprobleme zusätzlich aus einer Interferenz-Perspektive zu betrachten und andererseits vor dem Einsatz von Infrastruktur-Bestandteilen diese auf ihr Verhalten in Hinblick auf Interferenzen zu überprüfen (Laborcharakter). Ist eine Interferenz beobachtet, geht es darum, die relevanten Akanten zu bestimmen. Hier hilft wieder Klings Vorgehen und die Akteur-Netzwerk-Theorie zum Aufbau eines der Situation entsprechenden Netzes. Ausgehend von dem Interferenz-stiftenden Aktanten wird ein lokales Netz aufgebaut, um die Zusammenhänge in der Infrastruktur aufzudecken. Mithilfe der Dokumentation der Aktanten, die die jeweiligen in direkter Beziehung stehenden Teile der Infrastruktur beschreiben, kann herausgefunden werden, welche Veränderung Ursache für die Interferenz war. Um die Veränderung ist wiederum ein Akteur-Netz aufzubauen, um die maßgeblichen Akteure zu identifizieren, die für die Veränderung verantwortlich sind.

Ist ein entsprechendes Netz erarbeitet, kann die entdeckte Interferenz gegebenenfalls durch Allianzen reduziert werden, wenn die Möglichkeit besteht, entscheidende Akteure politisch einzubeziehen. Haben diese aus Versehen rücksichtslos gehandelt? Haben sie ein begründetes Interesse, das eigene Entwicklungsvorhaben zu unterstützen? Kann man sie dazu bringen, das eigene Vorhaben zu unterstützen? Gibt es sachliche Gründe für deren oder das eigene Vorgehen, sodass auf dieser Ebene eine Lösung gefunden werden kann?

Im Rahmen der Allianz-Strategie kann es sinnvoll sein, Interferenzen dadurch zu behandeln, dass man sich auf gemeinsame Werte besinnt. Diese

Strategie ist im Kontext von Infrastrukturen besonders viel versprechend, da alle an der Entwicklung beteiligten Akteure einen „Sinn“ für Schlichkeit, Standardisierung und Abgestimmtheit besitzen. Die Nutzung gemeinsamer Standards eröffnet zusätzlich Chancen für weiter gehende Allianzen.

Gibt es keine Möglichkeit, die Interferenz auf gemeinsamem Wege zu beheben, besteht eine mögliche Strategie darin, den relevanten Infrastruktur-Bestandteil auszugrenzen, indem man sich selbst Autarkie verschafft. Dies geschieht z. B. durch den Aufbau einer eigenen Teil-Infrastruktur, die dasselbe leistet wie die interferierende. Das zur Ursachenfindung erarbeitete Akteur-Netz kann hierbei helfen, den Bereich zu identifizieren, der durch eigene (oder fremd eingekaufte) Teil-Infrastruktur ersetzt werden soll. Für jeden einzelnen Aktanten, der mit dem abzukoppelnden Bereich in Beziehung steht, muss ein Entwicklungsvorhaben definiert werden, das diesen an die neue Teil-Infrastruktur ankoppelt. Alle einzelnen Entwicklungsvorhaben können vielfach gleichzeitig durchgeführt werden, da sie typischerweise ähnlich sind.

Ist der Aufwand für Autarkie zu groß, können Interferenzen gelindert werden, indem man durch den Einsatz von Gateways versucht, den für die Annahmen wichtigen alten Zustand (künstlich) wiederherzustellen. Hierfür werden mithilfe des zur Bestimmung der Interferenz erstellten Akteur-Netzes die relevanten Verbindungen ermittelt. Aus der Summe der Funktionen, die die Verbindungen leisten, ergibt sich der Funktionsumfang des zu erstellenden Gateways. Der Austausch gegen das Gateway findet in derselben Art statt wie unter dem Fall der Autarkie beschrieben. Ein entscheidender Nachteil, den man mit Gateways in Kauf nehmen muss, ist die zusätzliche Transformation, die Leistung kostet und zu Verzögerungen führt. Darüber hinaus wird die Infrastruktur durch Gateways komplexer und komplizierter. Komplexer wird die Infrastruktur, weil mindestens ein weiterer Aktant in das Netz aufgenommen wird, komplizierter wird die Infrastruktur, weil alte und neue Standards gleichzeitig betrieben werden.

Interferenz-Management kann nicht helfen, Interferenzen bei der Infrastruktur-Entwicklung zu vermeiden. Bieten kann es einen geordneten Umgang mit Interferenzen, der in verschiedenen Situationen fundierte Prozesse für die Weiterentwicklung bietet.

## 6.7 Software-Entwicklung als Infrastruktur-Entwicklung

Wir lassen nie vom Suchen ab,  
und doch, am Ende allen unseren Suchens  
sind wir am Ausgangspunkt zurück  
und werden diesen Ort zum ersten Mal erfassen. —  
T. S. Eliot, Little Gidding

In diesem abschließenden Kapitel wurden Prozesselemente vorgestellt, die in Verbindung mit klassischen Software-Entwicklungsprozessen die Durchführung eines Entwicklungsvorhabens in einer Software-Infrastruktur erlauben. Am Beispiel eines evolutionären Entwicklungsprozesses wurden die Belange der Etablierung, der Einbeziehung relevanter Personenkreise, der Herstellung, der Bildung von Systemversionen, des Einsatzes, der Kommunikation mit den Anwendern und der Einbettung in die Gesamt-Infrastruktur vorgestellt und eingeordnet. Besondere Beachtung fanden dabei parallel ablaufende Entwicklungsprozesse und durch sie induzierte Interferenzen. Für Letztere wurde ein Interferenz-Management, bestehend aus einem Maßnahmenbündel, skizziert. Die Abbildung 8 auf Seite 172 fasst das Akteur-Netz, die Entwicklungsvorhaben mit ihren Arenen und Interferenzen in einem Bild zusammen. Die Abbildung zeigt eine Infrastruktur und zwei vergrößerte Ausschnitte als Beispiele für Entwicklungsvorhaben. Die Wolke wurde als Symbol für die Infrastruktur gewählt, um zu unterstreichen, wie undurchsichtig sich das Gebilde „Infrastruktur“ darstellt und mit welcher Dynamik es behaftet ist. Jedes Entwicklungsvorhaben ist durch einen technischen Ausschnitt aus der Infrastruktur gekennzeichnet (symbolisiert durch das Computer-Piktogramm) und mehrere Personen, die sich verschiedenen Arenen und Praxisgemeinschaften zuordnen (symbolisiert durch Personen-Piktogramme und mehr oder minder präzise Gruppierungen). Das Piktogramm „Funken“ zwischen den beiden Entwicklungsvorhaben in der Abbildung steht sowohl für neutralen Austausch, für Zusammenhang, für Zusammenwirken als auch für mögliche Interferenzen. Diese Phänomene treten zwischen unabhängigen Entwicklungsvorhaben auf und können nicht gänzlich vermieden werden.

Alle vorgeschlagenen Prozesselemente entstammen der Erfahrung aus den empirischen Projekten. Insofern bin ich zuversichtlich, dass sie in dem

gegebenen beispielhaften Entwicklungsprozess Erfolg versprechend eingesetzt werden können. Darüber hinaus glaube ich, dass die vorgestellten Prozesselemente in anderen Software-Entwicklungsprozessen eingesetzt werden können.

Leider sind die Aussagen zum Interferenz-Management nur auf einem allgemeinen Niveau. Die Erfahrungen mit Interferenzen haben gezeigt, dass hier unterschiedliche Rahmenbedingungen zu keiner spezielleren Aussage führen können. Der Umgang mit Interferenzen und das gesamte Durchführen eines Entwicklungsvorhabens im Kontext von Software-Infrastrukturen bedarf, wie allgemein Projektmanagement, einer ausreichenden Erfahrung.

Die herausgearbeiteten Kerneigenschaften und die in diesem Kapitel vorgestellten Prozesselemente erlauben zusammen, im Software-Entwicklungsprozess besser Entscheidungen zu treffen. Diese Entscheidungen bewegen sich in komplexen Akteur-Netzen nicht mehr im Ja-Nein-Bereich, sondern auf einer Prioritätenskala. Dafür liefert das Gewicht der Kerneigenschaften eine Maßgabe für die Priorisierung bei Entwurfskonflikten. Denn Konflikte lassen sich nicht mehr ausschließlich aus technikimmanenten Kriterien klären, sondern müssen im aufgebauten sozio-technischen Geflecht beantwortet werden.

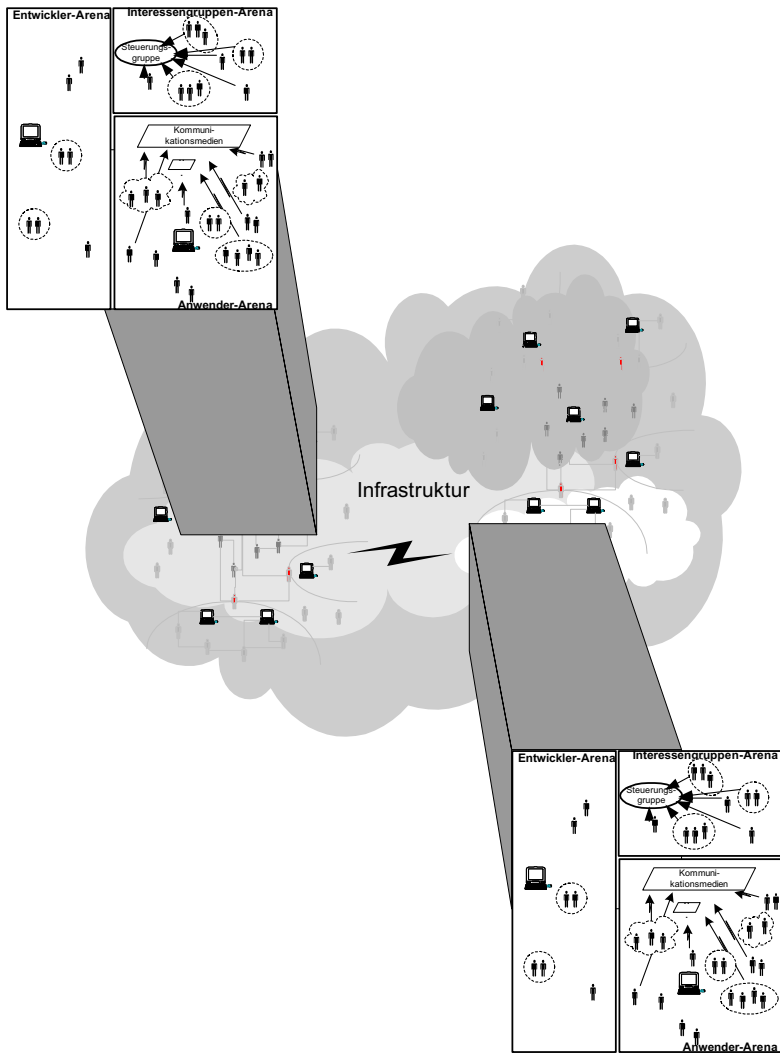


Abbildung 8: Exemplarische Entwicklungsvorhaben an einer Infrastruktur



## Super infrastruttura

There is never a forever thing — Pål Waktar

Auf der Grundlage eines qualitativen Forschungsansatzes habe ich in der vorliegenden Arbeit, ausgehend von Fallstudien, eine konstruktive Orientierung für die Entwicklung von Software-Infrastruktur aufgezeigt. Dabei wurde als Problem identifiziert, dass Methoden aus der Softwaretechnik nur selten die explizite Weiterentwicklung von Software und deren Vernetztheit betrachten. Infrastrukturen werden erst als solche erkannt, wenn sie bereits aufgebaut sind. Das heißt, der Umgang mit einer existierenden Basis, die sich als ein komplexes Netz aus ganz unterschiedlichen Elementen darstellt, ist unumgänglich. In diesem Netz aus Software, Technik, Standards, Akteuren, Beziehungen usw. gilt es den Entwicklungsgegenstand genauer zu bestimmen, Komplexität zu reduzieren, Wechselbeziehungen zu erkennen, Entwurfsentscheidungen zu fällen und Software-Entwicklung zu betreiben.

Um Software-Infrastruktur zu entwickeln, muss man mit einer existierenden Basis umgehen können. Entwurfsentscheidungen finden somit in einem komplexen, wechselwirkenden und nicht widerspruchsfreien Umfeld statt. Nebeneinander ablaufende Entwicklungsvorhaben unabhängiger Akteure an derselben Infrastruktur können mit dem eigenen Entwicklungsvorhaben in Wechselwirkung treten. Neben möglichen Wechselwirkungen sind die Rollen der beteiligten sozialen Akteure genauso wenig klar wie deren Einflussmöglichkeiten und ihre Relevanz für ein Entwicklungsvorhaben.

Große technische Systeme helfen, ein fundiertes Vorverständnis von Infrastrukturen aufzubauen und Charakteristika abzuleiten. Die Akteur-Netzwerk-Theorie auf die Software-Entwicklung im konstruktiven Bereich zu übertragen erlaubt es mir im Software-Entwicklungsprozess, die herausgearbeiteten Schwierigkeiten zu adressieren. Indem ANT bei der Identifikation des Entwicklungsvorhabens einerseits durch den Aufbau eines Akteurs-Netzes den relevanten lokalen Kontext erschließt und andererseits hilft, die

Komplexität im Entwicklungsprozess zu reduzieren, ist eine Grundvoraussetzung für Software-Infrastruktur-Entwicklung gegeben. Darüber hinaus erlauben die Konzepte von ANT einen differenzierten Umgang mit störenden Wechselwirkungen, die unter dem Begriff Interferenz von mir eingeführt wurden.

## 7.1 Zusammenfassung

Als Ausgangspunkt für die Auseinandersetzung mit Infrastrukturen habe ich in Kapitel 2 große technische Systeme eingeführt, die als Vorläufer von Infrastruktur aufschlussreiche Hinweise auf die Entstehung und den Betrieb geben. Ferner liefert die Forschung zu großen technischen Systemen Begriffe, mit denen Akteure, Phänomene und Phasen in der Entwicklung prägnant beschrieben werden können.

Die allgemeine Infrastruktur-Diskussion und insbesondere interdisziplinäre Forschung zu Infrastrukturen, die ich in Kapitel 3 ausgebreitet habe, bilden einen Ausgangspunkt für den hier vorgestellten Ansatz und offenbaren gleichzeitig das Forschungsdefizit, welches ich mit dieser Arbeit bearbeiten möchte. Derzeit existieren vorwiegend analytische Zugänge zu Infrastrukturen. Es mangelt an einer konstruktiven Orientierung für die Softwaretechnik, mit der Software-Infrastrukturen systematisch weiterentwickelt werden können.

Mittel für die fachliche Auswertung der Empirie ist eine Synthese aus sozio-technischen Ansätzen. Den dargestellten Ansätzen fehlen einzelne essenzielle Bestandteile. Ich benutze deshalb die jeweiligen Stärken des Web-Modells, der Infrastruktur-Kriterien und der Akteur-Netzwerk-Theorie, indem ich sie geeignet zusammenführe und als Grundlage für die Auswertung meiner empirischen Erfahrungen verwende. Insbesondere die Akteur-Netzwerk-Theorie hilft bei komplexen Software-Entwicklungsvorhaben, auf die relevanten Aktanten durch Kapselung zu fokussieren, wenn diese nach dem Vorgehen des Web-Modells und mit den Infrastruktur-Kriterien bewertet werden.

Für die Bearbeitung des empirischen Materials, welches ich in Kapitel 4 und im Anhang vorstelle und reflektiere, habe ich einen qualitativen Zugang gewählt, um mit gegenstandsbegründeter Theorie Anforderungen an einen Software-Entwicklungsprozess für Infrastrukturen herauszuarbeiten. Dieser Zugang zum Material ermöglicht mir trotz der eigenen Teilnahme

an den Projekten eine fundierte Theoriebildung. Drei Fallstudien habe ich dazu herangezogen.

Die am Ende des vierten Kapitels verdichteten Eigenschaften und Phänomene der Infrastruktur-Entwicklung ordne ich in Kapitel 5 vor dem Hintergrund des zuvor aufgebauten Technikverständnisses und grundlegender informatischer Ansätze ein. Das methodische und konzeptionelle Begriffsspektrum umfasst Begriffe, um sowohl Eigenschaften von Software-Infrastrukturen zu beschreiben und Phänomene bei der Infrastruktur-Entwicklung zu diskutieren als auch um Gruppen von Akteuren einzuordnen.

Eigenschaften von Software-Infrastruktur lassen sich auf der Grundlage der empirischen Arbeit mit Entwicklungsfähigkeit, Konfigurierbarkeit, Reproduzierbarkeit, Alltagsauglichkeit, Standardisiertheit und Aufbaubarkeit beschreiben. Entscheidendes Phänomen bei der Entwicklung von Infrastrukturen ist die Interferenz, die eine störende Wechselwirkung zwischen gleichzeitig, aber unabhängig stattfindenden Entwicklungsvorhaben an derselben Infrastruktur bezeichnet. Um große Akteurs-Gruppen besser zu handhaben und in einen Software-Entwicklungsprozess einzufügen, benutze ich u. a. die Konzepte Kapselung und Delegierter der Akteur-Netzwerk-Theorie. Damit den an der Entwicklung beteiligten Akteuren die verschiedenen Perspektiven deutlich werden und ein Software-Entwicklungsprozess diese ausreichend berücksichtigen kann, führe ich das Konzept der Innensicht und Außensicht ein und ordne es einem differenzierten Arenen-Begriff zu.

Ich stelle in dieser Arbeit bewusst keinen neuen Entwicklungsprozess für Software-Infrastruktur vor, sondern beschränke mich auf Prozesselemente, die in einem jeweils individuell gewählten Prozess vorzufinden sein sollten oder hinzugefügt werden. Diese Elemente präsentiere ich am Beispiel eines zyklischen und evolutionären Prozesses in Kapitel 6. Ausgehend von der Etablierung eines Entwicklungsvorhabens beschreibe ich, wie lokale Abgeschlossenheit im Infrastruktur-Geflecht hergestellt werden kann. Das ist Grundvoraussetzung, um klassische Entwicklungsmethoden einsetzen zu können. Der Prozess zum Bilden lokaler Abgeschlossenheit baut dafür eine Hülle um den im Zentrum der Entwicklung stehenden Aktanten. Dieser Prozess erlaubt es, unter Zuhilfenahme der in Arenen verstandenen Akteure relevante Infrastruktur-Bestandteile einzuschließen und das Entwicklungsvorhaben von nicht relevanten Bestandteilen abzugrenzen. Ich fordere darüber hinaus das regelmäßige Herstellen eines globalen Bezugs,

um die lokalen Entwicklungsaktivitäten mit der Infrastruktur-Umgebung in Beziehung zu sehen.

Im globalen Bezug des eigenen Entwicklungsvorhabens zur Infrastruktur werden Wechselwirkungen mit anderen Entwicklungsvorhaben in einen Kontext gesetzt. Der Bezug wird auf der Grundlage des Akteur-Netztes, der Delegierten und der vorhandenen Kommunikationskanäle hergestellt. Die Wechselwirkungen können sich als Störungen äußern, die ich als Interferenzen bezeichne. Ein geeignetes Interferenz-Management ist Bestandteil der Infrastruktur-Entwicklung, um mit Interferenzen koordiniert umzugehen.

## 7.2 Kritische Bewertung des Erreichten

*Quidquid id est, timeo Danaos et dona ferentes. —*  
Vergil, 2. Gesang der Äneis

Software-Infrastruktur erfordert den Umgang mit existierenden Installationen und damit Geschichtlichkeit, ein Aspekt, der in wenigen softwaretechnischen Methoden berücksichtigt wird. Sie erfordert den Umgang mit einem komplexen Netz aus sozialen Akteuren, deren berechtigte Teilnahme am Entwicklungsvorhaben sukzessive etabliert werden muss. Sie erfordert den Umgang mit einer Vielzahl von Standards, deren Relevanz durch die Vernetzung mit anderen Aktanten bestimmt wird. Und Infrastrukturen besitzen eine räumliche und zeitliche Ausdehnung, die aus Entwicklungssicht nur partielle Entwicklungsvorhaben zulässt.

In dieser Arbeit konnte ich zeigen, wie analytische Ansätze, die bisher im Kontext von Infrastrukturen verwendet wurden, als grundlegendes Hilfsmittel bei der Entwicklung von Software-Infrastrukturen eingesetzt werden können. Die Akteur-Netzwerk-Theorie als eine prominente Vertreterin dieser analytischen Ansätze stellt Konzepte zur Verfügung, die maßgeblich helfen, Komplexität bei Software-Entwicklung für Infrastrukturen zu meistern. Durch die Konzepte der großen technischen Systeme stehen aussagekräftige Begriffe zur Verfügung, die in den Akteur-Netzen eingesetzt werden können.

Der substantielle Schritt nach vorne liegt darin, die Verbindung von Akteur-Netzwerk-Theorie und großen technischen Systemen für die Softwaretechnik zu operationalisieren. In Kombination mit evolutionärer Software-

Entwicklung erlaubt ANT nicht nur einen analytischen Zugang, sondern kann zu Entwurfs- und Vorgehensentscheidungen anleiten.

Lokale Abgeschlossenheit hat sich als eine entscheidende Notwendigkeit für ein Software-Entwicklungsvorhaben erwiesen, um die Komplexität der gesamten Infrastruktur für einen begrenzten Zeitraum zu reduzieren. Dies ist ein weiterer Grund dafür, Software-Entwicklung in kleine Entwicklungsvorhaben aufzuteilen, wie es bereits im Continuous Engineering und bei der Agilen Software-Entwicklung propagiert wird. Mit der Synthese der analytischen Ansätze und der Verknüpfung zu evolutionären Software-Entwicklungsprozessen wird es möglich, kleine Entwicklungsvorhaben in komplexen Kontexten von Infrastrukturen zu finden und diese zur Gesamtheit in Beziehung zu setzen. Ein weiterer Vorteil ist, dass prinzipiell jeder seine bevorzugte und angemessene Entwicklungsmethode weiterhin einsetzen kann, wenn sie um die aufgezeigten Prozesselemente erweitert wird.

Eine entscheidende Barriere im Zusammenhang von Software als Infrastruktur konnte ich mit Innensicht und Außensicht genauer beschreiben und Möglichkeiten anbieten, sie zu überwinden: Kommunikation und ein neues Verständnis von Wartung. Die unterschiedlichen Perspektiven zu operationalisieren ermögliche ich mit meinem Arenen-Konzept. Es zeigt, zwischen welchen Akteuren Kommunikation notwendig ist.

Mit den empirisch fundierten Infrastruktur-Eigenschaften lassen sich Kriterien für die Entscheidungsfindung im jeweiligen Entwicklungsvorhaben bilden. Deren Wichtigkeit führt zu einer Prioritätenverschiebung in der Software-Entwicklung. Mit Software als Infrastruktur werden andere Dinge wichtig als bei klassischer Software, die als Applikation verstanden wird. Die von mir erarbeiteten Infrastruktur-Eigenschaften gewinnen bei Software-Infrastruktur-Entwicklung eine hohe Priorität, andere genannte Eigenschaften treten zurück.

In der Konsequenz plädiert die hier vorgestellte konstruktive Orientierung für Software-Infrastruktur dafür, zwangsläufig parallel stattfindende Software-Entwicklungsprozesse zu entkoppeln. Ein Anspruch, die gesamte Infrastruktur-Entwicklung zentral zu kontrollieren, wird aufgegeben. Lokale Entwicklungsvorhaben werden in regelmäßigen Abständen in einen globalen Bezug gesetzt. Verbindungen der eigenen Bestandteile werden expliziert und deren Weiterentwicklung in einem Prozess mit relevanten Akteuren abgestimmt.

Im Falle von Interferenzen ermöglicht das angebotene Interferenz-Ma-

nagement, die Ursprünge der Interferenz zu ermitteln. Spezielle Strategien helfen, verschiedenartig mit einer Interferenz umzugehen. Ich zeige, wie das Gateway-Konzept hierfür eingesetzt werden kann. Grundsätzlich ist der Entwicklungsprozess derart angelegt, dass Interferenzen durch gezielte Kommunikation an den entscheidenden Stellen vermieden werden. Damit wird eine Transparenz in der Gesamtentwicklung der Infrastruktur sowohl auf Seiten der Entwickler und betreibenden Organisationen als auch bei den Benutzern hergestellt.

Im Vergleich zum eingangs genannten Continuous Engineering setzt Software-Infrastruktur-Entwicklung abweichende Schwerpunkte. Es geht um eine ganzheitliche Entwicklungsperspektive, Software-Entwicklung in vernetzten Umgebungen und die Einbeziehung von Geschichtlichkeit. Aber es geht nicht um formale Methoden oder ein spezielles Komponentenmodell. Außerdem werden die Weiterentwicklung von Software und die Kombination von Software im Arbeitszusammenhang anders bewertet.

Die Grenzen der Aussagekraft meiner Erkenntnisse liegen in den ausgewählten Beispielen. An allen Fallstudien war ich maßgeblich beteiligt, d. h. meine Rolle war keine unabhängige. Die sorgfältige Anwendung der empirischen Methoden soll dies berücksichtigen, doch bleibt die Frage offen, inwieweit wegen meiner Teilnahme bestimmte Entscheidungen gefällt wurden, Entwicklungen stattgefunden haben und Probleme oder Erfolge aufgetreten sind.

Neben dem Problem der Verallgemeinerbarkeit von qualitativen Studien sind die Fallstudien frei ausgewählte Einzelfälle, die möglicherweise nur einen bestimmten Bereich von Software-Infrastrukturen abdecken. Es ist also anzunehmen, dass die herausgearbeiteten Kerneigenschaften nicht vollständig sind bzw. einen Schwerpunkt setzen, der bei anderen empirischen Untersuchungen unterschiedlich gelegt worden wäre.

Abschließend möchte ich als offenes Problem nennen, dass mein hier entwickelter Ansatz bisher nicht ausprobiert werden konnte. Es steht aus, die skizzierten Vorgehensweisen in konkreten Software-Entwicklungsprojekten einzusetzen und zu evaluieren. Damit würde eine zusätzliche Absicherung der ausgebreiteten Erkenntnisse gegeben und eine Überprüfung der vorgeschlagenen Prozesselemente durchgeführt.

### 7.3 Infrastrukturen für Infrastrukturen

Der Wunsch ist ein Wille, der sich selbst nicht ganz ernst nimmt. — Robert Musil

In der Beschäftigung mit Infrastrukturen ist deutlich geworden, dass sie subjektiv sind, d. h. sie definieren sich im Anwendungskontext des Benutzers. Diese situative Definition eröffnet eine neue Qualität: Infrastrukturen beziehen sich auf andere Infrastrukturen. Damit wird offensichtlich, dass Infrastrukturen wiederholt aufeinander geschichtet sind.

Infrastrukturen sind „Mittel zum Zweck“ und verschwinden – solange sie nicht ausfallen – aus dem Bewusstsein der Beteiligten. Deshalb werden sie leicht als Basis für andere Infrastrukturen genommen. In diesem Sinne ist die Elektrizitäts-Infrastruktur die Basis für die Telefon- und die Computer-Infrastruktur. Die Telefon-Infrastruktur ist Basis für das Internet usw. Diese Kette lässt sich wie das Auseinandernehmen einer russischen Puppe fortsetzen.

Oder aus der anderen Perspektive betrachtet: Die mir von einem Rechenzentrum zur Verfügung gestellte Infrastruktur ist für mich eine Infrastruktur, für die Mitarbeiter des Rechenzentrums ist sie Arbeitsgegenstand. Sie nutzen die Internet-, Telefon- und z. B. Backup-Infrastruktur eines anderen Rechenzentrums als Infrastruktur. Dieses andere Rechenzentrum betreibt die Backup-Infrastruktur als Arbeitsgegenstand usw.

Beide Blickrichtungen zeigen, dass Infrastrukturen aus der subjektiven Perspektive des Betrachters entstehen. Es liegt in seiner Entscheidung, welche Dinge er zur Infrastruktur deklariert. Damit werden Ansprüche gestellt und Eigenschaften attribuiert. Gibt es ein gemeinsames Verständnis innerhalb einer Praxismgemeinschaft, dann setzt sich eine Infrastruktur durch. Gegebenenfalls wird der Betrieb institutionalisiert und eine Verlässlichkeit hergestellt. Der Anspruch einiger Personen an Infrastruktur definiert die Arbeit für andere und lässt ein Beziehungsgeflecht entstehen, wie es mit den hier beschriebenen theoretischen Mitteln erkannt und weiterentwickelt werden kann.

## 7.4 Offene Fragen – Ein Blick nach vorne

Wenn ich mir was wünschen dürfte,  
käm' ich in Verlegenheit,  
was ich mir denn wünschen sollte,  
eine schlimme oder gute Zeit. — Friedrich Hollaender

Der Ausblick dieser Arbeit über Software-Infrastruktur gestaltet sich ganz im Sinne der Postmoderne multiperspektivisch und Vielfalt beachtend. Aus der Perspektive der Infrastruktur-Entwickler stellt sich z. B. als weiterführende Frage, wie Software-Infrastruktur getestet werden kann. Aus dem Blickwinkel der Infrastruktur-Benutzer stellt sich z. B. die Frage, wie Partizipation aufrechterhalten werden kann.

Das Testen von Software-Infrastruktur eröffnet neue Problemsituationen, denn weder die Reichweite der Entwicklung noch der Maßstab für das Testen stehen offensichtlich fest. Die Reichweite kann zwar über das hier beschriebene Akteur-Netz erfahrbar werden, beim Testen von Software steht allerdings die Anforderung im Vordergrund, ohne Störung der Benutzer einen Test zu absolvieren. Damit kann die existierende Infrastruktur nur eingeschränkt verwendet werden. Eine Kopie der Infrastruktur, das Testsystem, kann nur unter großem Aufwand erstellt werden. Insofern ist dieser Ansatz unrealistisch. Als Maßstab für das Testen stehen technische Gesichtspunkte nicht im Vordergrund. Die große Anzahl von beteiligten Akteuren erfordert es, deren Perspektiven im Sinne von Alltagstauglichkeit, Aufgabenangemessenheit usw. einzubeziehen. Es stellt sich ebenso die Frage nach der Machbarkeit.

Partizipative Software-Entwicklungsmethoden propagieren das systematische Einbeziehen der Benutzer in den Software-Entwicklungsprozess. Die Ausdehnung und die Anonymität in der Benutzung von Software-Infrastrukturen sind schlechte Voraussetzungen für die Einbeziehung von Benutzern. Alle Fallstudien haben gezeigt, dass es kontinuierlichen Aufwand erfordert, die Benutzer in den Entwicklungsprozess einzubeziehen. Die damit eingegangene Verpflichtung verlangt einen systematischen Umgang mit den aus der Kommunikation gewonnenen Informationen über die Verwendung der Infrastruktur. Dieser Rückkopplungsprozess verläuft bisher nicht optimal und ist weit entfernt von den Ansprüchen des Participatory Design.

Ein interessantes konzeptionelles Problem tritt bei der Dokumentation von Infrastrukturen auf. Die vorgeschlagene Vorgehensweise, jeweils einen



Aktanten und die in der Vernetzung entstehenden Wechselbeziehungen zu dokumentieren, baut eine den Hypertexten des World Wide Web ähnliche Dokumentenstruktur auf. Damit übertragen sich von dort bekannte Probleme auf die Dokumentation von Infrastrukturen (z. B. “Missing-Link-Problem”).

Insgesamt ist die Versionsbildung von Software-Infrastruktur kein gelöstes Problem. Da es unterschiedliche Versionsbegriffe nebeneinander geben wird und die Granularität, mit der Versionen vergeben werden, sich unterscheidet, entstehen neue Herausforderungen in der Kommunikation. Wie können Entwickler mit Benutzern über Versionen reden? Welche Bereiche der Infrastruktur werden mit welchen Versionen gekennzeichnet? Welche Kennzeichnungen ermöglichen den Benutzern zu erkennen, mit welchen Versionen von Software sie arbeiten?

Eine wichtige Tätigkeit bei der Infrastruktur-Entwicklung ist das Ermitteln der jeweils beteiligten Aktanten. Offensichtlich ist dieses Konzept für unterschiedliche Situationen gut einsetzbar: z. B. Etablierung, Kommunikation und Interferenz-Management. Es stellt sich die Frage, ob man diese Vorgehensweise verallgemeinern und in verschiedenen Zusammenhängen einsetzen kann.

Typisch für Infrastruktur ist der Umgang mit „vielen zusammengeffickten Dingen“. Um hier die philosophische Diskussion wieder aufzunehmen, darf ein erneuter Hinweis auf den Begriff „bricolage“ nicht fehlen. Damit wird auf eine Tätigkeit Bezug genommen, bei der aus einer heterogenen, aber begrenzten Sammlung Dinge so zusammengestellt werden, dass sie bei der aktuellen Aufgabe eine ausreichende Unterstützung liefern. In dieser Art entstehen Infrastrukturen aus der interaktiven Konstruktion vieler Benutzer. Sie werden durch wiederholtes Zusammenstellen geschichtet. Gerade die Begrenztheit des in diesem Moment Verfügbaren macht die Leistung „bricolage“ aus. In diesem Sinne geht es also gerade nicht um die systematische Zusammenstellung von geplanten Ressourcen, sondern um das Auskommen mit vielen Aufgaben in einer begrenzten Situation. Wenn gerade dieses Handeln typisch für Infrastrukturen sein sollte, bleibt die Frage stehen, ob ein systematisierter Ansatz überhaupt eine Lösung anbieten kann.



# **IV**

## **Anhang**



- A -

## **Empirieprojekt *Internationale Frauenuniversität***

Die *Internationale Frauenuniversität* (kurz *ifu*) war ein Projekt im Rahmen der Weltausstellung 2000 (Neusel 2000, ifu 2000a), dessen Ziel u. a. die Einführung eines neuen Studienkonzeptes für postgraduierte Studentinnen war. Die *ifu* fand in der für Deutschland vorlesungsfreien Zeit im Sommer 2000 (15. Juli bis 15. Oktober) parallel zur Weltausstellung statt. Es wurden sechs Studiengänge zur Auswahl angeboten (Körper, Stadt, Migration, Wasser, Arbeit und Information). Hauptstandort der Universität war Hannover, der Sitz der Weltausstellung. Der Projektbereich Information (Floyd et al. 2000, ifu 2000b) siedelte sich in Hamburg an und nutzte den Hauptcampus der Universität Hamburg mit deren freundlicher Unterstützung. Die Abbildung 9 auf Seite 188 gibt den groben zeitlichen Ablauf des Projektes wieder.

Prof. Cheri Kramer (internationale Dekanin) und Prof. Christiane Floyd (lokale Dekanin) leiteten den Projektbereich Information. Dr. Silvie Klein-Franke konzeptionierte den Lehrbetrieb, das Service-Center wurde von Tina Bach und die lokale Organisation von Dorit Heinsohn durchgeführt. Das Sekretariat besaß wechselndes Personal. Darüber hinaus wurde noch eine Vielzahl von studentischen Hilfskräften über die Projektlaufzeit engagiert. Während der Präsenzphase der Universität kamen die Lehrenden aus den unterschiedlichsten Ländern dazu.

Die Lehre des Projektbereichs Information gliederte sich in zwölf unabhängige Projekte. Studentinnen konnten sich einem Projekt zuordnen. Neben den Projekten gab es allgemeine Kurse und Vorlesungen. Als Projekte standen zur Verfügung: Citizenship, Community, Curiosity, Education, Gendering, Health Care, Identity, Kiosk, Knowledge, Media, Selfexpression und Virtuality. Die Tabelle 2 auf der nächsten Seite zeigt die Projekte mit den genauen Titeln und den verantwortlichen Lehrenden, um die internationale „Streuung“ zu zeigen.

Meine Rolle in dem Projektbereich Information umfasste drei Aufgabenbereiche, die eng miteinander verschränkt waren. Erstens ging es vor

Tabelle 2: Die Projekte des Projektbereichs Information

| <b>Projekt</b>                                    | <b>Director / Facilitator</b>                              |
|---|--|
| Visions of Citizenship                            | Esther Williams (Fiji) / Cirila P. Limpangog (Philippinen) |
| Community Development through Information         | Govind Kelkar (Indien) / Mona Dahms (Dänemark)             |
| Curiosity, Intuition and Information Technologies | Edla Faust-Ramos (Brasilien) / Yvonne Dittrich (Schweden)  |
| Future of Education                               | Cheris Kramerae (USA) / Zhang Wei (China)                  |
| Reconstructing Gender                             | Cheris Kramerae / Jutta Weber (Deutschland)                |
| Health Care Information                           | Tone Bratteteig (Norwegen) / Judith Gregory (Norwegen)     |
| Identities and Globalisation                      | Irma Pietrasanta (Mexiko) / Aleida Calleja (Mexiko)        |
| Information Kiosk                                 | Edla Faust-Ramos / Mona Dahms                              |
| Knowledge Architectures                           | Marsha Woodbury (USA) / Heike Winschiers (Namibia)         |
| Media Industries and Democracy                    | Irma Pietrasanta / Monika Pater (Deutschland)              |
| Cultural Modes, Selfexpression, and New Media     | Govind Kelkar / Dorcas Akande (Nigeria), Heike Winschiers  |
| Virtual Communities                               | Esther Williams / Yvonne Dittrich                          |

der Präsenzphase hauptsächlich um die Bereitstellung einer technischen Ausstattung und organisatorischen Unterstützung für die Büroarbeit eines Fachbereichs einer internationalen Universität. Zweitens war der Aufbau und Betrieb einer Internet-basierten Lernumgebung für die Studentinnen und Lehrenden gefordert, die während der Präsenzphase am Hauptcampus verfügbar sein sollte und auf die ein Zugriff nach dieser Zeit möglich ist. Diese Position wurde mit dem Titel *Technology Advisor* benannt, da der mit beiden Aufgaben verbundene dritte Bereich in der aktiven Beratung von Technikauswahl und -einsatz in den Projekten bestand.

Neben dem Aufbau einer Präsenzuniversität wurde von Anfang an Wert darauf gelegt, Virtualität und Nachhaltigkeit im Projekt zu berücksichtigen (Schelhowe 2000a). Dazu wurde das Forschungsprojekt *Virtuelle Frau-*

*enuniversität* (kurz *vifu*) etabliert (Schelhowe 2000b). Dessen Aufgabe war die Entwicklung und Bereitstellung einer Internet-basierten Kommunikationsplattform für alle Projektbereiche der Frauenuniversität. Damit verbunden war die Hoffnung, bereits vor Beginn der Universität eine reichhaltige Kommunikation zwischen den Teilnehmerinnen aufzubauen und während und nach der Präsenzphase aufrechtzuerhalten. Das Forschungsprojekt Virtuelle Frauenuniversität wurde federführend von der Humboldt-Universität Berlin unter Dr. Heidi Schelhowe geleitet.

Eine Besonderheit des gesamten Projektes und aller seiner Teilprojekte war, dass eine komplette internationale Universität für Postgraduierte nur für ein Semester aufgebaut wurde. Sowohl im administrativen Bereich als auch in der Lehre konnten nur die Erfahrungen aus anderen (nationalen) Kontexten übernommen werden – eine Vorerfahrung in einem gleichartigen Projekt existierte nicht. Konkrete Erfahrungen zu Abläufen, Bedürfnissen, Problemen und Situationen mussten erst gemacht werden. Es bestand für das gesamte Projekt allerdings eine hohe Aufmerksamkeit und Sensibilität bezüglich der Mehrsprachigkeit und Interkulturalität.

Die empirische Basis für dieses Projekt bildeten neben einer großen Anzahl von mit Mitarbeiterinnen und Studentinnen geführten Interviews eine Sammlung aller E-Mails, die im Laufe des Projektes ausgetauscht wurden, und Gesprächsnotizen während Telefonaten und Besprechungen sowie weitere Aufzeichnungen. Eine ausführliche Aufstellung des vorliegenden Materials wird in Kapitel 4.1.1 gegeben.

Basis für die Arbeit im Projektbereich Information war eine breite technische Ausstattung für die unterschiedlichsten Projekte. Allerdings musste diese technische Ausstattung im Vorwege antizipiert werden. Dabei bestand aufgrund der kurzen Einsatzzeitspanne von 3 Monaten wenig Raum für die Entwicklung von Konzepten und Lösungen. Allen Beteiligten wurde deshalb abverlangt, ihre Bedürfnisse und Anforderungen im Rahmen des Möglichen geltend zu machen. Jede Person musste mit viel Flexibilität an jede Situation herangehen, um auf Unerwartetes zu reagieren.

Das *ifu*-Projekt bietet mindestens drei Perspektiven, die hier benutzt werden können, um die Anforderungen an die technische und organisatorische Ausstattung näher zu untersuchen. Die Büro- oder Backoffice-Infrastruktur wurde geschaffen, um die Planungs- und Verwaltungstätigkeiten der lokalen Dekanin und des lokalen Personals zu unterstützen. Das Lehrpersonal (genannt *faculty*, *lecturers*, *directors* und *facilitators*) ver-

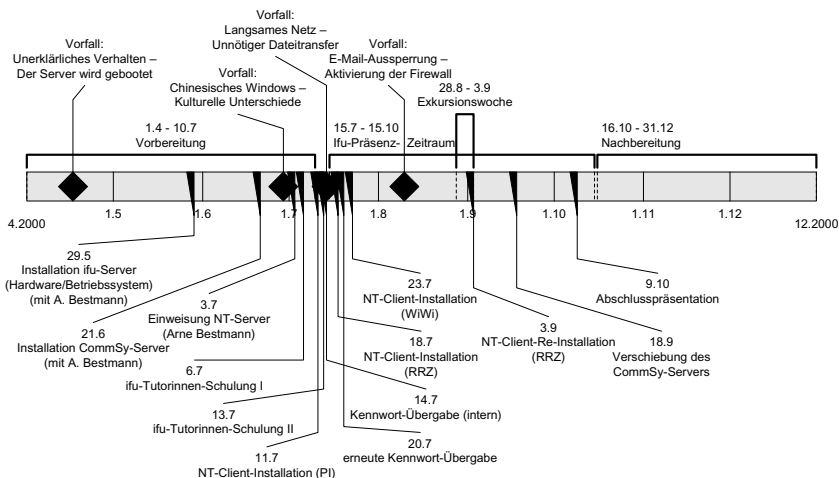


Abbildung 9: Ein Ausschnitt des Zeitverlaufs der *Internationalen Frauenuniversität*

folgte das Ziel, Lehrprojekte durchzuführen und dazu Informationstechnik möglichst innovativ einzusetzen (Floyd et al. 2000). Hierbei ging es neben der Verfügbarkeit am Campus auch um die Erreichbarkeit der technischen Dienste durch private Zugänge oder von fremden Fakultäten. Die dritte Perspektive wurde durch die Studierenden eingebracht, welche unabhängig vom Ort, an dem sie arbeiteten, die volle Funktionsfähigkeit erwarteten. Eine zentrale Frage in diesem Zusammenhang war, inwieweit die gesammelten Informationen von außerhalb verfügbar sein konnten.

## A.1 Die Büro-Infrastruktur

Der Projektbereich Information hatte aufgrund seiner Satellitenstellung die Notwendigkeit, sich selbst zu verwalten. Dafür wurde ein Büro gegründet, welches zuerst übergangsweise am Campus der Informatik (Hamburg-Stellingen) platziert wurde. Dieses Büro wurde vom Fachbereich Informatik mit Computern ausgestattet. Die Geräte wurden in die bestehende Infrastruktur des Arbeitsbereichs der lokalen Dekanin technisch und organisatorisch eingegliedert.

Nach erfolgreicher Akquise von zwei Büroräumen am Hauptcampus zog das Büro vom Stellingener Informatik-Campus im September 1999 an



den Hauptcampus der Universität. Die geliehenen Rechner sowie weitere Leihgaben von Arbeitsplatzrechnern und Druckern des Regionalen Rechenzentrums der Universität Hamburg bildeten die Grundausstattung dieses Büros.

#### *Anforderung A.1.1 (Distanz-Transparenz)*

Durch den Umzug erwuchs erstmals die Notwendigkeit, das am Informatik-Campus eingeführte technische Zusammenspiel zwischen Fileserver und Arbeitsstationen am Hauptcampus zur Verfügung zu stellen. Die Forderung begründete sich damit, dass die lokale Dekanin nunmehr zwei voneinander entfernte Arbeitsorte besaß, an denen sie dieselben technischen Möglichkeiten zur Verfügung haben wollte. (Darüber hinaus war der Aufwand, einen eigenen Fileserver einzurichten und zu betreiben, nicht verhältnismäßig und nicht ausreichend.)

Diese Anforderung konnte besonders leicht realisiert werden, weil es sich bei den Arbeitsplatzrechnern um Windows-NT-Clients handelte, die an einem Windows Domain Controller angemeldet waren. Die Universität Hamburg hatte in den vorangegangenen Jahren alle Räume mit einem IP-Netz (10/100 MBit) ausgestattet, sodass technisch eine Verbindung der Räume am Hauptcampus mit dem Server-Standort in Stellingen existierte. Es war lediglich notwendig, die Arbeitsplatzrechner im lokalen Netz mit Adressen zu versorgen und das Routing der Datenpakete zwischen den Standorten sicherzustellen.

Während des Betriebes traten mehrere Problemsituationen auf, die die oben beschriebene Distanz-Transparenz störten.

#### *Problem A.1 (Distanz-Transparenz – ausgefallener Server)*

Als Erfahrungswert in der Abteilung am Informatik-Campus gilt, dass der Windows-NT-Server etwa alle zwei Wochen ausfällt und neu gestartet werden muss. Der Administrator führt den Neustart deshalb prophylaktisch zu Zeiten, an denen niemand anwesend ist, durch, um einer Ausfallsituation vorzubeugen. Fällt der Server während der normalen Arbeitszeiten aus, trifft sich nach kurzer Zeit die Abteilung auf dem Flur und führt Smalltalk, bis der Server wieder verfügbar ist.

In beiden Situationen (prophylaktisches Neustarten, spontaner Ausfall) wurde niemals mit den Mitarbeiterinnen der Frauenuniversität gesprochen. Weder wurde vom Administrator erwartet, dass bereits um 7 Uhr morgens

jemand am Server arbeitete, noch war während der Ausfälle daran gedacht worden, die Mitarbeiterinnen am Hauptcampus zu informieren (vergleiche Vorfall 4.2.1).

*Problem A.2 (Distanz-Transparenz – unterbrochenes Netz)*

Durch Stromausfälle oder Wartungsarbeiten war die Verbindung zwischen dem ifu-Büro und dem Informatik-Campus von Zeit zu Zeit unterbrochen. Die Unterbrechung war für die Mitarbeiterinnen in der Wirkung nicht von einem ausgefallenen Server zu unterscheiden. Von Seiten des Regionalen Rechenzentrums war es unmöglich, die betroffenen Personen zu informieren, da weder ein einheitlicher Informationskanal zur Verfügung stand noch bekannt war, wo jemand arbeitete. Auf der Seite des den Server betreibenden Arbeitsbereichs war nicht bekannt, dass andere Personen vom Zugriff abgeschnitten waren.

*Problem A.3 (Distanz-Transparenz – Errichtung der Firewall)*

Parallel zum Aufbau der technischen Ausstattung für die ifu wurden vom Regionalen Rechenzentrum zusammen mit dem Rechenzentrum der Informatik Bestrebungen verfolgt, eine höhere Netzsicherheit zu etablieren. Dafür wurden Firewall-Konzepte entwickelt und implementiert, in deren Testphase die Kommunikation zwischen Hauptcampus und Informatik-Campus sehr häufig unterbrochen wurde.

Dies lag darin begründet, dass NT-Client und -server spezielle Ports für ihre Verbindung benutzen. Diese Ports wurden anfänglich gesperrt, weil nicht davon ausgegangen wurde, dass außerhalb des Informatik-Campus Zugriff auf die NT-Fileserver gewünscht wird. Darüber hinaus stellt dies ein grundsätzliches Sicherheitsproblem dar (vergleiche mit Vorfall 4.2.3).

Die Integration einer Mitarbeiterin von außerhalb stellte besondere Anforderungen an die lokalen technischen Gegebenheiten. Da sie hauptsächlich mit den zentralen elektronischen Dokumenten arbeitete und gleichzeitig eine Vielzahl von koordinierenden Aufgaben ausführte, benötigte sie direkten Zugang zu allen Daten. Hierzu musste ihr privater Arbeitsplatz eingebunden werden. Verschiedene technische Lösungen wurden diskutiert, die u. a. Lotus Notes und andere Datenbanken einbezogen, die Replizierungseigenschaften aufweisen. Allerdings stellten die ständig wechselnden Aufgaben keine klare Basis für eine Systemauswahl zur Verfügung. Darüber hinaus wurde der Entwicklungs- und Anpassungsaufwand so groß eingeschätzt,

dass diese Bemühungen nicht weiterverfolgt wurden. Man entschied sich deshalb dafür, weiterhin mit dem Fileserver zu arbeiten und einen FTP-Zugang zu den Dokumentverzeichnissen sicherzustellen. Abgleich und Versionierung wurden durch dazu abgesprochene Konventionen gelöst.

Um für die lokale Dekanin einen reibungslosen Zugriff auf Ihre E-Mail sowohl vom Arbeitsplatz am Fachbereich Informatik als auch im Büro der *ifu* sowie zu Hause sicherzustellen, wurde eine Umstellung der E-Mail-Clients durchgeführt. Vom vormals verwendeten POP-Protokoll (*post office protocol*, Myers und Rose 1988) wurde auf das IMAP-Protokoll (*internet message access protocol*, Crispin 1994) umgestellt. Wesentlicher Unterschied der Protokolle ist, dass IMAP es erlaubt, die E-Mails auf dem Server zu belassen. Damit hat man von jedem Ort dieselbe Sicht auf seine Ordner und kann gegebenenfalls sogar eine Offline-Synchronisation durchführen.

#### *Rückwirkung A.1 (ifu-E-Mail-Nutzung)*

Eine Rückwirkung dieser Umstellung erfolgte im Arbeitsbereich Softwaretechnik. Nachdem bekannt wurde, dass diese Möglichkeit besteht, haben andere Mitarbeiter ihre E-Mail auf das IMAP-Verfahren umgestellt. Hierdurch wurde eine interne Kommunikation angetrieben, in der man sich über die neuen Eigenschaften und deren Vor- und Nachteile verständigte.

## **A.2 Errichten einer Infrastruktur für die Lehre**

Ziel der Aktivitäten war die Errichtung einer Internet-basierten interaktiven Lernumgebung für die Studierenden der *Internationalen Frauenuniversität*. Diese Lernumgebung sollte in der Zeit vom 15. Juli bis zum 15. Oktober am Hauptcampus der Universität Hamburg zur Verfügung stehen und den unterschiedlichen Bedürfnissen der zwölf Projekte gerecht werden (vergleiche Tabelle 2 auf Seite 186).

Da alle Geräte als Leihgaben der Einrichtungen der Universität Hamburg genutzt wurden, standen entsprechende Ansprechpartner zur Verfügung, mit denen die Nutzungsmodalitäten ausgehandelt wurden. Trotz der freundlichen Unterstützung entstand alleine durch die vielen unterschiedlichen Personen ein hoher Aufwand in der Kommunikation und Koordination. Die verschiedenen Ansprechpartner waren Wolfgang Roehl vom Fachbereich Erziehungswissenschaft, Michael Myschik vom Fachbereich Wirtschaftswissenschaften, Angelika Rudolph am Regionalen Rechenzen-

trum, Monika Pater und Jens Schröter aus dem Institut für Journalistik und Andreas Heitmann vom Audiovisuellen Zentrum.

Bei der Planung und Vorbereitung der Lernumgebung waren mehrere besondere Anforderungen zu berücksichtigen, die im Folgenden näher betrachtet werden sollen.

### **A.2.1 Örtliche Verteilung**

Aufgrund der Gastsituation der *ifu* auf dem Gelände der Universität Hamburg war es nicht möglich, Räume nach eigenen Vorstellungen zu bekommen. Vielmehr nahm man dankbar die Räume an, die von der Universität, den Fachbereichen und anderen Einrichtungen zur Verfügung gestellt wurden. Diese waren über den gesamten Hauptcampus verteilt.

#### *Problem A.4 (Raumverteilung – Lange Wege zwischen Rechnerräumen)*

Es lagen bis zu 15 Minuten Fußweg zwischen den Rechnerräumen untereinander und von den Rechnerräumen zum Service Center.

#### *Problem A.5 (Raumverteilung – Projekt Arbeitszeiten)*

Während der Projektarbeit standen Rechnerräume nicht immer zur Verfügung, weil sie zwischenzeitlich von den Eigentümern genutzt wurden.

Rechnerräume waren von den Projekträumen teilweise zu weit entfernt (anderes Stockwerk).

#### *Problem A.6 (Raumverteilung – Ausstattung an unterschiedlichen Orten)*

Die gesamte technische Ausstattung für die *ifu* wurde von Einrichtungen der Universität Hamburg als Leihgabe zur Verfügung gestellt. Ausstattung wurde dort zur Verfügung gestellt, wo die Einrichtungen bereit waren, ihre Ressourcen zeitweise abzugeben. Allerdings waren manche Geräte (z. B. Videoschnitt) in ihrer Installation so komplex, dass sie nicht räumlich verlegt werden konnten. Studierende und Lehrende mussten dafür neben einer zeitlichen auch eine örtliche Planung erarbeiten.

### **A.2.2 Sprache**

Entgegen der üblichen Situation von ausländischen Studierenden, die an der Universität Hamburg studieren und Deutsch als Studiensprache haben, hatten die Studierenden der *ifu* Englisch als Studiensprache. Dies war ein wichtiger Punkt bei der Etablierung der lokalen technischen Arbeitsmittel und implizierte mehrere Anpassungen.

*Problem A.7 (Sprache – Tastatur)*

Durch die leihweise Verwendung bestehender Rechnerräume wurden die Rechner und Tastaturen weiterverwendet. Diese waren in Deutschland angeschafft und besaßen somit die deutsche Anordnung der Sonderzeichen, die deutschen Umlaute und – wie es in Deutschland üblich ist – die Platzierung des „Y“ links unten auf der Tastatur.

Es war aus verschiedensten Gründen nicht möglich, die Tastaturen für die *ifu* auszutauschen. Die Studentinnen und Dozentinnen mussten mit dieser Situation lernen umzugehen (vergleiche Vorfall 4.2.4).

*Problem A.8 (Sprache – Betriebssystem)*

Obwohl alle Rechner bereits mit einem Betriebssystem installiert waren, musste für die *ifu* eine alternative Version installiert werden. Nur so konnte sichergestellt werden, dass ein englisches Betriebssystem mit englischsprachiger Software vorhanden war.

Die Neuinstallation des Betriebssystems und der Anwendungssoftware stellte insofern ein Problem dar, als zwischen Semesterende der Hamburger Studierenden und Beginn der *ifu* nur wenige Tage lagen. Die Installation aller Rechner musste demnach gut vorgeplant und dann in einem effizienten Verfahren durchgeführt werden. Als englische Version stand zumeist nur die europäische englische Version zur Verfügung, die sich von der amerikanischen englischen Version unterscheidet.

Die allgemeine Anwendungssoftware (Textverarbeitung usw.) konnte in allen Fällen in englischer Sprache besorgt werden. Allerdings war dies bei spezieller Software (Videobearbeitung) und Treibern (z. B. für Drucker oder Scanner) nicht immer einfach oder möglich. Entweder lag die englische Version nicht vor oder sie war nicht kostenlos beziehbar.

*Problem A.9 (Sprache – Beschriftungen)*

Schalter und Tasten an den Geräten waren größtenteils in Deutsch beschriftet. Dies wurde bei den meisten Geräten zu einem Problem, da Beschriftungen in Dialogboxen auf dem Bildschirm (englische Software) nun nicht mehr mit den Geräten (deutsche Beschriftung) übereinstimmten.

*Problem A.10 (Normen – DIN A4)*

Die Drucker(-treiber) waren in allen Räumen auf DIN A4 eingestellt. Allerdings waren die meisten mitgebrachten Dokumente mit dem Papierformat Letter vorformatiert, welches in Asien und Nordamerika üblich ist. Die

meisten Drucker sind so eingestellt, dass sie solche Dokumente zum Ausdruck annehmen, dann aber warten, bis entsprechendes Papier bereitgestellt wird. Eine Aufforderung dazu wird im Display des Druckers angezeigt. Die Studierenden konnten mit den Fehlermeldungen der Drucker (in Deutsch) nur wenig anfangen. Teilweise wurden diese nicht wahrgenommen (vergleiche Vorfall 4.2.4).

### **A.2.3 Technik**

Obwohl bei allen Studierenden eine Vorerfahrung in der Rechnernutzung vorlag, waren es nicht alle gewöhnt, an Systemen zu arbeiten, die personalisiert, d. h. mit Login-Mechanismus arbeiten. Die Zugangskonten wurden in Berlin verwaltet. Etwa eine Woche vor Beginn der Präsenzlehre wurden die Konten-Namen und Kennworte übermittelt.

#### *Problem A.11 (Technik – Kennworte verteilen)*

Das organisatorische Problem bestand darin, zu einer Tabelle, deren Zeilen aus Personennamen, Konten-Namen und Kennwort bestanden, ein Verfahren zu entwickeln, diese Informationen an die betreffenden Personen zu verteilen. Hierfür wurde ein Dokument entwickelt, welches mithilfe eines Serienbriefes für jede Studentin eine kurze Beschreibung, den Login-Namen und das Kennwort enthielt (vergleiche Abbildung 13 auf Seite 216).

Das damit verbundene technische Problem lag in der Generierung der Kennworte. Um diese möglichst sicher zu gestalten, wurde eine Zahlen- und Buchstabenkombination generiert, die sowohl Klein- als auch Großbuchstaben enthielt. Es war nun für die Adressaten des Serienbriefes nicht erkennbar, ob es sich um eine „1“ (Eins) oder ein „i“ (Buchstabe i) oder ein „l“ (Buchstabe l), eine „0“ (Null) oder ein „o“ (Buchstabe o) usw. handelte. Dies lag zum einen an der gewählten Schriftart, war aber zum anderen durch die ungewohnte Situation, ein generiertes Kennwort zu bekommen, begründet.

Es wäre nützlicher gewesen, das Kennwort aus einfachen (englischen) Worten zusammenzusetzen und mit Ziffern von 2 bis 9 zu kombinieren. Dabei würde man neben den Ziffern „0“ und „1“ bei den Worten die Buchstaben „o“, „i“ und „l“ vermeiden. Eine hohe Sicherheit wäre damit immer noch durch die Kombination und die Verwendung von Ziffern gewährleistet gewesen.

*Problem A.12 (Organisation – An- und Abmelden)*

Nicht allen Studierenden war ersichtlich, welchen Zweck das persönliche Anmelden hat. Insbesondere meldeten sie sich nicht immer ab, sodass andere Studierende (außerhalb der *ifu*) ihre Konten benutzen konnten.

Damit hatten Außenstehende Zugriff auf *ifu*-Geräte. Das führte dazu, dass die Arbeitsplätze unnötig blockiert waren, dass persönliche Einstellungen unerwartet geändert wurden und dass potenzielle Sicherheitsrisiken entstanden.

*Entwurfsentscheidung A.1 (Serverseitige Profilspeicherung)*

Um sicherzustellen, dass die Studierenden an jedem Rechner ihre Einstellungen und Dokumente vorfanden (sowie Favoriten, zuletzt verwendete Dokumente etc.), wurde unter dem Windows-NT-Server die Möglichkeit genutzt, die Profilinformationen auf dem Server zu speichern.

Windows realisiert diese Funktionalität, indem das lokal abgelegte Profil am Ende der Sitzung auf den Server kopiert wird. Im Anmeldevorgang wird das zuletzt abgelegte Profil auf die lokale Arbeitsstation kopiert.

Zum Erkennen von Viren und zu deren Vermeidung wurde auf dem Fileserver ein Virenschutzprogramm installiert. Dieses untersucht jeden Speichervorgang an einer Datei und verhindert diesen, wenn ein Virus in der Datei erkannt wird. Im optimalen Fall wird der Virus aus der Datei entfernt; alternativ wird die Datei in Quarantäne verlegt.

*Problem A.13 (Viren – Makroviren)*

Der oben beschriebene Virenschutz unterstellt, dass alle Speicherungen auf dem Fileserver durchgeführt werden. Dies war vielfach nicht der Fall. Studierende und Lehrende legten einen Großteil ihrer Dateien (hauptsächlich Microsoft-Word-Dokumente) auf dem Desktop ab. Dieser unterlag nicht der Prüfung, da er auf dem lokalen Rechner gespeichert ist. Ursache war hier ein unzureichender lokaler Virens Scanner. Beim Abmelden wurden die Dateien des Desktops auf den Server kopiert (vergleiche Entwurfsentscheidung A.1) und auf dem nächsten Rechner wiederhergestellt. Dieser Mechanismus unterlief die Virenprüfung auf dem Server, was zur Ausbreitung beitrug.

*Entwurfsentscheidung A.2 (Netscape-User-Profiles)*

Um sicherzustellen, dass die Studierenden an jedem Rechner ihre Browser- und E-Mail-Einstellungen vorfinden (sowie Favoriten, Adressbuch etc.),

wurde der Netscape Communicator so konfiguriert, dass er die Profilinformation nicht auf der lokalen Maschine ablegt, sondern auf einem Directory-Server.

Dies erfordert, dass man sich beim Start des Communicators zuerst autorisiert, damit die Profilinformationen vom Server geladen werden können. Danach kann der Communicator wie gewohnt verwendet werden. Beim Beenden des Communicators kopiert dieser selbstständig die geänderten Einstellungen auf den Server.

Die vorbereitete und zusammengestellte Installation beinhaltete den Internet Explorer 4.x. Kurz vor Beginn der *ifu* erschien die neue Version 5.0 des Produktes, in der viele bekannte Fehler behoben waren. Dies führte zu der Entscheidung, anstelle der älteren Version die neue zu installieren.

#### *Problem A.14 (Technik – Browser-Cache)*

Während in den Versionen vor 5.0 der Internet Explorer seinen Cache<sup>1</sup> in einem temporären Verzeichnis unterhalb des Betriebssystems ablegt, wird hierfür in den Versionen ab 5.0 ein benutzerbezogenes Verzeichnis unterhalb der Profilinformationen verwendet. Dieses Verzeichnis wurde aber wegen der Entwurfsentscheidung A.1 auf der vorherigen Seite beim Abmelden auf den Server und beim Anmelden auf den Client kopiert (vergleiche Vorfall 4.2.2).

Obwohl dieses Problem erkannt worden war, gestaltete es sich vielschichtig, eine Lösung zu etablieren. Erstens mussten die Cache-Dateien für jeden Benutzer gelöscht werden, um die Symptome zu beseitigen, und zweitens mussten die Einstellungen des Internet Explorer für jeden Benutzer so geändert werden, dass die Cache-Dateien nur lokal auf der jeweiligen Maschine gelagert wurden, damit die Ursache behoben war.

Dies wurde dadurch gelöst, dass mithilfe einer Batch-Datei die Cache-Dateien regelmäßig auf dem Server gelöscht wurden. Somit konnte die Übertragung auf die Arbeitsstationen vermieden werden. Dies löste allerdings nicht das Problem, dass die Benutzerinnen bei der Abmeldung oh-

---

<sup>1</sup> Jeder Browser speichert alle geladenen Dateien zwischen. Während einer Sitzung werden die bereits geladenen Objekte (Seiten und Grafiken) im Hauptspeicher gehalten, und darüber hinaus werden diese für einige Tage auf der Festplatte gespeichert. Beides geschieht, um während der Navigation die Ladezeiten zu verkürzen, da es sehr wahrscheinlich ist, dass bereits geladene Objekte mehrfach aufgerufen werden.



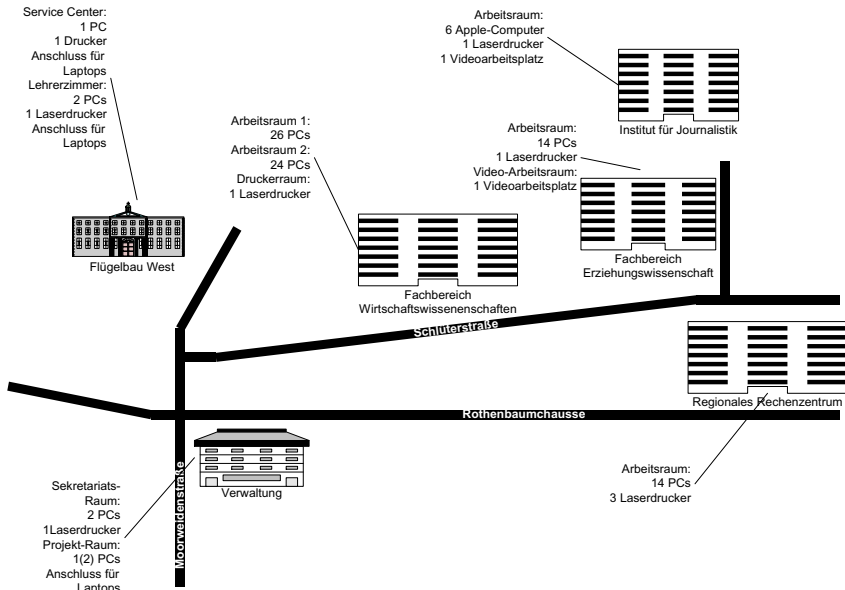


Abbildung 10: Lageplan der akquirierten Räumlichkeiten am Hauptcampus mit technischer Ausstattung

ne ihr Wissen neue Cache-Dateien auf den Server kopierten. Hierfür wurden mehrere E-Mails verschickt, die die Benutzerinnen baten, ihre Browser-Einstellungen zu ändern (Beleg E-Mail 31-07-2000). Der Erfolg dieser Aufrufe war leider nur gering.

Die Universität Hamburg leistete eine breite Hilfestellung für die *ifu*. Jede Einrichtung (Fachbereiche, Rechenzentrum), die angesprochen wurde, stellte Teile ihrer Räumlichkeiten, Geräte und Dienstleistungen zur Verfügung. Allerdings hatte keine Einrichtung so viel Kapazität, dass die gesamte *ifu* in ein Gebäude einziehen konnte. Insgesamt sechs verschiedene Gebäude sind deshalb genutzt worden:

- Moorweidenstraße 18: Hier wurden zwei Räume für etwa fünf bis sieben Personen als Büro genutzt. Vier Arbeitsplatzrechner standen für administrative Aufgaben zur Verfügung. Anschlüsse für Laptops waren verfügbar. Das Büro stand der *ifu* von März 1999 bis Dezember 2000 zur Verfügung.

- Flügelbau West: Im Flügelbau wurde ein Zwei-Zimmer-Büro als so genanntes *Service-Center* mit Theke genutzt. Hier stand ein Arbeitsplatzrechner mit Drucker für administrative Zwecke zur Verfügung (dieser Computer wurde aus der Moorweidenstraße während der Präsenzzeit entnommen); Anschlüsse für Laptops waren ebenfalls nutzbar. Das Büro war in der Zeit von Juli 2000 bis Ende Oktober 2000 benutzbar. Ein Vorbereitungsraum für die Lehrenden befand sich im 1. Stock des Gebäudes, in dem ebenfalls zwei Arbeitsstationen und ein Drucker standen.
- Pädagogisches Institut (PI): Im PI stand ein Raum mit 14 Computern und einem Laserdrucker sowie einem Farbdrucker im obersten Geschoss zur Verfügung.
- Institut für Journalistik (IfJ): Im IfJ stand im 1. Stock ein Raum mit 6 Apple-Computern zur Verfügung. Zu dieser Ausstattung gehörten ein Laserdrucker und ein Farbdrucker. Eine Station zur Videobearbeitung war ebenfalls vorhanden.
- Wirtschaftswissenschaften (WiWi): Hier wurden zwei Räume mit je 25 Arbeitsplätzen bereitgestellt. Ein Laserdrucker war in einem Nachbarraum aufgestellt.
- Regionales Rechenzentrum (RRZ): Das RRZ konnte einen Raum zur PC-Nutzung bereitstellen, in dem 14 Computer standen. Im Nachbarraum waren insgesamt drei Laserdrucker verfügbar.

Dies sind allerdings nur die Räume, in denen Zugriff auf Computer möglich war. Darüber hinaus wurde eine große Anzahl von Seminar- und Projekt-Räumen verwendet.

#### *Problem A.15 (Organisation – Sechs Gebäude)*

Durch die sechs verschiedenen Gebäude waren längere Wege in Kauf zu nehmen, unterschiedliche Ausstattung und Ansprechpartner vorhanden und der Zugang war individuell geregelt.

#### *Problem A.16 (Organisation – Unzureichende Arbeitsplätze)*

Obwohl durch die o. g. Rechnerräume den Studierenden insgesamt 84 Arbeitsplätze zur Verfügung standen, reichte diese Anzahl nicht immer aus. Hauptgrund für diesen nicht erwarteten Engpass war, dass die Studierenden alle zur selben Zeit die Rechner nutzen wollten (z. B. vor der morgendlichen Vorlesung).

Tabelle 3: Dienste des Fileservers

| Dienst          | Beschreibung  |
|-----------------|---|
| Dateidienste    | Bereitstellung von ca. 6 GB Speicherplatz                   |
| Backup          | regelmäßige Datensicherung                                  |
| Virenprüfung    | Untersuchung und Entfernung von Viren während des Betriebes |
| FTP             | Zugriff auf den Speicherplatz mittels FTP                   |
| Druckdienste    | Betrieb von 6 Drucker-Warteschlangen                        |
| PDF-Generierung | Umwandlung von Dokumenten in PDF                            |

### *Problem A.17 (Organisation – Unterschiedliche Ausstattung)*

Aufgrund der unterschiedlichen Einrichtungen der Universität Hamburg, die ihre Rechnerräume zur Verfügung stellten, waren die Rechner selbst von unterschiedlichem Alter und besaßen unterschiedliche Ausstattungsmerkmale. Dies betraf sowohl die Rechenleistung als auch das Vorhandensein von besonderen Speichermedien (ZIP-Laufwerke) oder von Soundkarten.

## **A.2.4 Aufbau des Hamburger ifu-Servers**

Der Hamburger ifu-Server wurde primär als Fileserver benutzt und war in dieser Eigenschaft die zentrale technische Einrichtung zum Ablegen und Austauschen von Daten.<sup>2</sup> Es handelte sich um einen Windows-NT-4.0-Server (Pentium Pro mit 192-MB-Hauptspeicher und einer 2-GB-System-Festplatte sowie 20-GB-Daten-Festplatte). Neben der Hauptaufgabe, Speicherplatz für Dateien zur Verfügung zu stellen, wurden auf dem Fileserver mehrere weiterführende und ergänzende Dienste angeboten (siehe Tabelle 3).

Die Abbildung 11 auf der nächsten Seite zeigt die Struktur der Verzeichnisse, wie sie sich den Studierenden darstellte.

Unter der Freigabe „home“ fanden die Studierenden eine Reihe von Verzeichnisbäumen vor, die eine Organisation der Dateiablage vorgaben. Der Bereich „projects“ enthielt Verzeichnisse für alle Projekte mit den vordefinierten Unterverzeichnissen „public“ und „private“, die entsprechend für interne oder öffentliche Dokumente genutzt werden konnten. Im Baum „user“ befanden sich die persönlichen Verzeichnisse der Studentinnen, Mit-

<sup>2</sup> Der Aufbau und Betrieb des Fileservers sowie der Arbeitsstationen wurde freundlicherweise von Arne Bestmann unterstützt.

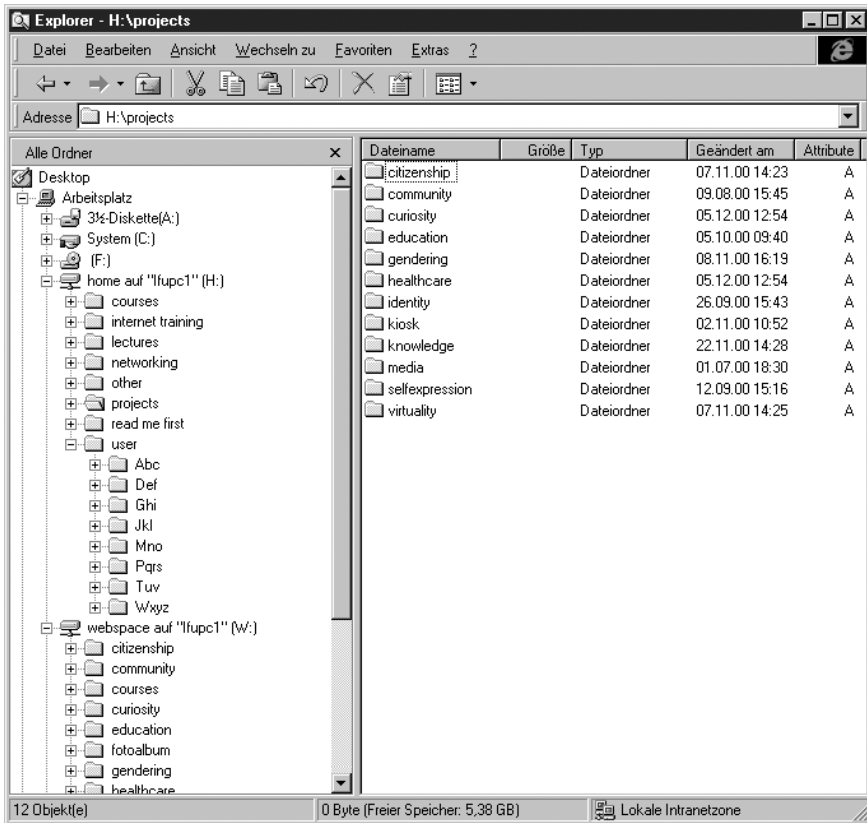


Abbildung 11: Struktur der Verzeichnisse auf dem Fileserver

arbeiterinnen und Dozentinnen. Diese waren aufgrund der hohen Teilnehmerinnenzahl noch einmal in Buchstabengruppen aufgeteilt. Die Bereiche „courses“, „lectures“ und „internet training“ enthielten die entsprechenden Kursunterlagen. Im Verzeichnis „Drucker“<sup>3</sup> befanden sich alle verfügbaren Drucker mit den Raumbezeichnungen als Namen (leider nicht in der Abbildung zu sehen).

3 Dieses Verzeichnis trägt auf englischen Arbeitsstationen den Namen „Printer“; die Abbildung wurde auf einer deutschen Arbeitsstation erstellt, weil zu diesem Zeitpunkt keine englische Installation mehr zur Verfügung stand.

### *Entwurfsentscheidung A.3 (Homeverzeichnisse)*

Die persönlichen Verzeichnisse der Benutzer wurden auf der Freigabe „home“ im Verzeichnis „user“ angelegt. Diese Freigabe wurde den Anwendern unter dem Laufwerksbuchstaben „H“ automatisch zur Verfügung gestellt.

Neben dem reinen Angebot von Speicherplatz für Dateien ist es notwendig, andere Dienste anzubieten, um das Gesamtangebot zu komplettieren. Der Server-Betreiber geht die Verpflichtung ein, eine regelmäßige Datensicherung (Backup) anzufertigen und muss gleichzeitig sicherstellen, dass diese jederzeit wieder eingespielt werden kann. Die Sicherheit der Daten muss durch Virenschutz gewährleistet werden; eine Überprüfung der Dateien im Moment des Speicherns ist unumgänglich. Darüber hinaus wird der Zugriff auf die Dateien außerhalb des Windows-Netzes gefordert, das im Allgemeinen durch FTP ermöglicht werden kann.

Die Druckdienste in einem NT-Netz werden ebenfalls von dem Fileserver angeboten, da dieser direkt zur Verfügung steht und die Auswahl erleichtert. Dazu wurden alle Drucker in den betreffenden Räumen (siehe Abbildung 10 auf Seite 197) über den Fileserver verbunden und entsprechende Warteschlangen eingerichtet.

Die Erzeugung von PDF-Dokumenten hat sich im wissenschaftlichen Bereich durchgesetzt und ist besonders vorteilhaft, um komprimierte plattformunabhängige Dokumente weiterzugeben, die zur Ansicht kein kostenpflichtiges Produkt erfordern.

## **A.2.5 Aufbau des Webservers**

Auf dem Hamburger *ifu*-Server wurde zusätzlich zum Fileserver ein Webserver eingerichtet (Apache 1.3.12), um den Studierenden die Möglichkeit zu geben, mit eigenen Webseiten zu experimentieren und das *CommSy* zu betreiben (siehe hierzu Kapitel 4.4). Konkret wurden folgende Web-Bereiche angeboten:

- allgemeine Webseiten (Information und Navigation),
- 13 separate Projekträume (*CommSys*),
- frei gestaltbarer Bereich für jedes Projekt,
- dynamische Webseiten für einzelne Projekte.

Zu dem Webserver kam ergänzend ein Datenbankserver (MySQL 3.22.32), in dem die *CommSy*-Einträge abgelegt wurden und für die dynamischen Webseiten ebenfalls Tabellen angelegt worden sind.

Im Betrieb stellte sich heraus, dass die Kombination von Fileserver mit den o. g. Diensten und Webserver mit Datenbankserver keine günstige Zusammenstellung war. Der Speicherausbau des Gerätes bzw. die Betriebssystemarchitektur führten bei gleichzeitiger Nutzung aller Dienste zu einem starken Leistungseinbruch (kontinuierliches Auslagern und Einlesen von ausgelagerten Hauptspeicherbereichen), sodass noch während der Präsenzphase die Entscheidung gefällt wurde, die *CommSys* (wegen ihrer intensiven Nutzung von dynamischen Seiten und der Datenbank) von dem Server auf einen anderen zu verlagern.

#### *Zusammenspiel von Fileserver und Webserver*

Die Notwendigkeit, sowohl Fileserver als auch Webserver auf derselben Maschine zu betreiben, liegt darin begründet, dass die vom Webserver ausgelieferten Seiten als Dokumente im Dateisystem gespeichert sind. Es ist deshalb am effizientesten, wenn der Webserver auf der Maschine läuft, auf der die Dateien physikalisch liegen. Das Dokumenten-Verzeichnis des Webservers lag deshalb auf der Festplatte „Data“ des Fileservers.

Darüber hinaus gab es eine Freigabe „webSPACE“, die ebenfalls durch den Webserver im WorldWideWeb zugreifbar war. Unterhalb der Freigabe waren die Unterverzeichnisse für jedes Projekt mit den jeweiligen Projektnamen angelegt. Die Projekte konnten hier Dokumente ablegen, die danach über das WWW zugänglich waren. Darüber hinaus gab es jeweils einen Ordner „private“, dessen Inhalt nur für die Mitglieder des Projektes zugänglich war. Hier konnten Experimente durchgeführt bzw. Testapplikationen entwickelt werden.

### **A.2.6 E-Mail-Server und Server für Benutzerkonten**

Der E-Mail-Server und der Server für die Benutzerkennungen wurden vom Projekt *vifu* in Berlin betrieben. Dort stand ein Linux-PC-Server zur Verfügung, auf dem entsprechende Dienste konfiguriert waren. Die E-Mail-Postfächer konnten per POP3 und IMAP4 angesprochen werden. Der Versand wurde über SMTP unterstützt. Die Kennworte wurden mit einem Directory-Server verwaltet. Um dieselben Kennworte in Hamburg für die Anmeldung

unter Windows NT zu verwenden, die für E-Mail-Postfächer und geschützte Webseiten verwendet wurden, wurde der Synchronisations-Dienst des Directory-Servers genutzt. Alle Informationen über Benutzerkennungen für die Studierenden und Lehrenden in Hamburg wurden zwischen dem Hamburger Fileserver und dem Berliner Server automatisch abgeglichen.

Der Netscape Communicator stellt eine Kombination von Webbrowser, E-Mail-Client, News-Reader und HTML-Editor dar. Zusammen mit seinen bestehenden Integrationen von anderer Software (Plug-In-Standard) und der Plattformunabhängigkeit hielt das *vifu*-Team ihn für geeignet (Beleg E-Mail 23-02-2000 bs, Entscheidung auf der Januar-Sitzung des Drittmittel-Projektes) zum Einsatz für die *ifu*. Die Installation unter den in Hamburg verfügbaren Plattformen stellt sich einfach dar. Um sicherzustellen, dass mehrere Standardeinstellungen bereits vorgenommen wurden, stellt Netscape das so genannte Client-Customization-Kit (Netscape Corporation 2000) zur Verfügung. Es ist für unternehmensweite Installationen gedacht, bei denen sowohl Vorbelegungen als auch Einschränkungen durchgeführt werden. In unserem Einsatzfall wurde neben der Startseite im Webbrowser die E-Mail-Anwendung vorkonfiguriert und das Adressbuch der *ifu* eingestellt.

#### *Entwurfsentscheidung A.4 (Voreinstellungen)*

Um es den Studierenden zu erleichtern, die „richtige“ Homepage zu finden, den Mailserver korrekt einzustellen und andere Basis-Einstellungen am Netscape Communicator durchzuführen, wurden diese Voreinstellungen in einer vorbereiteten Konfigurationsdatei abgelegt.

#### *Entwurfsentscheidung A.5 (Freie Einstellungen)*

Es wurde den Studierenden bewusst die Möglichkeit gelassen, alle Einstellungen im Netscape Communicator weiterhin selbst durchzuführen. Es wäre möglich gewesen, in der Konfigurationsdatei einzelne Bereiche zu fixieren, sodass sie nicht mehr hätten geändert werden können. Mit Rücksicht darauf, dass für bestimmte Bedürfnisse gegebenenfalls andere Einstellungen sinnvoller sein könnten, wurde hierauf verzichtet.

#### *Problem A.18 (E-Mail – Voreinstellungen)*

Die gewählten Voreinstellungen für die E-Mail-Clients führten insofern zu Verwirrung, als dass ohne Änderung der Einstellungen das Programm noch

Tabelle 4: Spezielle Einstellungen des Netscape Communicators für den Betrieb bei der *ifu*

| Einstellung      | Vorbelegung                      | Erläuterung  |
|------------------|----------------------------------|--|
| Homepage         | www.ifu.uni-hamburg.de           | Homepage des Projektbereichs   |
| Sprache          | Englisch bevorzugt               | Universitätssprache  |
| Cookies          | Nur Cookies für denselben Server | minimale Sicherheit  |
| Proxy            | WWW-Cache des RRZ                | Cache des lokalen Rechenzentrums                                       |
| Cache-Domain     | uni-hamburg.de                   | Seiten, die nicht gecached werden                                      |
| Mail Identity    | International Women's University | autom. Angabe im Mail-Kopf   |
| Mailserver       | mail.vifu.de                     | Mailserver der <i>ifu</i>  |
| Directory-Server | ldap.vifu.de                     | Adressbuch der gesamten <i>ifu</i>                                     |
| Roaming Server   | ldap.vifu.de                     | Speicherung persönliches Adressbuch und Bookmarks auf zentralem Server |

nicht nutzbar war. Die Studierenden nahmen aber an, dass bereits alle persönlichen Eintragungen vorhanden waren, weil sie sich mit ihrer Kennung unter Windows angemeldet hatten.

#### *Problem A.19 (E-Mail – Mehrere E-Mail-Konten)*

Der Netscape Communicator bietet in der verwendeten Version keine Möglichkeit, mehr als ein E-Mail-Konto zu verwalten, es sei denn, alle Konten unterliegen dem IMAP-Standard. Dies war zwar für die E-Mails der *ifu* der Fall, reichte aber nicht aus, wenn die Studentinnen ihre private oder bei einer anderen Organisation beherbergte E-Mail lesen wollten. In diesem Fall mussten sie entweder den E-Mail-Client umkonfigurieren, wobei sie ihre *ifu*-Einstellungen verloren, oder über einen anderen Weg die E-Mail lesen.

#### *Problem A.20 (E-Mail – Verlorene E-Mails)*

Aufgrund der Gegebenheiten nutzten die Studentinnen jeweils unterschiedliche Arbeitsplätze zum Lesen der E-Mails (jeweils die zu dem Zeitpunkt verfügbaren). Nur bei der IMAP-Konfiguration wurden die Mails auf dem Server belassen und waren somit beim Wechsel des Arbeitsplatzes weiterhin verfügbar. Wechselten die Studentinnen auf persönliche E-Mail-Konten, die über POP3 angesprochen wurden, wurden die E-Mails zumeist auf



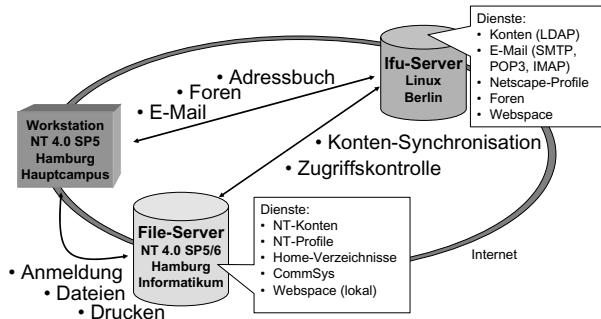


Abbildung 12: Verwendete Dienste nach Standorten

den lokalen Rechner kopiert und waren am nächsten Arbeitsplatz nicht ansprechbar. Hier wurde meist eine „leere Inbox“ beklagt.

### A.2.7 Zusammenspiel von Directory-Server und lokalen Servern

Der Directory-Server zur Verwaltung aller Konten der *ifu* wurde in Berlin betrieben. Auf ihm waren ca. 1500 Konten auf der Basis der Immatrikulationen und Personallisten angelegt worden. Auf der Basis dieses Kontenstammes bietet der Berliner Server E-Mail-Adressen und Postfächer an.

Die Konten für die lokale NT-Domäne in Hamburg wurden durch Spiegelung der entsprechenden Berliner Konten erstellt. Dazu wurde den entsprechenden Konten ein zusätzliches Attribut hinzugefügt, welches über ein Synchronisations-Plug-In mit dem Hamburger NT-Server kommunizierte. Änderungen auf dem Directory-Server wurden so mit dem NT-Server synchronisiert und umgekehrt.

Geschlossene Benutzergruppen für Webseiten und andere Anwendungen lassen sich über den Directory-Server realisieren, indem dort auf der Grundlage der vorhandenen Konten Gruppen angelegt werden, deren Mitgliedschaft Voraussetzung für einen Zugriff ist. Diese Möglichkeit wurde für experimentelle Webseiten und z. B. das *CommSy* benutzt.

Einen Überblick über die Zusammenhänge zwischen den Standorten Berlin und Hamburg gibt die Abbildung 12.

### A.2.8 Lernumgebung *CommSy*

Eine zentrale Aufgabe beim Aufbau der technischen Ausstattung für die *ifu* war die Bereitstellung einer interaktiven Lernumgebung. Diese sollte in den zwölf Projekten des Projektbereichs Information genutzt werden. Das Community System *CommSy* ist eine web-basierte Lernumgebung, die zur Unterstützung von Lerngruppen in projektorientierter Arbeit verwendet werden kann (siehe Kapitel C.1). Um ein *CommSy* als Lernumgebung verwenden zu können, sind neben organisatorischen mehrere technische Voraussetzungen zu erfüllen, siehe dazu Tabelle 11 auf Seite 271.

Darüber hinaus wurden für die *ifu* Erweiterungen am *CommSy* durchgeführt, ohne die der Einsatz nicht möglich gewesen wäre. Dazu zählt die Internationalisierung der Software. Das System musste in englischer Sprache genutzt werden können. Als Zweites musste ein Anschluss des Systems an den Directory-Server der *ifu* hergestellt werden, damit keine zusätzlichen Konten verwaltet werden müssen. Drittens wurde die Datenbankstruktur so verändert, dass mehrere Systeme in derselben Datenbank untergebracht werden konnten, um den zwölffachen Verwaltungsaufwand zu vermeiden. Nähere Informationen zu diesen Änderungen und Erweiterungen des *CommSy* finden sich in Kapitel C.2.4.

Im Rahmen des *ifu*-Einsatzes wurden am *CommSy* zwei studentische Projekte durchgeführt. Das erste Projekt trug den Titel „Workspaces“ und ergänzte das *CommSy* um die Möglichkeit, im System Arbeitsergebnisse mithilfe von strukturierten HTML-Dokumenten festzuhalten und zu präsentieren. Im zweiten Projekt (Jeenicke 2001, Kapitel 3) wurde die damals rudimentär vorhandene Quellenverwaltung an wissenschaftliches Arbeiten angepasst.

## A.3 Infrastruktur-Perspektiven

Neben der Perspektive der Infrastruktur-Gestalter, die in diesem Kapitel vorherrschend ist, gibt es die Mitarbeiterinnen in der Verwaltung, die Lehrenden und die Studierenden. Darüber hinaus fließen die Perspektiven der beteiligten Organisationen mit ein.

Die Mitarbeiterinnen in der Verwaltung der *ifu* äußerten keine großen Ansprüche bezüglich der technischen Ausstattung ihrer Arbeitsplätze. Darüber hinaus war auch kein großes Interesse für die Struktur und den Aufbau

der technischen Ausstattung vorhanden. Dies liegt darin begründet, dass sie der technischen Infrastruktur allgemein wenig Interesse schenkten, denn bei der von ihnen geleisteten Arbeit ist eine funktionierende und einfach zu benutzende technische Ausstattung erforderlich und gleichzeitig ausreichend.

Die Perspektive der Studierenden ist keine einheitliche. Sie lässt sich nur durch subjektiv ausgewählte Beispiele andeuten. Viele Studierende äußerten das Bedürfnis, ihre private und wissenschaftliche Korrespondenz fortzuführen. Dafür war im Allgemeinen ein Zugang zu ihrem bereits bestehenden E-Mail-Konto notwendig. War dies nicht über das WWW erreichbar, ergaben sich Probleme bei der Einrichtung des E-Mail-Client. Ein signifikanter Anteil der ausländischen Software und Webseiten nutzt internationale Zeichensätze, die nicht vorhanden bzw. installiert waren. Die Studierenden äußerten den Wunsch, diese selber installieren zu können oder installiert zu bekommen.

*Problem A.21 (Schriften – internationale Zeichensätze)*

Auf den Arbeitsplatz-Rechnern waren nur zentraleuropäische Zeichensätze installiert. Studentinnen anderer Länder äußerten den Wunsch, z. B. koreanische Zeichensätze im Web-Browser nutzen zu können. Dies musste abgelehnt werden, weil sonst diverse außereuropäische Zeichensätze auf allen Arbeitsstationen hätten installiert werden müssen.

*Problem A.22 (Zugriff – am Wochenende)*

Obwohl die Arbeit an den fünf Tagen in der Woche von allen Beteiligten bereits viel Zeit in Anspruch nahm, wünschten trotzdem einige die Nutzung von Arbeitsstationen am Wochenende. Dies war nur in einem Raum möglich (RRZ mit 14 PCs), da die anderen Gebäude verschlossen waren und in den Räumen keine Aufsicht vorhanden war.

Die Lehrenden der *ifu* hatten eine besondere Rolle und eine besondere Situation. Dabei lassen sich unterschiedliche Rollen benennen: *Lecturers* waren Wissenschaftlerinnen, die für kurze Zeit eingeladen waren und neben einer Vorlesung gegebenenfalls noch kurze Kurse gaben. *Facilitators* und *Project Directors* kümmerten sich während der drei Monate um ihr Projekt und leiteten die Studierenden bei ihrer Arbeit an.

Alle diese eingeladenen Lehrenden verbanden mehrere Eigenschaften. Erstens waren sie in ihre eigene Lehre am Heimatort eingebettet und insofern war für sie die *ifu* nur ein Zwischenspiel und Experiment. Zweitens

stellte ein internationales Kollegium alle Beteiligten vor neue Herausforderungen. Die meisten besaßen keine Ortskenntnisse, fast alle hatten Englisch nicht als Muttersprache und es gab wenig Zeit der Vorbereitung, sodass erst kurz vor Beginn das vollständige Kollegium versammelt war.

Insbesondere für die Lehrenden war der Zugriff außerhalb der Öffnungszeiten der Räume wichtig. Sie wollten aus ihren Unterkünften über ihren Internet-Provider oder ihre lokale Universität auf die Ressourcen der *ifu* zugreifen. Eine Möglichkeit bestand darin, die Modem-Einwahl mit eigenem Computer über das Regionale Rechenzentrum der Universität Hamburg zu benutzen.

#### *Problem A.23 (Zugriff – aus der Unterkunft)*

Die Zugriff auf größere Teile der Infrastruktur von außen war zum Start der Präsenzphase noch nicht verfügbar, da hierfür nicht genügend Vorbereitungszeit vorhanden war. Die Lehrenden forderten dies aber, um während ihrer Reisen und von ihren Unterkünften aus weiterhin arbeiten zu können.

#### *Problem A.24 (Installation – andere Betriebssysteme)*

Andere Wissenschaftlerinnen benötigten für mitgebrachte Programme Betriebssysteme in der entsprechenden Landessprache (z. B. Chinesisch). Dafür existierte kein organisatorischer Rahmen. Es reichte nicht aus, dass sie eine Installations-CD des jeweiligen Betriebssystems mitbrachten, da z. B. keine Vorkehrungen (Personal, technische Einrichtungen) existierten, beliebige Geräte für die entsprechenden Lehrveranstaltungen umzukonfigurieren.

#### *Problem A.25 (Zugriff – während der Lehrzeiten)*

Für die Vorbereitung ihrer Lehrveranstaltungen benötigten mehrere Lehrende Zugriff auf Rechner und das Internet. Da sie keine eigenen Räume an der Universität besaßen (Lehrerzimmer oder Büro), mussten sie die Rechnerräume verwenden, die den Studierenden zur Verfügung standen. Dies führte zu Konflikten, da in den Stoßzeiten bereits nicht für alle Studierenden gleichzeitig Rechner zur Verfügung standen.

### **A.3.1 Sprache und Kultur**

Um über Infrastruktur reden zu können, müssen einige Grundvoraussetzungen erfüllt sein. Es muss ein gemeinsames Verständnis darüber existieren,

was Bestandteil der Infrastruktur ist und was nicht. Um sich darüber zu verständigen, ist eine gemeinsame Sprache nötig. Die Kommunikation fand dabei auf unterschiedlichen Ebenen zwischen unterschiedlichen Personenkreisen statt. Als Ebenen sind zu nennen die Lehrinhalte und damit verbundenen Handlungen, die Verwaltung der Organisation sowie die technische Administration der Geräteausstattung. Die Personenkreise entsprachen den bisher eingeführten: Mitarbeiterinnen der *ifu*, Studentinnen, Lehrende, Studierende der Universität Hamburg, Mitarbeiter der einzelnen Fachbereiche und Einrichtungen sowie die Personen zur technischen Administration.

Sprache, die Art der Unterhaltung und die Vorannahmen übereinander sind über die Kultur festgelegt. Die *ifu* war ein internationales und interkulturelles Projekt. Schon aus diesem Grund war zu keinem Zeitpunkt eine einzelne Kultur unter den Teilnehmenden ein verbindendes Element. Die Kultur ist im Gegenteil genau das abgrenzende Moment.

Der Satz „You can easily access your files from any workstation on the campus“<sup>4</sup> ist eine typische Aussage des Technology Advisor (also des Autors dieser Arbeit) im Gespräch mit Studierenden oder Lehrenden. Bereits an dieser Aussage lassen sich mehrere Annahmen und Kontextbezüge herausarbeiten:

- Der Begriff *any workstation* bezieht sich nur auf die Arbeitsplatzrechner, die für die Studentinnen der *ifu* vorbereitet wurden. Es gab Nachbarräume, in denen ähnliche Rechner standen, die nicht benutzt werden konnten. Deshalb bezieht diese Aussage immer die Grundannahme ein, dass (nur) die ca. 85 vorbereiteten Arbeitsplatzrechner auf dem Campus gemeint sind. Diese waren auf drei unterschiedliche Gebäude und vier Räume verteilt.
- In Abhängigkeit von der angesprochenen Person bedeutet das Zugreifen auf Dateien unterschiedliche Dinge.
- Die Phrase *access your files* unterstellt ein Basiswissen über das von Windows vorgelegte Dateikonzept des hierarchischen Dateisystems und die Nutzung einer grafischen Benutzungsoberfläche. Darüber hinaus wird eine gewisse Routine in der Benutzung des Programms *Explorer* und im Umgang mit der Maus erwartet.

---

4 Da die Universitäts-Sprache der *ifu* Englisch war, wurde ein Großteil der Unterhaltungen in englischer Sprache geführt. Diese sind deshalb hier in Englisch wiedergegeben.

- Die Phrase *your files* unterstellt ein Verständnis des Konzeptes von eigenen Dateien und Dateien anderer. Dieses Konzept wird vom Fileserver durch Rechte, die Dateien und Verzeichnissen zugeordnet sind, implementiert. Die Verzeichnisstruktur unterstützt dies durch eine entsprechende Benennung. Die Freigabe mit den gemeinsamen und privaten Verzeichnissen wurde *home* genannt. Dort standen für jede Person ein persönliches Verzeichnis unterhalb eines mit dem Anmeldenamen bezeichneten Verzeichnisses zur Verfügung. Für jedes Projekt wurden ebenfalls Verzeichnisse mit den Projektnamen angelegt.
- Die Bereitstellung der Dateiablage auf dem Fileserver ist notwendig, um auf die Dateien unabhängig von dem Arbeitsplatzrechner an jedem Ort (*any workstation*) zugreifen zu können. Dazu mussten die Studentinnen einer Absprache folgen und den Fileserver zur Speicherung ihrer Daten verwenden, dessen Freigabe unter dem Laufwerksbuchstaben H zugreifbar war.<sup>5</sup>

Eine andere kulturelle Begegnung fand mit den einzelnen Fachbereichen der kooperierenden Einrichtungen statt. Ein Beispiel hierfür ist das Bezahlen von Ausdrucken bei den Pädagogen (je Seite 10 Pfennige). In der administrativen Zusammenarbeit mit dem Regionalen Rechenzentrum wurden die Administratoren der *ifu* mit dem Terminus technicus „putzen“ für das Neuinstallieren von Arbeitsstationen konfrontiert.

## A.4 Zusammenfassende Betrachtung

Die für die *Internationale Frauenuniversität* entwickelte Zusammenstellung von Gebäuden, Räumen, Rechnern, verbindender Technik, Software und unterstützenden Personen stellt eine Infrastruktur dar (vergleiche hierzu Kapitel 5). Der Zweck dieser Infrastruktur lag in der Unterstützung eines (begrenzten) universitären Projektes. Der Prozess zum Aufbau dieser Infrastruktur gestaltete sich vielschichtig und nutzte teilweise eine bereits bestehende Infrastruktur – nämlich die der Universität Hamburg. Zu Beginn der

---

<sup>5</sup> Windows NT erlaubt es, Fileserver-Verzeichnisse als so genannte Freigaben mit der Arbeitsstation zu verbinden. Dazu wird diese Freigabe wie ein lokales Laufwerk zur Verfügung gestellt. Hierfür wird ihr ein Laufwerksbuchstabe zugewiesen. Es wurde der Buchstabe H gewählt, um das *home*-directory als “‘h’-drive” bezeichnen zu können.

Tabelle 5: Verwendete Software der *ifu*

| Anwendung              | Produkt/Beschreibung  | Version          |
|------------------------|---|------------------|
| Operation System       | Windows NT 4.0, Service Pack 5  | 4.0, SP5 english |
| Office Software        | Microsoft Office 97, Service Release 2 (includes Word, Excel, Powerpoint, not Access) | 97, SR2 english  |
| FTP-Software           | WS FTP light  |                  |
| E-Mail and News client | Netscape Communicator (Messenger)   | 4.7 english      |
| Anonymous FTP-Client   | Netscape Communicator (Navigator)   |                  |
| Web-Browser            | Netscape Communicator (Navigator)   |                  |
| Web-Editor             | Netscape Communicator (Composer)  |                  |
|                        | Frontpage Express   | 2.0 english      |
| Archive Tool           | WinZip  | 6.3 english      |
| PDF-Reader             | Acrobat-Reader  | 4.0 english      |
| Postscript-Reader      | Ghost-View  |                  |
| Movie Player           | Quicktime   | 4.0 english      |
| PublicKeyTool          | PGP   |                  |
| Secure Shell           | SSH   |                  |

Präsenzphase lag eine „rohe“ Infrastruktur vor, die durch die in den Lehre- und Forschungsprojekten entwickelten Aufgaben und andere generierte Forderungen geformt wurde.

#### A.4.1 Rückwirkungen

Durch das Projekt entstandene Rückwirkungen betreffen die Bereiche E-Mail, Benutzerprofile und die Kontenverwaltung. Alle drei Bereiche wirkten von der *ifu* auf den Arbeitsbereich der lokalen Dekanin zurück.

Um die E-Mail an den Standorten „Arbeitsplatz Informatikum“, „Büro Moorweidenstraße“ und „Private Wohnung“ lesen zu können, wurde das E-Mail-Konto der lokalen Dekanin von POP3 auf IMAP umgestellt. Die E-Mails verblieben auf dem Server und wurden nur in Kopie auf den aktuellen Rechner übertragen. Somit waren sie an allen Standorten identisch bearbeitbar und konsistent. Die Rückwirkung bestand darin, dass zunehmend Mitarbeiter des Arbeitsbereichs an der Informatik daraufhin ebenfalls auf diese Form der E-Mail umstiegen, um die Vorteile nutzen zu können.

Die Verwendung von Netscape Roaming Profiles war für die *ifu* notwendig, um den Studierenden das Einloggen an beliebigen Rechnern zu ermöglichen, an denen sie trotzdem ihre persönlichen Browser-Einstellungen vorfanden (vergleiche Entwurfsentscheidung A.2 auf Seite 195). Dieses wurde von den Mitarbeitern des Arbeitsbereichs übernommen, damit sie ihr Adressbuch, die Bookmarks und andere Einstellungen ebenfalls zu Hause und an anderen Orten nutzen konnten.

Der Directory-Server für die Kontenverwaltung wurde im Arbeitsbereich bereits eingesetzt. Durch das *ifu*-Projekt eröffneten sich neue Nutzungsmöglichkeiten, die im Arbeitsbereich für unterschiedliche Anwendungsgebiete genutzt wurden (Abgleich mit NT-Server, Web-Zugriff, Kalender etc.).

#### **A.4.2 Wahrgenommene Problemfelder**

Die im Rahmen des Projektes *Internationale Frauenuniversität* gesammelten Erfahrungen haben eine Vielzahl von Problemfeldern offenbart, die eng mit der Etablierung von Infrastrukturen verbunden sind. An dieser Stelle werden deshalb die oben geschilderten Erfahrungen und Probleme zusammengefasst und daraus Problemfelder gebildet, die für die weitere Arbeit am Infrastruktur-Begriff hilfreich sein werden.

Die Festlegung des Homeverzeichnis auf dem Laufwerksbuchstaben „H“ (Entwurfsentscheidung A.3), die Voreinstellungen des Netscape Communicators (Entwurfsentscheidung A.2) und die Problematik des Internet Explorer Cache (Problem A.14) sind Beispiele für *Konfigurierbarkeit*. Konfigurierbarkeit steht im Kontext von Infrastrukturgestaltung hier für die Möglichkeit, Eigenschaften von Software im Vorwege einzustellen. Während die Konfiguration der Freigabe-Zuordnung mit dem Laufwerksbuchstaben funktionierte und in der Kommunikation mit den Benutzern erfolgreich war, konnte innerhalb der Laufzeit nicht mit vertretbarem Aufwand die Konfiguration der Cache-Einstellungen für alle Anwenderprofile durchgeführt werden.

Die Verwendung der Verzeichnisstruktur, die Probleme bei der Änderung der Cache-Einstellungen des Internet Explorers (Problem A.14), die Unterweisung in die An- und Abmeldeprozedur (Problem A.12), die Anpassung der E-Mail-Voreinstellungen (Problem A.18) und die Verwaltung



mehrerer E-Mail-Konten (Problem A.19) sind Beispiele für *Kommunizierbarkeit*.

Eine besonders schlecht ausgeprägte Eigenschaft einer Infrastruktur in Problemsituationen ist die *Erkennbarkeit* eines Problems. Hierzu können insbesondere der ausgefallene Server (Problem A.1), das unterbrochene Netz (Problem A.2) und die Funktionen der Firewall (Problem A.3) gezählt werden. Beim Ausdruck von Dokumenten im Letter-Format (Problem A.10) und bei den verlorenen E-Mails (Problem A.20) findet man dieselbe Problemkategorie vor.

Neue Infrastrukturen bauen zumeist auf gegebenen, äußeren Rahmenbedingungen auf. Deshalb ist ein Teil der Probleme der neuen Infrastruktur in diesen *Gegebenheiten* begründet. Beispiele dafür sind die langen Wege zwischen den Räumen (Rechnerräume vs. Vorlesungs- und Projekträume; Problem A.4) und die Verteilung der Räume auf sechs Gebäude (Problem A.15). Damit einher geht – begründet durch die Leihgaben der einzelnen Fachbereiche – die Verteilung der Ausstattung auf unterschiedliche Standorte (Problem A.6), die bei einer Neugestaltung sicherlich anders verteilt worden wäre.

Ihre Ursache im Bereich der *Interkulturalität* haben Probleme, die sich z. B. auf das Ausdrucken von Dokumenten beziehen. Das in Deutschland übliche Papierformat DIN A4 wird von den Druckern und der Software problemlos verarbeitet. Werden z. B. im Letter-Format vorformatierte Dokumente aus anderen Ländern mitgebracht, müssen diese angepasst oder die Drucker bei jedem Druckauftrag entsprechend eingestellt werden (vergleiche Problem A.10). Das Tastaturlayout fällt in die gleiche Kategorie. Die vorhandenen Tastaturen hatten neben den Umlauten, der Anordnung der Sonderzeichen über den Zifferntasten, der Vertauschung der Buchstaben Y und Z auch eine andere Beschriftung für „Ctrl“, „Enter“, „Del“ usw. (vergleiche Problem A.7 und Schritt 4 der Abbildung 13 auf Seite 216).

Ein Teil der entstandenen Probleme lässt sich mit fehlender Qualifikation begründen, die nicht vorausgesetzt werden konnte. So gingen E-Mails verloren (vergleiche Problem A.20), weil Studierende ihre E-Mail-Einstellungen dahingehend änderten, dass die E-Mails endgültig vom Server auf die Arbeitsstation verschoben wurden. Das Programm war aber so eingestellt, dass es von Kopien ausging, die am Ende der Sitzung gelöscht werden konnten.

In diesem Kontext sind der Zugang zu Arbeitsstationen am Wochenende (Problem A.22) und der Zugriff auf die Infrastruktur bereits während der Präsenzphase von außerhalb (z. B. aus der Unterkunft) (vergleiche Problem A.23) *unerwartete Anforderungen*, welche sich nur bedingt realisieren ließen. Letzteres wurde erst für die Zeit nach der Präsenzphase berücksichtigt.

Die bereits installierten Versionen des Betriebssystems auf den geliehenen Geräten stellten als *Gegebenheiten* insofern ein Problem dar, als dass sie eine deutsche Version des Betriebssystems besaßen. Die Re-Installation des Betriebssystems auf diesen Rechnern stellte ein logistisches Problem dar (*Übertragbarkeit, Reproduzierbarkeit*), denn es blieben nur wenige Tage, um auf ca. 90 Geräten neue Betriebssysteme und Anwendungen zu installieren und zu konfigurieren. Hierbei geht es also um den Austausch eines Bestandteils innerhalb einer Infrastruktur (Software auf den Rechnern), um dieselbe Infrastruktur für eine andere Nutzergruppe benutzbar zu machen.

#### **A.4.3 Infrastrukturelle Herausforderungen**

Die für die *Internationale Frauenuniversität* aufgebaute Infrastruktur war insofern nutzbar, als dass die geschaffenen Voraussetzungen für die Projektarbeit während der Präsenzphase genügten und danach externen Zugriff ermöglichten. Es hat sich allerdings in der Nutzung deutlich gezeigt, dass der Weg der reinen Antizipation nicht ausreichen kann.

Der erzielte Erfolg und die Nutzbarkeit der entstandenen Infrastruktur waren deshalb nicht zuletzt darauf begründet, dass größtenteils eine bekannte und über Jahre hin entwickelte Infrastruktur (nämlich die des Arbeitsbereichs Softwaretechnik am Fachbereich Informatik) zu großen Teilen reproduziert wurde. Allein die neuen und anderen Rahmenbedingungen (ca. 90 Arbeitsstationen, andere räumliche Gegebenheiten, andere Sprachversionen der Software usw.) reichten für das Auftreten zusätzlicher Probleme und Herausforderungen aus.

Die Herausforderungen betreffen dabei drei Bereiche:

##### *Aufbau der Infrastruktur*

Antizipation der Anforderungen kann nur ein schwacher Ersatz für die zyklische partizipatorische Entwicklung der Infrastruktur sein. Die Heraus-

forderung beim Aufbau einer Infrastruktur besteht deshalb darin, neben der antizipatorischen Vorleistung genügend Raum für Entwicklung und flexible Reaktion zu lassen.

### *Reproduktion*

Durch das Kopieren wesentlicher Element einer gewachsenen Infrastruktur kann ein Anfang für eine neue Infrastruktur gewählt werden. Die Herausforderung besteht hierbei im Nachbau einer bestehenden Infrastruktur und in der Bewertung der zu ändernden Bestandteile.

### *Raum für evolutionäre Entwicklung*

Notwendige Anpassungen und Weiterentwicklungen der Infrastruktur an die Gegebenheiten und Anforderungen müssen im Projekt verankert werden. Eine Herausforderung ist die Entscheidung, an welchen Stellen die Freiheiten explizit ermöglicht werden und wodurch.


| INTERNATIONAL WOMEN'S UNIVERSITY • PROJECT AREA INFORMATION  |                |            |            |          |                |                |            |   |
|--|----------------|------------|------------|----------|----------------|----------------|------------|---|
| <b>Your personal password</b> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 33%; padding: 5px;">Name</th> <th style="width: 33%; padding: 5px;">Login name</th> <th style="width: 33%; padding: 5px;">Password</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">«Full_Name_cn»</td> <td style="padding: 5px;">«Login_ID_uid»</td> <td style="padding: 5px;">«Password»</td> </tr> </tbody> </table>  |                | Name       | Login name | Password | «Full_Name_cn» | «Login_ID_uid» | «Password» |  |
| Name   | Login name     | Password   |            |          |                |                |            |   |
| «Full_Name_cn»   | «Login_ID_uid» | «Password» |            |          |                |                |            |   |
| <p><b>How to login</b></p> <ol style="list-style-type: none"> <li>1. We have three different buildings where you can work with computers. You will find names, locations and opening times of the buildings in your ifu brochure.</li> <li>2. We mainly provide Windows NT 4.0 workstations and some Macintosh Computer. <b>This document only deals with Windows NT.</b> Macintosh is done differently.</li> <li>3. If your selected computer is not switched on, switch it on. There may be a boot loader (a program which lets you choose different operating system). Choose 'Windows NT 4.0' or 'ifu workstation'.</li> <li>4. After workstation boot is complete, press 'Strg'+ 'Alt'+ 'Entf' keys at the same time. (You may be prompted to press 'Ctrl'+ 'Alt'+ 'Del' but you won't find the keys because of different tags on German keyboards. 'Strg'+ 'Alt'+ 'Entf' substitutes for 'Ctrl'+ 'Alt'+ 'Del' on German keyboards).</li> <li>5. You are asked to enter three fields: login name, password, and domain:<br/> Login name: usually is your last name without special characters<br/> Password: only you know it (if you login for the first time use the password written on top of this page).<br/> Domain: IFUPAI</li> <li>6. Windows should start now.</li> <li>7. If you login for the first time, please change your password and choose a password private to you and easy to remember.</li> </ol> <p><b>How to change your password</b></p> <ol style="list-style-type: none"> <li>1. Press 'Strg'+ 'Alt'+ 'Entf' keys at the same time. Choose ,change password'</li> <li>2. Please, follow the tasks in the dialog and remember the password you choose. It's a good idea not to change the password at the end of the week.</li> </ol> <p><b>How to logout</b></p> <ol style="list-style-type: none"> <li>3. Select the ,start' menu</li> <li>4. Choose ,log off username'</li> <li>5. Wait until the login dialog appears</li> <li>6. Finished</li> </ol> <p><b>What to do if you forgot your password</b></p> <ol style="list-style-type: none"> <li>1. Contact the Service Center ()</li> <li>2. Tell them your full name</li> <li>3. You will be given a new password within 24 hours</li> <li>4. You have to come to the Service Center personally to get your new password</li> </ol> |                |            |            |          |                |                |            |   |
| <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <small>WOLF-GIDEON BLEEK • 16.07.2000 12:08 • PAGE 1 OF 1 • VERSION 6</small>   |                |            |            |          |                |                |            |   |

Abbildung 13: Beispiel des Informationsdokumentes, mit dem die Anmeldenamen und Kennworte an die Studentinnen verteilt wurden (dieses Dokument ist die Serienbriefvorlage und enthält keine konkreten personenbezogenen Daten).

## LETTER OF CONSENT

### International Women's University \* Project Area Information CommSy Use and Evaluation

The purpose of this meeting is to discuss your every day project work and your needs for technological support. We show you (additional) features of CommSy and discuss with you how they might be adapted and used to support your project. We want to learn how eventual cooperation within the project beyond *ifu* in hamburg could be supported. We also want to find out if you and how you use the different feature of CommSy. This helps us to determine how to further develop the system in order to match your needs.

The meeting will take about one hour. We plan an open end discussion. You are free to leave at any time. The support will be given to you by a team of two of us familiar with CommSy. We will take notes during the meeting. If you agree, the meeting will be audio taped. The information gathered will be used for scientific work about the use, the adaptation, and the accommodation of CommSy

*At the beginning we will introduce ourselves and explain what we want to do. Then you will have the opportunity to ask us questions about the CommSy functionality. We will use the context of questions to show you parts of the functionality of CommSy. In the second part of the meeting we want to foster a discussion about forms of using CommSy. The session will end with an opportunity for you to let us know about your ideas for future use of the CommSy.*

We will ask you to answer questions about your use of the CommSy in an open way. You need to take part in an open discussion. Your statements might be recorded and transcribed for being used in further scientific work. You might be contacted after the meeting for clarifying further questions.

You will benefit from the meeting by getting to know more about the functionality of CommSy. You will also get the opportunity to influence the future development of CommSy; results of this evaluation will be used for the next versions.

We herewith assure you that any information you provide will be handled confidentially in strict accordance with data protection regulations and will only leave our hands in an aggregated and anonymous form.

We will protect your individuality by storing any audio tapes and notes in a safe place that only the CommSy evaluation team (consisting of Arne Bestmann, Wolf-Gideon Bleek, Yvonne Dittrich, Martti Jeenicke) have access to. No names will appear in the transcripts of the meeting and no real names or any other information making it possible to identify the person will be used in any publication. We will never mention your names in a conversation about the meeting with a third party

Abbildung 14: Letter of Consent für die geführten Interviews<sup>6</sup>

<sup>6</sup> Die zweite Seite wurde für die notwendigen Unterschriften verwendet.



## - B -

# Empirieprojekt *hamburg.de*

Seit Juni 1996 betreibt die Stadt Hamburg die Website *hamburg.de*. Auf dieser wurden bis dato hauptsächlich allgemeine Informationen über die Stadt und ihre Einrichtungen präsentiert. Einige wenige Anwendungen (wie z. B. DiBIS; vergleiche DiB 2000; Senat der Freien und Hansestadt Hamburg 2000a, S. 6; Senat der Freien und Hansestadt Hamburg 2000b, S. 4) erlaubten es, interaktive Anfragen zu formulieren. Die Situation der Internet-Präsenz wurde im Jahr 1999 vom Senat wie folgt eingeschätzt (Antwort des Senats auf eine „schriftliche kleine Anfrage“):

„Unter *hamburg.de* werden seit Juni 1996 Informationsdienstleistungen im Internet angeboten. Auf der Grundlage der Bedeutung dieser Präsentationsplattform für die öffentliche Hand haben in den vergangenen beiden Jahren überwiegend Behörden und öffentliche Einrichtungen Informations- und Serviceangebote im Internet bereitgestellt. Der *Direkte Bürger Informations-Service* (DiBIS) wurde Bestandteil dieses Internetangebots. Nach einer Betriebszeit von zwei Jahren haben die Träger von *hamburg.de* (Staatliche Pressestelle, Wirtschaftsbehörde und Finanzbehörde) eine Bestandsaufnahme des bisherigen Angebots durchgeführt, um der wachsenden wirtschaftlichen und gesellschaftlichen Bedeutung des Internetauftritts durch eine inhaltliche, organisatorische und technische Weiterentwicklung Rechnung tragen zu können.“

(Kähler 1999)

Um im Wettbewerb der Städte untereinander eine führende Position einzunehmen und um den aktuellen Anforderungen gerecht zu werden, beschloss der Hamburger Senat, den Internet-Auftritt der Stadt Hamburg an ein privates Unternehmen abzugeben (Zuckerer et al. 1999, Kähler 1999, Kähler 2001), um dadurch schneller den wechselnden Bedürfnissen gerecht zu werden und gleichzeitig ein größeres Spektrum von Basisangeboten zur Verfügung zu stellen.

Dazu wurde im Sommer 1999 eine Ausschreibung durchgeführt (Senat der Freien und Hansestadt Hamburg 1999a, Senat der Freien und Hansestadt Hamburg 1999b), an der sich insgesamt 18 Unternehmen beteiligten, von denen 12 zur Abgabe eines Angebotes aufgefordert wurden (Freytag und Ehlers 1999, S. 1), unter anderem ein Konsortium mit Beteiligung der

Universität Hamburg: ASSIST New Media, MAZ Harburg, HITeC e.V. und die Firma Dr. Neuhaus. Den Zuschlag erhielt im Januar 2000 das Konsortium der norddeutschen Sparkassen, vertreten durch S Online. Mit ihnen wurde die *hamburg.de* GmbH und Co. KG gegründet, die seitdem die Verantwortung für den Betrieb besitzt.<sup>1</sup> Sie hat den Technologietransfer-Verein des Fachbereichs Informatik der Universität Hamburg, HITeC e.V., gegeben, bei der Entwicklung des Projektes beratend zur Seite zu stehen und die Umsetzung des Lebenslagen-Konzeptes (Bleek und Floyd 2000, Bleek und Floyd 2001, Bleek 2001) sicherzustellen. Diese Aufgabe wurde seitdem von mir übernommen.

Im Rahmen dieses Projektes ist eine aufwändige technische Plattform erstellt worden, die u. a. für ca. 120.000 E-Mail-Adressen, 3.500 Homepages und bis zu 140.000 Informationseinheiten die technische Basis bereitstellt (Stand 10. Oktober 2001). Darüber hinaus werden neben der Website *hamburg.de* noch die Portale von anderen Städten betrieben.<sup>2</sup> Es handelt sich mittlerweile um eine mandantenfähige Plattform (s. u.), die mit geringem Aufwand an weitere Kunden vermietet werden kann.

Bei dieser technischen Plattform kann eine Vielzahl von Phänomenen untersucht werden, die charakteristisch für diese Art von Projekten sind:

- Aus welchen Basisbestandteilen wird die Plattform zusammengesetzt?
- Wie verläuft der Entwicklungsprozess?
- Welche Schwierigkeiten treten auf und welche Lösungsansätze werden aufgrund welcher Kriterien gewählt?
- Welche Abhängigkeiten gibt es?

Ziel dieses Kapitels ist es, den Prozess der Entwicklung zu dokumentieren. Anhand dessen können Prozesscharakteristika benannt werden; es kön-

---

1 „hamburg.de ist ein Gemeinschaftsunternehmen von: Freie und Hansestadt Hamburg, S NetLine, Hamburger Sparkasse, Hamburgische Landesbank, Sparkasse Harburg-Buxtehude.“ <http://www.hamburg.de/info/1,3004,impressum,00.html>, Stand 11. Juli 2001. Im Bericht des Haushaltsausschusses der Hamburger Bürgerschaft (Freytag und Ehlers 1999) wird die prozentuale Verteilung angegeben als: 30,1%, 20%, 25,1%, 20%, 4,8%.

2 [www.schleswig-holstein.de](http://www.schleswig-holstein.de), [wir.schleswig-holstein.de](http://wir.schleswig-holstein.de), [www.neu-wulmstorf.de](http://www.neu-wulmstorf.de), [www.seevetal.de](http://www.seevetal.de), [www.salzhausen.de](http://www.salzhausen.de), [www.landkreis-harburg.de](http://www.landkreis-harburg.de), [www.winsen.de](http://www.winsen.de), [www.tostedt.de](http://www.tostedt.de) etc.



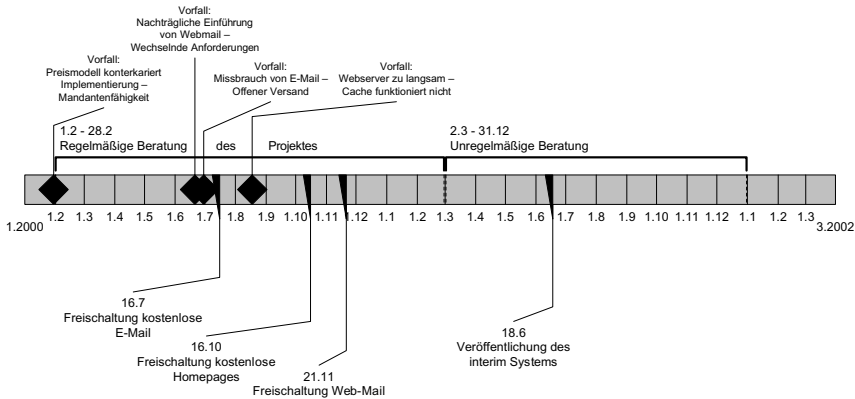


Abbildung 15: Der Zeitverlauf des *hamburg.de*-Projektes

nen Komplexitäten, die durch zusätzliche Dimensionen hinzukommen, beschrieben und es kann die Rolle von Skalierbarkeit untersucht werden. Von besonderem Interesse sind hierbei die Umstände, unter denen Projekte dieser Art des Aufbaus von technischen Plattformen scheitern können. Die Abbildung 15 zeigt den zeitlichen Ablauf des Projektes mit wesentlichen Ereignissen. Das Projekt selbst läuft über diesen Zeitraum hinaus, wird hier aber nur bis Ende 2001 betrachtet.

In der Senatsbilanz vom 20. April 2001 konnte man unter der Überschrift *Ausbau des Informationsangebots im Internet* nun Folgendes lesen:

„Das Internetangebot der Freien und Hansestadt Hamburg unter [www.hamburg.de](http://www.hamburg.de) bietet unter einem Dach Informationen der Verwaltung, Electronic-Commerce sowie Inhalte und Dienstleistungen aus allen Lebensbereichen. Das regionale Informations- und Kommunikationsportal wird als Public-Private-Partnership von einer privatwirtschaftlich organisierten Gesellschaft betrieben. [www.hamburg.de](http://www.hamburg.de) bietet allen Hamburgerinnen und Hamburgern einen kostenlosen Internetzugang, eine kostenlose E-Mail-Adresse und eine kostenlose Homepage einschließlich eines Tools für ihre Erstellung. Das Leistungsangebot wird ständig erweitert. Ende 2000 besuchten über 2,5 Millionen Internetnutzer mit über 12,6 Millionen Seitenaufrufen das Angebot von [hamburg.de](http://hamburg.de).“

(Pressestelle 2001)

Die empirische Basis für dieses Projekt bilden die Protokolle der regelmäßigen Projektsitzungen, der E-Mail-Verkehr zwischen den Beteiligten und

Gesprächsnotizen aus Workshops und Telefonaten. Eine ausführliche Aufstellung des vorliegenden Materials wird in Kapitel 4.1.1 gegeben.

Dieses Kapitel gliedert sich in vier Abschnitte. Der erste wird die organisatorischen Rahmenbedingungen skizzieren, die bei Projekten mit öffentlichen Trägern und einer Vielzahl von beteiligten Interessenvertretern naturgemäß recht facettenreich sind. Im zweiten Abschnitt werden die während des Projektes entwickelten Anforderungen dokumentiert. Der dritte Teil liefert einen Überblick über die verwendeten Software-Bestandteile und die damit verknüpften Randbedingungen. Abschließend wird der vierte Abschnitt die Ergebnisse zusammenfassen.

## **B.1 Organisatorische Rahmenbedingungen**

Die Anforderungen an die Architektur der zu errichtenden technischen Plattform ergeben sich aus dem Vorhandensein einer Vielzahl von beteiligten Akteuren. Eine wichtige Rolle nimmt dabei die Freie und Hansestadt Hamburg (FHH), vertreten durch den Senat, ein, indem sie politische Forderungen an den Internet-Auftritt knüpft. Diese politischen Forderungen sind teilweise eine Reaktion auf Anfragen aus der Bürgerschaft. Aus einer Anfrage an den Senat:

- „1. Wie lautet der Ausschreibungstext?
2. Wird Hamburg die prominente Webadresse ‚hamburg.de‘ kostenlos überlassen?
3. Wird das Land für die Gestaltung seiner Homepage Geld bezahlen?
4. Wird die Finanzierung des Angebotes durch das gewählte Unternehmen durch ‚ECommerce‘ und Werbung erfolgen?
5. Wird es eine Verknüpfung von Dienstleistungen an bestimmte Unternehmen etwa in dem Sinne geben, daß wer ‚online‘ einen neuen Sellschein ordern will, dafür ein Online-Konto bei der Haspa braucht?
6. Werden die Hamburger/innen die Möglichkeit haben, im Internet-Angebot der Stadt ihre persönliche E-Mail-Adresse kostenlos einzutragen?
7. Werden die Hamburger/innen die Möglichkeit haben, eine private ‚Homepage‘ kostenlos einzurichten?
8. Wird ein Bürgerforum, werden Diskussionsforen und ‚Chaträume‘ eingerichtet, wo auch politisch diskutiert werden kann?

9. Soll das gefundene Unternehmen ebenso an dem digitalen Signaturverfahren, an dem unter anderem auch debis und die Bundesdruckerei arbeiten?
10. Ist Hamburg bestrebt, eine allgemeingültige digitale Signatur für alle Verwaltungsvorgänge nutzen zu können?
11. Wird es im Stadtgebiet multifunktionale Informationssäulen geben?“

(Kähler 1999, S. 2)

Im Bericht des Haushaltsausschusses der Hamburgischen Bürgerschaft werden die Rahmenbedingungen für *hamburg.de* wie folgt festgelegt:

„hamburg.de werde auf drei Säulen stehen:

- Für den Bereich des Themenfeldes *hamburg.de* und die Beziehung zu den Bürgerinnen und Bürgern der Stadt (Bürgersäule):
  - Jede Bürgerin und jeder Bürger solle eine E-Mail-Adresse mit ihrem/seinem Namen erhalten können.
  - Der Internet-Zugang solle kostenlos sein.
  - Non-Profit- und kostenpflichtige Angebote seien überprüfbar abzugrenzen.
  - Der Betreiber solle sich nicht um die Inhalte kümmern.
  - Es müßten sich Foren organisieren können.
- Für den Bereich des kommerziellen Themenfeldes (kommerzielle Säule):
  - Die Anbieter erarbeiteten ein sogenanntes Lebenslagenkonzept; definierte Lebenssituationen gewährleisteten den zielgerichteten Zugang zu *hamburg.de*, der für die Betreiber Ansätze für kommerzielle Nutzungsmöglichkeiten eröffne.
- Für den Bereich des Themenfeldes öffentliche Verwaltung (Verwaltungssäule):
  - Inhalte würden verantwortlich von der Stadt geliefert und von den Betreibern redaktionell aufbereitet.
  - *hamburg.de* böte die Möglichkeit, gemeinsam mit den Kunden der öffentlichen Verwaltung zu Geschäftsprozessen zu kommen.
- Dem Konzept liege inhaltlich der sogenannte virtuelle Marktplatz zugrunde, der das nach Lebenslagen strukturierte Drei-Säulen-Modell umfasse. Die Bezahlssysteme würden das Modell ergänzen, sobald Sicherheit für Bestell- und Bezahlvorgänge durch die digitale Signatur geschaffen worden sei.
- Es werde einen kostenlosen Zugang zum Informationssystem geben. Der jeweilige Anbieter zahle umsatzorientierte Transaktionsgebühren. Eine

wesentliche Zielgruppe möglicher Anbieter seien die klein- und mittelständischen Betriebe (mit bis zu ca. 500 Mitarbeiterinnen und Mitarbeitern), für die ein großer Nachholbedarf zur Eigenpräsentation im Internet erwartet werde. Den Bürgerinnen und Bürgern werde die alltägliche Umwelt der nachbarschaftlichen Region mit dem Zugang zu wirtschaftlichen und kulturellen Aktivitäten erschlossen. Der Betreiber sei letztlich Dienstleister, der dafür Sorge, daß die Basistechnik nach bestimmten Regeln verfügbar sei. Er werde die Kunden verwalten und betreuen sowie den Zahlungsverkehr abwickeln.

- Besondere Akzente würden durch innovative Projekte in der digitalen Kommunikation zwischen Bürgerinnen und Bürgern sowie der Verwaltung gesetzt werden. Eine anteilige Finanzierung durch den Betreiber werde auf diese Weise externes Geld erschließen, um Voraussetzungen z. B. für die digitale Signatur und den elektronischen Geldfluß zwischen Bürgerinnen und Bürgern einerseits und Verwaltung andererseits aufzubauen. Die Basisleistungen dieser Plattform seien für die Stadt kostenfrei, Rechnerleistungen und redaktionelle Aufbereitung trage der Betreiber.
- Mit dem Aufbau der Marke *hamburg.de* am Markt werde eine Vertriebsstruktur zu entwickeln sein, in deren Rahmen der Vertriebspartner mit *hamburg.de* eine Plattform für seine Angebote und einen neuen Vertriebsweg finde. Den Zugang könne eine Telefongesellschaft wie HanseNet vermitteln; Alternativen blieben offen.
- *hamburg.de* plane vielfältige technische Zugangswege, die sich an den Anforderungen bestimmter Zielgruppen orientieren würden. So sei das Konzept neben den üblichen stationären Zugängen in Büro und Haushalt auch offen für den Zugang über WAP-Phones (Wireless Application Protocol) oder Kiosk-Systeme. Für das Informationssystem *hamburg.de* werde der Betreiber wenig eigene Technik aufbauen.
- Wesentliche zeitliche Eckpunkte der technischen Umsetzung seien die Migration der jetzigen Angebote in die neue Umgebung von *hamburg.de* sowie erste qualitative Fortentwicklungen Mitte 2000. Von einer detaillierten Arbeitsplanübersicht werde im Vertragstext abgesehen. Statt dessen solle eine Arbeitsorganisation geschaffen werden, die schnell mit konkreter Arbeit anfangen könne.
- Eine Vertragslaufzeit von zehn Jahren solle dem Vertragspartner eine ausreichende ökonomische Sicherheit gewährleisten, die ihm den Handlungsspielraum für die notwendigen, von ihm zu tragenden Investitionen eröffne.“

(Freytag und Ehlers 1999, S. 1-2)

Die städtische Website *hamburg.de* wird seit der Ausschreibung von der Gesellschaft *hamburg.de GmbH & Co KG* betrieben. Dieses Unternehmen ist eine Tochter der Hansestadt Hamburg zusammen mit den Sparkassen (s. o.). Diese wiederum hat die Firma *S NetLine* beauftragt, den technischen und wirtschaftlichen Betrieb sicherzustellen. Dies umfasst die Bereitstellung von Technik für Dienstleistungen, den Kauf von Inhalten sowie den technischen Betrieb der Hard- und Software. Auf der wirtschaftlichen Seite kommen Produktentwicklung und Vertriebsorganisation dazu.

Im Rahmen der vertraglichen Vereinbarungen wurden Dienstleistungen definiert, die bei Start des Projektes verfügbar sein sollten. Dazu zählen kostenlose E-Mail-Adressen, ein Zugriff auf E-Mail über eine Web-Schnittstelle („Web-Mail“), private Internetseiten („Homepages“) und ein Redaktionssystem zur Pflege der von der Stadt publizierten Informationen.

Die Firma *S NetLine* verfolgt darüber hinaus die Absicht, dieselbe Portalentwicklung für andere Kommunen, Länder und Städte bereitzustellen, um die finanziellen Investitionen zu rechtfertigen. Dies wird unter dem Stichwort *Mandantenfähigkeit* oder Mehrlagerfähigkeit subsummiert. Auf diese Anforderung, ihre Umsetzung und die Folgen der Forderung auf das Gesamtsystem werde ich im Folgenden immer wieder zurückkommen.

Unter *Mandantenfähigkeit* versteht man die Fähigkeit einer Software, eine Dienstleistung in unterschiedlichen Kontexten zur Verfügung zu stellen, ohne dass sie erneut (gegebenenfalls auf separater Hardware) konfiguriert werden muss (Stahlknecht 1990). Eine wesentliche Eigenschaft der *Mandantenfähigkeit* ist deshalb die Ausnutzung von bestehenden Ressourcen für verschiedene Auftraggeber (Mandanten). Dafür ist es allerdings unabdingbar, dass die Software eine Unterscheidung der Mandanten zulässt.

Es ist wichtig zu bemerken, dass die Anforderung der *Mandantenfähigkeit* nur deshalb in das Projekt aufgenommen wurde, weil die betreibende Gesellschaft mit derselben Installation mehrere Kunden (mit dem Ziel der Kostenersparnis im Bereich Entwicklung, Pflege und Hardware) bedienen möchte. Im Kontext des Projektes mit der Stadt Hamburg war diese Zusatzanforderung nicht notwendig.

*S NetLine* besitzt das Konzept für ein Portal, die Vertriebsstruktur und den wirtschaftlichen Hintergrund und konzentriert sich damit auf die eigenen Kernkompetenzen. Die Entwicklungsarbeit wird an ein Unternehmen abgegeben, das Erfahrungen in der Entwicklung großer Softwareprojekte besitzt und die notwendigen Voraussetzungen auf der wirtschaftlichen Sei-

te mitbringt. S NetLine hat sich für das *debis Systemhaus* (jetzt T Systems) Hamburg entschieden.

Zum Betrieb der für das Projekt notwendigen Hardware wurde ein Tochterunternehmen der Sparkassen, die *nbg* (Netzbetriebsgesellschaft), ausgewählt. Zur Beratung standen von der Firma Vignette Corporation, Niederlassung Hamburg, und von HITeC e.V. jeweils eine Person zur Verfügung.

## B.2 Anforderungen an das Portal

Das Portal gliedert sich in einzelne (relativ unabhängige) Teile, die allerdings zusammenarbeiten können. Die E-Mail und das Web-Mail-Angebot müssen zwar unabhängig vom Rest des Systems laufen, die Anmeldung und das damit verbundene Konto sollen aber auf dem Portal für die Personalisierung weiterverwendet werden können. Gleiches gilt für die privaten und gewerblichen Homepages, die ebenfalls eine Anmeldung benötigen. Sowohl aus der Sicht der Softwareentwickler als auch aus der Benutzersicht wäre es unerwünscht, wenn hier unterschiedliche Anmeldungen verwendet werden würden.

Eine saloppe Definition von „Portal“ findet sich bei Zschau, Traub und Zahradka:

„Unter einer *Portal*-Website versteht man eine Website, die ein ‚Eingangstor zum Internet‘, einen ersten Anlaufpunkt für das Surfen im World Wide Web, darstellen will. Was die Homepage für eine Website ist, wollen Portale für das Internet sein. Allerdings zeigen neuere Untersuchungen, dass die wenigsten allgemeinen Portale profitabel arbeiten. Der Trend geht daher zu B2B- sowie themenbezogenen Portalen. Im Online-Marketingmix stellt die Kooperation mit einer Portalsite für einen Webanbieter eine großartige Möglichkeit dar, seine Seiten einem großen Publikum zu präsentieren. Allerdings ist die Chance, eine Partnerschaft mit einem Portal einzugehen, recht gering und in der Regel mit hohen Kosten verbunden.“

(Zschau et al. 2002, S. 414)

Unter einem *Portal* versteht man (Detlor 2000, Koenemann et al. 2000) eine unter eine URL integrierte Zusammenfassung von thematisch spezialisierten Diensten, die mit für das jeweilige Themengebiet erhöhter Informationsqualität und speziellen Dienstangeboten die so genannten vertikalen Portale schaffen (Schumacher und Schwickert 1999, Morrisa et al. 1999). Als erster Vertreter wird im Allgemeinen Yahoo! (Kim 2000) genannt.

Ein Web-Portal ist eine Website im WorldWideWeb, die Informationen aus verschiedenen ausgewählten Quellen zusammenfasst und ihren Benutzern über einen Standard-Web-Browser einen (personalisierten) Zugang mittels Suche und/oder Navigation von Verzeichnisstrukturen bietet, gegebenenfalls ergänzt um redaktionellen Inhalt, Funktionalität zur Kommunikation und/oder Informationsverarbeitung (Koenemann et al. 2000).

Wesentliches Charakteristikum des Portals ist, dass es den gemeinsamen Zugang zu einer Fülle von weiteren Informationen unter einem Dach (siehe Gebäude) bildet (single point of access). An den Einstieg schließt sich natürlich meist reichhaltige Funktionalität an. Grundsätzlich beschreibt der Begriff des Portals also nach außen hin für Benutzer sichtbare Informationen und Funktionalität (Wegner 2002, S. 52).

Als Personalisierung bezeichnet man sowohl die Möglichkeit als auch den Vorgang bei der Anpassung eines Portals an die (dem Benutzer) eigenen Bedürfnisse. Die besondere Herausforderung bei Portalen und Web-Applikationen liegt darin begründet, dass eine große Menge persönlicher Einstellungen gespeichert und zugeordnet werden muss und gleichzeitig die Erkennung des Anwenders möglichst bequem geregelt werden sollte.

Die Errichtung der technischen Plattform wurde auf Standardprodukte abgestützt (Stahlknecht 1990, Rolf 1998), die bereits am Markt verfügbar sind. Prämisse war es, diese Produkte zwar anzupassen, aber keine gleichwertigen Eigenentwicklungen durchzuführen. Bei der Auswahl der Software gab es keine generelle Einschränkung, ob es sich um kommerzielle oder frei verfügbare Produkte handelt.

Der Begriff „Standardprodukt“ kann in diesem Zusammenhang in mindestens zwei Granularitäten diskutiert werden. Auf der einen Seite sind Datenbankserver, Webserver usw. Standardprodukte, die von verschiedenen Herstellern angeboten werden. Diese Gruppe von Produkten liefert keine oder nur beschränkte Applikationslogik und kann allenfalls als Basisbestandteil verwendet werden. Darüber hinaus gibt es eine Gruppe von Standardprodukten, die ganze Dienstleistungen und komplette Services vereinen. Hierzu zählen zum Beispiel so genannte „Freemailer“ (vergleiche gmx, web.de, hotmail etc.) oder Unified Communication Server (vergleiche Canbox [www.canbox.com](http://www.canbox.com), [smartvia.de](http://smartvia.de)). Sowohl die UCS-Dienstleister als auch die Freemailer bieten eine Kombination aus E-Mail-Dienst und verschiedenen Zugriffskanälen (POP3, Web etc.) an. Im Bereich der UCS kommen noch ergänzende Dienstleistungen (Fax, Voicebox etc.) hinzu. Bei-

Tabelle 6: Ausgewählte Standardprodukte für die Errichtung des Webportals

| Anwendung/Kategorie             | Produkt/Firma               | Typ              |
|---------------------------------|-----------------------------|------------------|
| Content-Management-System       | Vignette Story-Server 5.0.1 | kommerziell      |
| Datenbank                       | Oracle 8.1.7i               | kommerziell      |
| Webserver                       | Apache 1.3.19               | frei             |
| E-Mail-Server                   | Qmail                       | frei             |
| E-Mail-Server                   | Courier IMAP, POP3          | frei             |
| Directory-Server                | OpenLDAP                    | frei             |
| Applikations-Server             | Bea Weblogic                | kommerziell      |
| FTP-Server                      | wuFTP                       | frei/kommerziell |
| Schlagwortkatalog               | bremen.de <sup>3</sup>      | kommerziell      |
| Suchfunktionalität              | Intermedia                  | inklusiv         |
| Werbe-Banner-Server             | DoubleClick                 | kommerziell      |
| Shopsystem                      | telemall                    | kommerziell      |
| Page-View-Ermittlungssoftware   | IVW <sup>4</sup>            | kommerziell      |
| Web-Mail <sup>5</sup>           | Canbox, Amis                | kommerziell      |
| Community Software <sup>6</sup> | caput <sup>7</sup>          | kommerziell      |

de Gruppen bieten an, auf ihrer Hardware (oder fremder) entsprechende Dienstleistungen zu installieren, die in Optik und Funktionsumfang an die Bedürfnisse des Kunden angepasst werden. Ein auf den Fall der Nutzung abgestimmtes Lizenzmodell regelt die Kosten.

Die als Standardprodukte für die technische Plattform ausgewählten Produkte (Software) finden sich in Tabelle 6. Darüber hinaus wurde über Ergänzungen diskutiert, die in Tabelle 7 auf der nächsten Seite aufgeführt sind. Anhand dieser Auswahl können unterschiedliche Fragen diskutiert

3 Der Zuschnitt eines Schlagwortkataloges ist so speziell, dass es hier keinen wirklichen Markt für Produkte gibt, sondern eher zufällige Kontakte zu dieser Auswahl geführt haben.

4 Dieses Produkt ist in Deutschland von der Vereinigung der Werbewirtschaft vorgeschrieben, um für Abrechnungen verwendbare Zahlen zu generieren.

5 Das Produkt wurde durch eine Eigenentwicklung ersetzt, um das gewählte Content-Management-System verwenden zu können.

6 Diskussionsforen, Chat etc.

7 Das Produkt ist nicht zum Einsatz gekommen.



Tabelle 7: Ergänzungsprodukte für das Webportal

| Kategorie                | Produkt/Firma | Typ         |
|--------------------------|---------------|-------------|
| Kalender                 | DayByDay      | kommerziell |
| Geo-Informationen-System |               | kommerziell |
| Indexierung              | Autonomy      | kommerziell |
| Avatare                  |               | kommerziell |

werden. Aus infrastruktureller Sicht ergibt sich z. B. als ein wichtiges Problem, wie man zu diesen Produktkategorien kommt und dann die entsprechenden Produkte auswählt.

#### *Entwurfsentscheidung B.1 (Content-Management-System)*

Von Beginn an wurde das Projekt unter der Vorgabe durchgeführt, das Content-Management-System Vignette zu verwenden.

#### *Entwurfsentscheidung B.2 (Datenbank)*

Als Datenbank wurde Oracle eingesetzt. Diese Entscheidung stand für die Steuerungsgruppe ebenfalls nicht zur Debatte.

### **B.2.1 Bestimmung des minimalen Kernsystems**

Das minimale Kernsystem für die *hamburg.de*-Plattform sollte zur Erstellung dynamischer Webseiten aufgrund von verschiedener Datenquellen geeignet sein. Inhaltsverzeichnisse, Übersichten und Präsentationen von Daten und Dokumenten sollten unter verschiedenen Blickwinkeln im Web angeboten werden können.

„Ein Kernsystem ist [...] ein zu definierender Ausschnitt aus dem integrierten Softwaresystem. Es bildet eine Art Klammer, um Spezialsysteme anzubinden und den Informationsaustausch zwischen den Teilsystemen zu ermöglichen.“  
(Floyd et al. 1997, S. 17-18)

„Die fachliche Bestimmung des Kernsystems erfolgt im Einzelnen nach den folgenden Merkmalen:

- Mit dem Kernsystem werden in erster Linie eng verbundene, wichtige Bereiche der Organisation unterstützt.

- Im Kernsystem sind diejenigen Aufgaben zu unterstützen, die einen wesentlichen Beitrag zur Gesamtaufgabe der Organisation leisten, die akute Anforderungen (z. B. gemäß gesetzlicher und betriebswirtschaftlicher Bestimmungen) umsetzen oder durch häufige Erledigung gekennzeichnet sind.
- Das Kernsystem soll durch die Bereitstellung von gemeinsam genutzten wesentlichen Daten einen grundlegenden Kooperationservice bieten und darüber hinaus einen einheitlichen Satz von BasisKooperationsdiensten zur Verfügung stellen.
- Das Kernsystem muß so angelegt sein, daß die notwendigen Spezialsysteme daran angebunden werden können.

Nach der inhaltlichen Festlegung des Kernsystems wird dies in einem Dokument niedergelegt.“ (Krabbel et al. 1997, S. 4-5)

Die Intention, Standardprodukte aus den beiden o. g. Ebenen für die Systemgestaltung zu verwenden, widersprach der Erkenntnis, dass eine Vielzahl von Anforderungen artikuliert wurde, die mit Standardprodukten nicht zu erreichen waren (vergleiche Tabelle 8 auf der nächsten Seite).

Die Verwendung eines Content-Management-Systems stand außer Frage, da es sich gut eignet, eine große Menge von Daten in unterschiedlichem Layout zu präsentieren. Hierfür ist ein Web-Content-Management-System (WCMS) prädestiniert (Büchner et al. 2001, Zschau et al. 2002, Nakano 2001). Zur Speicherung der genannten Daten ist eine Datenbank notwendig. Dies muss nun im Zusammenhang mit dem WCMS betrachtet werden, da zwischen diesen beiden Bestandteilen der zukünftigen Plattform eine hohe Bindung besteht (das WCMS reagiert auf neu eingespielte Daten ebenso wie auf die Entfernung bestehender). Die Datenbank liefert alle anzuzeigenden Inhalte zur Formatierung aus. Das gleiche Kriterium gilt für die Auswahl des Webservers. Hier ist ebenfalls auf die gute Zusammenarbeit mit dem WCMS zu achten. Damit ist das minimale Kernsystem konfiguriert.

Die Wahl der Hardware-Plattform und des Betriebssystems wurde im Projekt relativ spät behandelt (Beleg Protokoll 21-01-2000). Wiederholt waren verschiedene Einflussfaktoren für die Entscheidung verantwortlich. Zum einen erlaubte das WCMS nur noch eine beschränkte Auswahl (Sun-SPARC mit Solaris, IntelPC mit Windows NT), zum anderen wurde aber nach einer in anderen Projekten bewährten und intensiv unterstützten Hardware-Plattform (und einem ebensolchen Betriebssystem) gesucht.

Tabelle 8: Probleme beim Einsatz eines E-Mail-Standardproduktes

| Produkt            | Randbedingung                           | Vorteil  | Nachteil  |
|--------------------|---|--|---|
| Freemailer         | Betrieb auf Hardware des Anbieters      | Keine Hardware-Einrichtung,<br>-Reparatur,<br>-Wartung; Ausbau erfolgt automatisch | Daten und Prozesse liegen beim Anbieter; schwere Integration der Anwendung in die eigenen Applikationen |
| Mail-Applikationen | Eigene Installation und eigener Betrieb | Hoheit über die Daten  | Sonderanforderungen nur schwer umsetzbar  |

### *Entwurfsentscheidung B.3 (Betriebssystemwahl)*

Als Betriebssystem für die Plattform wurde Unix gewählt. Diese Entscheidung fußt auf der Empfehlung des Content-Management-System-Herstellers und den Erfahrungsberichten anderer Betreiber von kostenfreien E-Mail-Servern.

## **B.2.2 Bestandteil Anmeldeprozess**

Der Anmeldeprozess erfüllt drei Funktionen. Zuerst wird für den Kunden ein Konto eingerichtet, das mit einem Anmeldenamen versehen ist. Dieser Anmeldenamen ist für die spätere Arbeit am Portal vorgesehen (Anmeldung in geschützten Bereichen). Die zweite Funktion ist die Vergabe einer kostenlosen E-Mail-Adresse.

### *Entwurfsentscheidung B.4 (Mehrsprachigkeit)*

Das neue Portal und alle seine Bestandteile sollten mehrsprachig sein, um nach außen hin zu dokumentieren, dass Hamburg eine weltoffene Stadt ist. Das alte Portal unterstützte bereits die Sprachen Deutsch und Englisch, das neue Portal sollte darüber hinaus für weitere Sprachen offen sein (z. B. Russisch, Türkisch, Dänisch etc.; vergleiche Beleg Protokoll 21-01-2000).

Zur Vorbereitung für die Vergabe von freien E-Mail-Adressen und kostenlosen Homepages wurde ein mehrsprachiger Anmeldeprozess benötigt, bei dem sich Benutzer nach dem Eintragen persönlicher Daten ihre E-Mail-Adresse in gegebenen Grenzen aussuchen können. Hierfür wird bereits die

Unterstützung einer Applikationslogik benötigt, da als Erstes die Überprüfung der E-Mail-Adresse auf Gültigkeit (im Rahmen der gesetzten Regeln), als Zweites die Überprüfung, ob diese noch frei ist, und als Drittes die dann notwendige Einrichtung durchgeführt werden müssen.

Das WCMS bietet von Haus aus einen Tcl-Interpreter für Applikationslogik an, der allerdings nur für den Seitengenerator zur Verfügung steht. Zentrale Prozesse müssen von einem Applikations-Server bedient werden.

#### *Entwurfsentscheidung B.5 (Applikationslogik in Java)*

Zu Beginn des Projektes wurde entschieden, dass die gesamte Applikationslogik des Systems in der Programmiersprache Java implementiert wird.

Damit wurden die Schablonen des WCMS lediglich zum Zusammenbauen der Seiten, bestehend aus festem und variablem HTML-Code, verwendet. Es existiert ein unerwünschter Bruch zwischen der Programmiersprache des WCMS für die Schablonen (hier Tcl) und der des Applikationsservers für die Applikationslogik (hier Java). Eine einheitliche Nutzung von Java wurde erst in einer späteren Version des WCMS hergestellt, die im Projekt nicht zum Einsatz gekommen ist.

Der Ablauf der Applikationslogik erfordert einen Applikations-Server, der die gleichzeitig laufenden Programme verwaltet, ihnen Ressourcen zuordnet und sie mit den entsprechenden Webserver-Prozessen in Verbindung bringt. Er liefert dafür eine skalierbare Architektur und ein Session-Management.

Session-Management erweitert den verbindungslosen Dienst, der über HTTP (Fielding et al. 1997) realisiert wird, zu einem verbindungsorientierten Dienst. Sollen Informationen von einer Seitenanforderung zur nächsten übernommen werden, bietet Session-Management eine Möglichkeit, beliebige Daten an eine serverseitige Session zu assoziieren.

Die Session wird durch eine Zeichenkette (meist Zahlen-Buchstaben-Kombination), genannt Session-ID, repräsentiert und kann entweder in die URLs kodiert oder durch einen Cookie übertragen werden (siehe Abbildung B.2.2).

#### *Session-Management*

Sessions werden verwendet, um dem verbindungslosen Webdienst (über das verbindungslose Protokoll HTTP) eine virtuelle Verbindung (Session) zu geben.

Eine Session wird durch einen Identifikator realisiert, der bei der ersten Seitenanfrage erzeugt und dann von Seitenanfrage zu Seitenanfrage übertragen wird. Der Server pflegt damit die „Geschichte“ einer Verbindung und ordnet jede neue Seitenanfrage dieser wieder zu. Der Identifikator ist meist eine (lange) Zeichenfolge aus Buchstaben und Ziffern. Auf der Server-Seite werden mithilfe dieses Identifikators Daten assoziiert.

Konkret kann ein Session-Identifikator entweder über einen Session-Cookie (auf der Seite des Anwenders) oder über einen URL-Parameter (auf der Seite des Servers) realisiert werden. Bei der Verwendung eines Session-Cookies ist die Implementierung gegenüber dem Anwender, falls keine entsprechende Warnung aktiviert ist, nicht erkennbar. Wird der URL-Parameter benutzt, ist jede im Adressfeld angezeigte URL und jeder Link um die Session-ID erweitert (s. u.). Diesen Vorgang nennt man URL-Rewriting (Engelschall 2001).

Die Sicherheit der Verbindung ist wegen der Session-Identifizierung in der URL besonders anfällig. Wird eine Session nicht explizit beendet und hat jemand Zugang zu einem Browser, bei dem gerade eine Session verwendet wurde, kann durch Nutzung der Browser-History die Session weitergeführt werden. Hiervor schützt ein explizites Beenden der Sitzung und ergänzend ein serverseitiger Timeout.

Darüber hinaus können Sessions abgehört oder erraten werden. Das Abhören kann nur verhindert werden, wenn einer Session die konkrete IP-Adresse des Klienten zugeordnet wird. Ist die Session-ID nicht besonders aufwändig gestaltet, kann sie durch Probieren erraten werden. Letzteres kann durch kombinierte Session-IDs und Prüfsummen verhindert werden.

Beispiele für Session-IDs sind:

<http://commsy.wisspro.de/index.php?PHPSESSID=8e0b599f0a...cbf611f5&cid=commsy>

<http://mein.hamburg.de/1,1695,,00.html?ssuid=1921680012...417dfc6cf09b>

Eine Session-ID kann für verschiedene Zwecke eingesetzt werden. Neben dem Speichern von Benutzereinstellungen werden diese verwendet, um das Benutzerverhalten besser analysieren zu können. (Welche Wege auf der Website nimmt ein Anwender? Wie lange ist die Verweildauer bei einzelnen Angeboten? Bestehen inhaltliche Zusammenhänge zwischen den besuchten Seiten?) Hierfür liefern WCMS bereits Mechanismen mit, die die Verwendung von Session-IDs erleichtern und automatisch statistische Daten erheben.

*Problem B.1 (Zuständigkeit – Sessionverwaltung)*

Sowohl das WCMS als auch der Applikationsserver bieten ein Session-Management an. Beide gehen davon aus, dass sie diejenigen sind, die eine Session verwalten. Man ist in der Entwicklung gezwungen, die Entscheidung zu fällen, welches der beiden Systeme diese Aufgabe übernimmt.

*Definition 5 (URL-Rewriting)*

Unter *URL-Rewriting* versteht man die (serverseitige) Fähigkeit, eine Session-ID in alle URLs einer HTML-Seite einzupflegen. Dabei wird die Session-ID in jeden Link und jedes Formular der Seite als Parameter eingebaut.

*Problem B.2 (Zuständigkeit – Seitengenerierung)*

URL-Rewriting kann nur von dem System durchgeführt werden, das für die Session-Verwaltung verantwortlich ist. Am effektivsten liegt diese Aufgabe beim WCMS, da es erst Seiten-Bestandteile zwischenspeichert und nach dem Zusammensetzen zu einer Seite die Session-ID in die URLs einpflegt. Durch die Wahl des Applikationsservers für die Session-Verwaltung wurde dieser Vorteil aufgegeben.

Die Umsetzung der Mehrsprachigkeit kann technisch auf verschiedene Arten realisiert werden. Zur Verfügung stehen unter anderem verschiedene Schablonen im CMS (je Sprache eine). Diese aufwändige Lösung bietet am meisten Freiheit in der Gestaltung. Eine alternative Umsetzung besteht in der Vergabe von Textschlüsseln, die in den Schablonen verwendet und bei Seitenerzeugung durch Einträge in der richtigen Sprache aus der Datenbank ersetzt werden. Diese Lösung reduziert die Anzahl der Schablonen, erreicht ihre Grenze aber bei Grafiken und anderen Objekten. Ein Ressourcen-Manager kann binäre Daten verwalten, sodass z. B. Grafiken ebenfalls sprachabhängig verwendet werden können.

**B.2.3 Bestandteil Free-E-Mail-Plattform**

Eine Maßgabe des Projektes war, jedem Bürger eine kostenlose E-Mail-Adresse unter der Domain *hamburg.de* anzubieten. Dafür musste eine entsprechende Plattform aufgebaut werden. Die Plattform für die Free-E-Mail wurde unabhängig vom Gesamtsystem in der Hinsicht gestaltet, dass sowohl die Entscheidung bezüglich der Hardware als auch der Software unabhängig gesehen wurde. Dennoch sollte in mehrfacher Hinsicht die Möglichkeit der Verknüpfung zwischen den beiden Systemen bestehen. Aus

dem Kernsystem sollten E-Mails abgesetzt werden können (Benachrichtigungen etc.) und über die Web-Oberfläche des Kernsystems sollten E-Mails gelesen werden können (Web-Mail).

Die Definition der technischen Basis für eine Free-E-Mail-Plattform gestaltet sich einfach bzgl. der Softwareauswahl. Es wird ein so genannter *Mail Transfer Agent* (MTA) benötigt, der nach SMTP (simple mail transfer protocol; Postel 1982) arbeitet. Hierfür bieten sich mehrere freie (sendmail, qmail etc.) und kostenpflichtige Produkte (z. B. iPlanet Messaging Server) an. Die Wahl des Produkts wird durch die Kosten, die Sicherheit, die Konfigurierbarkeit und die Skalierbarkeit bestimmt. Der MTA nimmt E-Mails entgegen (von fremden Servern oder E-Mail-Programmen) und legt die eingehende Post in den Postfächern auf dem Filesystem ab. Das dafür verwendete Format ist z. B. maildir (jede Mail als einzelne Datei) oder mailfile (alle E-Mails eines Ordners in einer Datei). Aus historischen Gründen findet bei der Entgegennahme der E-Mail keine Überprüfung der Absender-Identität statt.

Um auf die Postfächer zuzugreifen, bedarf es eines Mailservers, der entweder POP3 (Post Office Protocol – version 3; Myers und Rose 1988) oder IMAP4 (Internet Message Access Protocol – version 4; Crispin 1994) unterstützt. Diese beiden Protokolle werden von den E-Mail-Programmen zum Lesen der E-Mails verwendet. In der Standard-Version verwenden beide Protokolle Kennworte im Klartext zur Autorisierung.

Bei der Planung des Systems wurde von mehreren hunderttausend Konten ausgegangen, da das System in späteren Ausbaustufen viele Domains betreiben wird. Auf dieser Grundlage war es notwendig, über den Speicherverbrauch der Postfächer, die Performanz und die Skalierung des Systems nachzudenken.

#### *Entwurfsentscheidung B.6 (Postfächer)*

Die Postfächer wurden im *mailfile*-Format erzeugt, um die Anzahl der Dateien möglichst klein zu halten. Jede Datei benötigt unter dem Unix-Dateisystem eine so genannte *i-node*. Die Anzahl der *i-nodes* ist je nach Betriebssystemversion beliebig, aber fest während des Betriebs. Der Platzverbrauch einer großen Datei ist ebenfalls günstiger als der vieler kleiner Dateien, da auf der Festplatte immer nur ganze Blöcke einer festen Größe vergeben werden können.

Als Hardware für die Free-E-Mail-Plattform wurden Rechner auf Basis

der Intel-PC-Architektur mit dem Betriebssystem Solaris verwendet. Ausschlaggebend für die Entscheidung war der günstige Hardware-Preis eines PCs. Während des Projektes wurden mehrfach Rechner hinzugenommen, um die Leistungsfähigkeit zu erhöhen (siehe Abbildung 17 auf Seite 244).

Nach der Freischaltung des kostenlosen E-Mail-Dienstes unter *hamburg.de* wurde das System intensiv genutzt. Die lokale Presse nahm dies zum Anlass, das System näher unter die Lupe zu nehmen, und fand eine Schwachstelle im E-Mail-Versand (Röttger 2000). Es war möglich, E-Mails mit beliebigen Absendern zu verschicken; insbesondere konnte als Absender eine existierende E-Mail-Adresse unterhalb von *hamburg.de* verwendet und über den Server von *hamburg.de* verschickt werden (vergleiche Vorfall 4.3.1, hier: `buergermeister@hamburg.de`).

#### *Entwurfsentscheidung B.7 (SMTP after POP)*

Zur Sicherung des Vertrauens in den Mailserver wurde eine Funktion eingefügt, die es nur erlaubt, E-Mails zu verschicken, nachdem man vorher versucht hat, neue E-Mails vom Server abzuholen. Diese Funktion wird *SMTP after POP* genannt, weil erst das POP3-Authentifizierungs-Verfahren verwendet wird, bei dem sich der Server die Adresse des anfragenden Rechners merken kann, und danach das Versenden nur unter der eben verwendeten E-Mail-Adresse von demselben Rechner aus möglich ist.

Obwohl diese Anpassung das System „sicherer“ macht, hat sie doch Nachteile. Erstens hat nicht jedes E-Mail-Programm eine eingebaute Unterstützung dafür (z. B. alte Versionen von Microsoft Outlook Express), was zu unverständlichen Fehlermeldungen beim Benutzer führt. Zweitens spiegelt dies eine Sicherheit vor, die nicht existiert, denn jeder andere SMTP-Server könnte trotzdem unautorisierte E-Mails verschicken. Drittens kann es sein, dass verschiedene Rechner (z. B. alle eines Firmennetzes) zum E-Mail-Server hin mit derselben Adresse erscheinen; dann können alle diese Rechner zum Versenden benutzt werden.

Die Auswahl des Web-Mail-Produkts fand explizit *nach* der Gestaltung der technischen Basis für die Free-E-Mail-Plattform statt, obwohl beide Entscheidungen zusammenhängen. Web-Mail ermöglicht den Zugriff auf die eigenen E-Mails mittels einer Web-Schnittstelle. Aus diesem Grund wurde zuerst ein Produkt über den Hersteller des Content-Management-Systems gesucht. Ein Anbieter einer englischsprachigen Lösung für Web-Mail für den Vignette Story Server wurde gefunden.



*Problem B.3 (Web-Mail-Integration – Speicherort)*

Das in Betracht gezogene Web-Mail-Produkt realisiert die E-Mail-Inbox in der Art, dass bei einem Zugriff alle E-Mails vom Mailserver gelesen und in einer Datenbank gespeichert werden. Dadurch wären diese entweder doppelt gespeichert worden (würden die E-Mails auf dem Server belassen, existierten potenzielle Inkonsistenzen zwischen zwei Orten) oder sie wären nur noch über die Datenbank zugreifbar gewesen (es würde ein POP3-Server benötigt, der zusätzlich eine Datenbank-Schnittstelle realisiert). Eine Integration in die Anmeldung des Portals hätte zusätzlich implementiert werden müssen.

Das vorgenannte Web-Mail-Produkt wurde wegen des Datenbankproblems und der fehlenden Sprachunterstützung nicht gewählt. Der Entwurf hätte erfordert, dass langfristig ein großer Teil der E-Mails in der Datenbank gespeichert würde. Die bisherigen Entwurfsentscheidungen hatten aber nicht ausreichend Festplattenspeicher für die Datenbank vorgesehen. Hier wären zusätzliche Kosten entstanden. Darüber hinaus war die Lösung ausschließlich mit englischen Texten implementiert. Zusätzlicher Aufwand hätte in die Eindeutschung und gegebenenfalls Internationalisierung investiert werden müssen.

Von einem namhaften Datenbankhersteller wurde eine Web-Mail-Integration angeboten, die die gesamte E-Mail über eine Datenbank abwickelt. Alle notwendigen Schnittstellen (SMTP, POP3, IMAP) wurden unterstützt. Eine Java-Implementierung stellte eine Schnittstelle zum Web auf der Basis des IMAP-Protokolls zur Verfügung.

*Problem B.4 (Web-Mail-Integration – Produktreife)*

Ein entscheidendes Problem des zweiten Web-Mail-Produkts lag ebenfalls in der schlechten Einpassbarkeit des E-Mail-Volumens im Bereich des Datenbank-Festplattenspeichers. Allerdings stellte sich noch während der Sondierung heraus, dass weder die Software zum aktuellen Zeitpunkt produktreife erlangt hatte noch es Referenzinstallationen gab, die mehr als 20.000 Benutzer besaßen. Schwerwiegend war, dass das Produkt eine eigene Implementierung eines SMTP-, POP- und IMAP-Servers mitlieferte. Eventuelle Fehlerkorrekturen müssten also von diesem Hersteller kommen. Darüber hinaus war eine Integration in die Anmeldung des Portals augenscheinlich nicht möglich, was gegen die Strategie der Entwicklung lief.

Damit schied auch das zweite Produkt zur Realisierung von Web-Mail aus.

Eine einfache und kostengünstige Alternative stellte ein freies Web-Mail-Produkt mit dem Namen „IMP“ dar (IMP 2001). Es ist in PHP implementiert, nutzt das ebenfalls frei verfügbare Paket „Horde“, unterstützt Mehrsprachigkeit und ist in vielerlei Hinsicht anpassbar. Vorteil dieser Lösung wäre die freie Verfügbarkeit des Source Codes und die Einpassung in bestehende Standards.

#### *Problem B.5 (Web-Mail-Integration – freie Software)*

Erstes Problem des freien Web-Mail-Produkts „IMP“ war es, dass die Lizenzfrage ungeklärt war und nicht weiter aktiv untersucht wurde. Darüber hinaus erforderte die Verwendung von PHP als Skriptsprache die zusätzliche Installation dieses Interpreters auf dem Kernsystem, was die Installation komplexer gestaltet hätte. PHP hätte außerdem weitere Sicherheitsfragen aufgeworfen. Eine Integration in die Anmeldung des Portals hätte darüber hinaus nachträglich eingebaut werden müssen.

#### *Problem B.6 (Web-Mail-Integration – Migration)*

Alle oben aufgeführten Produkte hätten eine mehr oder minder aufwändige Migration der bestehenden E-Mail erfordert. Dieses Argument wiegt insofern schwer, als dass jede Migration potenziell mit Fehlern, Ausfallzeiten und somit schlechter Reklame verbunden ist.

Letztendlich entschied man sich wegen der jeweils vorhandenen Probleme gegen alle Fremdprodukte und nutzte das über deren Architekturen gewonnene Wissen im Sinne einer Studie. Es wurde unter Vignette Story Server über die in den Standard-Java-Klassenbibliotheken vorhandene IMAP-Schnittstelle eine eigene Web-Mail-Implementierung verfasst. Der bereits laufenden Free-E-Mail-Plattform wurde ein IMAP-Mailserver hinzugefügt, der einen transparenten Zugriff auf die bereits existierenden Inboxes erlaubte und die zusätzlichen IMAP-Funktionen (Ordner usw.) zur Verfügung stellte.

### **B.2.4 Bestandteil Free-Homepage-Plattform**

Zusätzlich zum kostenlosen E-Mail-Angebot sollten ebenfalls kostenlose private Homepages unterhalb von *hamburg.de* zur Verfügung gestellt werden.<sup>8</sup> Man setzte sich das Ziel, die Homepages durch einen web-basierten

---

<sup>8</sup> <http://mein.hamburg.de>.

Editor zu erstellen oder aber durch das Hochladen von HTML-Seiten zu ermöglichen. Rahmenbedingung war, dass der Platzverbrauch für die kostenlose Homepage 10 Megabyte pro Benutzer nicht überschreiten durfte.

Bei der technischen Realisierung der Homepages war es zuerst einmal notwendig, für jeden Benutzer einen Ort bereitzustellen, an dem die Homepage technisch abgelegt wird. Hierbei muss unterschieden werden zwischen Homepages, bei denen beliebige Dateien verwendet werden können, und mit dem Homepage-Editor erstellten Homepages, die sich bestimmter Schablonen bedienen. Letztere wurden in der Datenbank gespeichert und mithilfe von Templates über das WCMS erzeugt. Die Entwicklung des Homepage-Editors und der mitgelieferten Templates wurde in einem separaten Projekt durchgeführt.

Um das Hochladen von beliebigen Dateien zu ermöglichen, musste für jeden Benutzer ein Bereich geschaffen werden, an dem Dateien abgelegt werden können. Dieser Bereich musste mit einem Limit versehen werden können (Quota), damit nicht beliebig viele Dateien gelagert werden. Darüber hinaus mussten Sicherheitsaspekte berücksichtigt werden, damit Anwender z. B. nicht den Inhalt der Verzeichnisse anderer Benutzer lesen konnten. Hierfür wurde ein FTP-Server eingerichtet (file transfer protocol, Postel und Reynolds 1985).

#### *Entwurfsentscheidung B.8 (Verteilung der Home(page)-Verzeichnisse)*

Die Home(page)-Verzeichnisse wurden mithilfe eines Hashing-Verfahrens gleichmäßig über die physikalischen Festplatten verteilt, um eine ausgewogene Auslastung der Hardware sicherzustellen und insbesondere unnötige Engpässe im Zugriff zu vermeiden.

Der Zugang zu diesem Dateibereich ist mit dem Konto möglich, das man bei der Anmeldung eingerichtet hat. Darüber hinaus ist es ebenso wichtig, eine leicht verständliche Adresse (URL) bereitzustellen, mit der die Homepage öffentlich zugänglich ist. Dies ist nach außen hin `http://mein.hamburg.de/homepage/<userid>`. Darüber hinaus konnte sich jeder Anwender dafür entscheiden, in einem öffentlichen „Adressbuch“ aufgeführt zu werden.

## B.2.5 Bestandteil Kontenverwaltung

Die Kontenverwaltung des Systems nimmt eine zentrale Stellung ein, da E-Mail, FTP und Personalisierung des Portals davon abhängig sind. Darüber hinaus stellen sich Anforderungen in besonders großem Maßstab, da das Gesamtsystem mit seiner Mandantenfähigkeit mehrere hunderttausend Konten unterschiedlicher Kunden verwalten und zuordnen soll. Aus technischer Sicht bietet sich folgendes Bild: Zusammen mit einem Vertrag<sup>9</sup> erhält der Anwender eine Anmeldung mit Kennwort, das *Portal-Konto*, das mit einem Vertragspaket verknüpft ist. An diese können E-Mail-Adressen, Homepages usw. geknüpft werden. Pro E-Mail-Adresse wird wiederum eine Anmeldung mit Kennwort vergeben; diese sind unterschiedlich, da in der Anmeldung die Domain der E-Mail eingebaut ist (ein Beispiel dafür ist `hans.mayer!hamburg.de`), das *E-Mail-Konto*. Pro Homepage wird eine Anmeldung für den FTP-Zugang benötigt, das *FTP-Konto*. Diese muss aus Sicherheits- und organisatorischen Gründen von der E-Mail verschieden sein. Standardmäßig werden bei Privatpersonen eine E-Mail-Adresse und eine Homepage gleichzeitig mit dem Portal-Konto vergeben. Die Kontenverwaltung muss dann diese Zuordnung leisten.

Zur Speicherung der drei Kontenarten werden unterschiedliche Medien verwendet. Das Portal-Konto wird in einer relationalen Datenbank zusammen mit den angegebenen Daten abgelegt. Das E-Mail-Konto wird in einem Directory-Server verwaltet. Für die FTP-Konten wird ebenfalls ein Directory-Server verwendet, der zusammen mit dem FTP-Programm bei der Anmeldung dem Anwender eine virtuelle Umgebung zur Verfügung stellt.

### *Entwurfsentscheidung B.9 (Portal-Konten)*

Die Entscheidung, die Portal-Konten in einer relationalen Datenbank abzulegen, ist vom Entwicklerteam getroffen worden, als noch nicht über die anderen Bestandteile (E-Mail, FTP) der Kontenverwaltung entschieden worden war. Der Zugriff auf eine relationale Datenbank gestaltete sich technisch leichter als auf einen Directory-Server.

Die Entwurfsentscheidung B.9 erscheint vor dem Hintergrund einer stark diversifizierten Systemlandschaft problematisch. Für die Verwaltung der

---

<sup>9</sup> Eine kostenlose E-Mail-Adresse oder kostenlose Homepage kommt durch einen Vertrag zustande.

Konten wurden Directory-Server mit speziellen Eigenschaften entwickelt, die eine hierarchische Datenbank darstellen, die mit Hinblick auf lesende Zugriffe, Kennwortvergleiche und Replizierung besonders optimiert ist (Yeong et al. 1995).

Diskutiert man die bis hierhin vorgestellte Architektur, müssen verschiedene Kriterien berücksichtigt werden: *Performanz*, *Systemvielfalt*, *Erweiterbarkeit* und *Nutzbarkeit*. Die *Performanz* des Gesamtsystems ist ein entscheidendes Kriterium, das gegenüber den Anwendern und Auftraggebern eine hohe Priorität besitzt. Ihnen sind Implementierungsdetails unwichtig. Die tägliche Nutzung muss bequem und schnell möglich sein. *Systemvielfalt* bezeichnet die Anzahl unterschiedlicher Komponenten (Software- und Hardware-Server) im System, die für verschiedene Aufgaben eingesetzt werden. Aus Sicht der Programmierer und der wirtschaftlichen Rechnung möchte man so wenig unterschiedliche Komponenten wie möglich haben, um nicht für alle das jeweilige Know-how einzukaufen und deren Lizenzen bezahlen zu müssen. Mit der Frage nach *Erweiterbarkeit* wird mittlerweile jedes System konfrontiert und verschiedene Entscheidungen können frühzeitig die Erweiterbarkeit unterstützen oder verhindern. Offene Standards sind meist eine Hilfe bei der Vorbereitung von Erweiterbarkeit. Allerdings können zukünftige Entwicklungen nur prognostiziert werden; zuverlässige Auskünfte gibt es nicht. Die *Nutzbarkeit* bezeichnet eine Anforderung aus der Sicht des Entwicklerteams. Unabhängig davon, ob ein Produkt verfügbar ist, müssen dessen Funktionen verständlich und nutzbar sein. Dazu gehört eine entsprechende Ausbildung in den betreffenden Konzepten und in der konkreten Nutzung des Funktionsumfangs des Produkts.

Die Eigenschaften Performanz, Nutzbarkeit, Erweiterbarkeit und Systemvielfalt lassen sich gut am Beispiel der Kontenverwaltung diskutieren. Konten lassen sich prinzipiell in relationalen Datenbanken verwalten. Hierbei besitzt der Programmierer alle Freiheiten. Es gibt allerdings bereits ein gefestigtes Wissen darüber, wie man Konten am besten verwaltet, und Produkte, die dies nach Standards tun, die so genannten Directory-Server. Durch das Einführen eines Directory-Servers in das System erhöht man die Systemvielfalt. Im Programmiererteam ist zusätzliches Know-how nötig; die Schwellenangst vor einer neuen Technik muss überwunden werden. Die Performanz des Systems ist bei der Verwendung einer Datenbank in deren Grenzen frei gestaltbar; bei einem Directory-Server ist diese bereits auf schnelles Lesen – den häufigsten Anwendungsfall – optimiert. Die Er-

weiterbarkeit des Systems steigt durch den Einsatz eines Directory-Servers, da er mit einem standardisierten Protokoll anderen Produkten die Möglichkeit zum Anschluss gibt. Es sind keine Anpassungen auf beiden Seiten nötig, die nicht schon im Modell vorbereitet wären. Die Nutzbarkeit mag durch die Einführung des Directory-Servers sinken, da dem Programmier-team nicht alle Möglichkeiten bewusst sind.

#### *Problem B.7 (Directory-Server – Verständnis)*

Das Konzept einer hierarchischen Datenbank für einen Directory-Server und die komplexe Syntax für Anfragen (Howes et al. 1995) erschweren die Nutzung.

### **B.2.6 Allgemeine Probleme bei der Produktauswahl**

Obwohl alle Arten von Standardprodukten verfügbar waren und eine Reihe von Produkten in die nähere Auswahl gezogen wurden, sind doch nicht in allen Bereichen Standardprodukte zum Einsatz gekommen. Dies lag an zum Teil unterschiedlichen Kriterien und Argumenten.

#### *Problem B.8 (Lizenz – Preismodell)*

Die verwendeten Produkte besitzen unterschiedliche Lizenzmodelle, die beim Einsatz für mehrere Mandanten zum Teil teuer werden können. Es war deshalb vielfach ausschlaggebend, dass das Lizenzmodell nicht von der Anzahl der verwendeten Maschinen, Prozessoren oder Mandaten abhängig ist.

Dieses Problem traf u. a. bei dem Schlagwortkatalog, dem Content-Management-System und der Community-Software zu (vergleiche Vorfall 4.3.4).

#### *Problem B.9 (Benutzungsschnittstelle – Sprache)*

Da ein Großteil der Software vom amerikanischen oder internationalen Markt stammt, liegt sie zumeist nur in englischer Sprache vor. Unterschiedliche Varianten zur Anpassung an ein europäisches System sind dann möglich. Da das Zielsystem hier mit mehreren Sprachen betrieben werden sollte (Deutsch, Englisch, Türkisch, Dänisch, Russisch etc.), musste jedes Produkt die Mehrsprachigkeit unterstützen.

Die Anforderung nach Mehrsprachigkeit hat z. B. bei Web-Mail zum Ausschluss des aussichtsreichsten Produkts beigetragen.

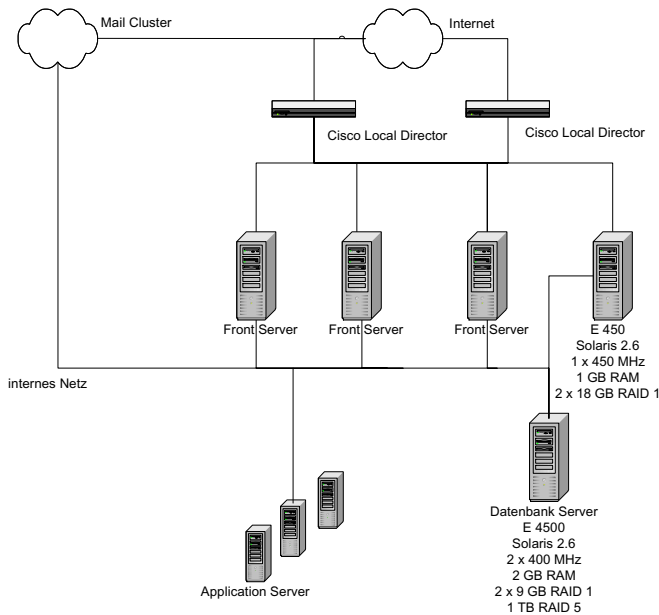


Abbildung 16: Hardware-Architektur von *hamburg.de* für das Webportal

## B.2.7 Softwaretechnische Probleme

Eine softwaretechnische Herausforderung stellt das Einspielen eines Updates in das hier skizzierte System dar. Wenngleich alle Bestandteile der technischen Plattform mehrfach ausgelegt sind, so ist doch unklar, wie man ohne Abschalten des Systems Teile der Software aktualisieren kann.

Das im Projekt gewählte Vorgehen sah ein Abschalten des Zugangs für die Öffentlichkeit am Montagmorgen vor. Es wurde eine Sicherungskopie angefertigt. Danach wurde eine Aktualisierung des Systems versucht und getestet. War diese nicht erfolgreich, wurde gegen 11 Uhr die Sicherung eingespielt und das System gegen 12 Uhr wieder öffentlich geschaltet. Im Erfolgsfall konnten die Anwender das neue System nutzen.

Dieses Vorgehen ist aus softwaretechnischer Sicht unbefriedigend gelöst. Kein Projektbeteiligter war bereit oder in der Lage, die damit verbundene Fragestellung weiterzuverfolgen.

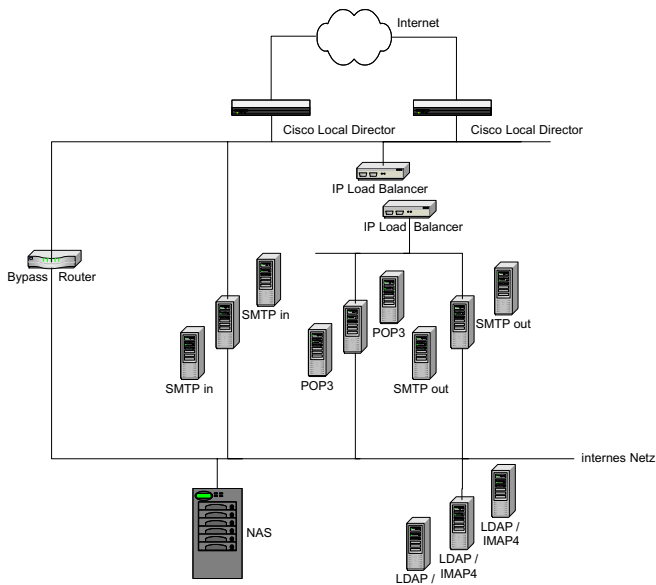


Abbildung 17: Hardware-Architektur von *hamburg.de* für die Free-E-Mail

## B.3 Verwendete Software-Komponenten

Zur Implementierung der Infrastruktur von *hamburg.de* wurde eine große Anzahl von Softwarekomponenten verwendet. An dieser Stelle soll auf die wesentlichen eingegangen werden, um einen Eindruck von der Diversität zu geben.

### B.3.1 Content-Management-System und Webserver

Der Zweck eines Content-Management-Systems (CMS) ist es, eine Unterscheidung zwischen Layout und Inhalt bereitzustellen, bei der Inhalte erst zu einem späten Zeitpunkt in das Layout eingefügt werden, um Flexibilität in Bezug auf die Präsentation zu besitzen und darüber hinaus eine große Zahl von sich regelmäßig ändernden Inhalten zu verwalten. Das CMS stellt Dienste zur Verfügung, die jedem Inhalt Erscheinungs- und Verfallsdatum zuordnen können, um den Zeitraum der Präsentation genau festzulegen (so genanntes „Freischalten“). Darüber hinaus können die gestalterischen Ele-



mente der Webseiten in Komponenten zerlegt werden, die einzeln gepflegt werden können. Eine Webseite setzt sich somit aus Komponenten zusammen. Die besondere Leistung des CMS besteht nun darin, diese Seite auf Anforderung (z. B. durch den Betrachter der Seite) aus den Komponenten unter Beachtung der Regeln zusammenzubauen. Eine weitere wichtige Eigenschaft des WCMS ist es, bereits generierte Komponenten nur dann erneut zu generieren, wenn sich die Informationen darin geändert haben. Somit wird Rechenzeit bei der Seitenerstellung eingespart und die Performance des Gesamtsystems verbessert.

Ein CMS muss mandantenfähig sein, damit es in diesem Projekt eingesetzt werden kann. Dies begründet sich aus der speziellen Gestaltung des Systems, bei dem alle Inhalte in einer Datenbank verwaltet werden. Der Einstieg über die Portalseite eines Mandanten legt bestimmte Kriterien fest (z. B. Standardsprache, regionaler Bezug). Die Abrechnung und weitere vertragliche Bedingungen sind nun an diesen Einstieg gebunden, d. h. die Grundlage für die Abrechnung des Betreibers gegenüber dem Mandanten, die Logdatei, muss in jedem Eintrag eine Referenz auf den Mandanten enthalten, oder es müssen getrennte Logdateien für die Mandanten angelegt werden.

### **B.3.2 Shopsystem**

Ein Shopsystem stellt elektronische Läden zur Verfügung, die pro Inhaber separat verwaltet und für die jeweils eigene Produktbestände gepflegt werden können. Zu jedem Produkt kann neben einer Beschreibung ein Bestand hinterlegt werden, der durch Verkäufe reduziert wird. Das Shopsystem stellt Mechanismen zum Einkauf für Kunden zur Verfügung, die es erlauben, ein virtuelles Geschäft zu betreten, einen Einkaufswagen mit Waren zu füllen und an der Kasse zu bezahlen (Einkaufswagen-Metapher). Das Shopsystem stellt hier unterschiedliche Module zur Bezahlung zur Verfügung, die vom Geschäftsinhaber gewählt werden können. Unterschiedliche Verfahren zur Berechnung des Rechnungsbetrags (Liefer- und Zahlungsbedingungen) können in Abhängigkeit von den Produkten definiert werden.

Die zum Zeitpunkt der Auswahl auf dem Markt verfügbaren Shopsysteme treffen bestimmte Grundannahmen über die Struktur des Portals zur internen Verwaltung wie zur Präsentation, die mit dem Portalkonzept des hier beschriebenen Systems nicht zusammenpassen.

*Problem B.10 (Shopsystem – Marktplatz)*

Ein Shopsystem definiert durch die Summe der in ihm eingetragenen Geschäfte einen Marktplatz. Dieser dient als Promenade für den Endanwender. Diese Annahme erwies sich als unbrauchbar im konkreten Projekt, da das System für alle Shops auf den unterschiedlichen Domains gleichzeitig verwendet werden soll. Diese sind regional aber stark verstreut und sollen jeweils durch den aktuell vom Anwender gewählten Ort und Umkreis bestimmt werden. Das Shopsystem erlaubt aber keine regionale Einschränkung des Marktplatzes.

*Problem B.11 (Shopsystem – Artikelsuche)*

Ein Shopsystem stellt eine eigene Implementierung einer Artikelsuche über alle Shops oder innerhalb eines Shops zur Verfügung. Diese Funktion ist wünschenswert, wenn das Shopsystem einzeln eingesetzt wird. Im Kontext eines Stadtinformationssystems ist eine Integration in die Suche des Portals gefordert, um dem Anwender eine konsistente Suche anzubieten. Dies war nicht möglich, weil weder eine Schnittstelle für die Suche zur Verfügung stand noch eine Schnittstelle für das Auslesen des Suchergebnisses zur Präsentation.

Ein Widerspruch in der Gestaltung der Software-Architektur wird im Zusammenhang mit dem gewählten Shopsystem deutlich: Das letztendlich gewählte Shopsystem „Telemall“ erfordert den PHP-Skriptinterpreter auf den Maschinen zur Auslieferung der Webseiten. Genau wegen dieses Skriptinterpreters wurde das Web-Mail-Produkt „IMP“ nicht genommen.

**B.3.3 Integration von Werbeformen**

Werbung auf dem Portal ist ein wichtiger Punkt, da hier die wirtschaftlichen Aspekte auf die technischen treffen. Werbung auf Webseiten ist in verschiedenen Formen möglich. Werbebanner sind die üblichste Form. Sie erscheinen meist als (bewegte) Grafiken am Anfang der Seite über die volle Seitenbreite. Eine wichtige Forderung der Auftraggeber ist es, dass das Banner immer innerhalb der ersten 600 Pixel erscheint. Vielfach werden die Banner am Fuß der Seite wiederholt. Neben den Bannern gibt es als Werbeformen Sponsoren-Logos, schmale Banner und so genannte Skyscraper. Sponsoren-Logos erscheinen direkt an einer Nachricht oder Information, um klarzustellen, wer hierfür die Kosten übernommen hat. Schmale Banner

werden in die Seite eingestreut, um beim Lesen der Seite besser im Sichtfeld des Benutzers zu bleiben. Skyscraper sind Banner im Wolkenkratzer-Format, die an der linken oder rechten Kante der Seite entlanglaufen.

#### *Problem B.12 (Werbung – Zwischenspeichern)*

Die Einführung von Werbebannern ist aus technischer Sicht ein starker Eingriff in das System, weil die Zwischenspeicherung von Seiten(-bestandteilen) dadurch erschwert wird. Banner dürfen nicht zwischengespeichert werden, da sie in Abhängigkeit von der gekauften Menge bzw. der Tageszeit usw. geschaltet werden. Damit muss das Konzept zur Zwischenspeicherung der Seitenbestandteile revidiert werden.

Ein weiteres technisches Problem bei Werbeformen ist die Abrechnung. Der Kunde bestimmt, in welchen Situationen sein Werbebanner gezeigt werden soll (z. B. Tageszeit). Er kauft eine Menge so genannter Page-Impressions und erwartet dann eine detaillierte Abrechnung, wann das Banner wie häufig gezeigt wurde. Diese Abrechnung muss aus den vom System gesammelten Daten erfolgen.

## **B.4 Zusammenfassende Betrachtung**

Das *hamburg.de*-Projekt hat gezeigt, wie eine komplexe Infrastruktur neu gestaltet werden kann. Ein besonderer Problemschwerpunkt des Projektes lag in der großen Anzahl beteiligter Organisationen und deren unterschiedlichen Interessen. Hinzu kommen verschiedene Software-Produkte mit unterschiedlichen Integrations-Ansprüchen. Es ergibt sich ein komplexes Zusammenspiel von Technik und Organisationen.

### **B.4.1 Rückwirkungen**

In diesem Projektkontext gibt es zwei Ebenen der möglichen Rückwirkungen. Eine Ebene sind die Rück- bzw. Auswirkungen von *hamburg.de* auf Bürger bzw. Benutzer des Portals. Die Ebene werde ich hier nicht betrachten, da im Rahmen des Projektes wenig Arbeit in dieser Richtung praktiziert wurde. Die zweite Ebene sind die Rückwirkungen der Entwicklungsarbeit von *hamburg.de* auf die an der Entwicklung beteiligten Organisationen.

Eine Rückwirkung auf die persönliche Infrastruktur der Mitglieder der Steuerungsgruppe bestand in der E-Mail-Nutzung.

#### *Rückwirkung B.1 (hhde-E-Mail-Nutzung)*

Ein Großteil der Kommunikation innerhalb der Steuerungsgruppe erfolgte über E-Mail. Um die Sicherheit der Kommunikation zu gewährleisten, wurde Pretty Good Privacy (PGP) eingesetzt, mit dem die E-Mails verschlüsselt wurden. Alle Beteiligten der Steuerungsgruppe installierten sich deshalb diese Software, um dem E-Mail-Verkehr untereinander folgen zu können.

Der Einsatz dieses Verschlüsselungsverfahrens für E-Mails führte dazu, dass die Beteiligten nach verschiedenen Lösungen für ihre gewohnten E-Mail-Programme suchten und die unterschiedlichen Eigenschaften und Probleme diskutierten.

### **B.4.2 Wahrgenommene Problemfelder**

Zu den wahrgenommenen Problemfeldern im Projektverlauf gehört zentral die Frage der Entscheidungsfindung. Aufgrund der vielen beteiligten Organisationen, deren unterschiedlichen Zielen und unklaren Zuständigkeiten wurden viele Entscheidungen spät getroffen oder häufig rückgängig gemacht. Das Problem der Entscheidungsfindung beruht dabei auf zwei anderen Problemen: der Artikulation von Interessen und der Notwendigkeit, eine Vielzahl von Interessen zu antizipieren. Die Artikulation von Interessen findet gegenüber der Steuerungsgruppe nur vermittelt statt. Dadurch gehen Details verloren und Rückfragen werden schwer möglich. Ungeklärte Fragen führen zu Antizipationen im Entscheidungsprozess. Diese methodisch abzustützen und rückzukoppeln ist in der Projektkonstellation nicht gelungen.

### **B.4.3 Infrastrukturelle Herausforderungen**

Die organisatorische Herausforderung bei der Entwicklung dieser Infrastruktur lag in der großen Anzahl von beteiligten Organisationen. Wie oben bereits ausgeführt, wurden dadurch Entscheidungsfindungen schwierig und widersprüchliche Interessen artikuliert. Wiederholt auftretende Zielkonflikte auf der technischen Ebene sind nicht zuletzt durch Interessenverschiebungen auf der organisatorischen Ebene zustande gekommen.

*Mandantenfähigkeit*

Eine Infrastruktur kann von einem Betreiber in mehreren Kontexten eingesetzt werden – an mehrere Kunden verkauft oder vermietet werden –, wenn die Infrastruktur sich gegenüber den Kunden als eine Einheit darstellt, die nur für ihn gestaltet wurde. Die Herausforderung besteht darin, durch – im Vergleich zu separater Einrichtung – weniger Aufwand die Infrastruktur in diese Lage zu versetzen.

*Ausfallsicherheit*

Eine Infrastruktur kann als Ganzes oder teilweise ausfallen. Ein ganzer Ausfall liegt dann vor, wenn die Infrastruktur für einen Klienten nicht nutzbar ist. Ein teilweiser Ausfall liegt vor, wenn zwar Teile der Infrastruktur genutzt werden können, bestimmte aber nicht zur Verfügung stehen. Die Herausforderung besteht nun darin, dem Anwender den Eindruck zu vermitteln, dass jederzeit alle (oder möglichst viele) Teile der Infrastruktur zur Verfügung stehen. Dies kann z. B. durch Redundanzen (s. u.) hergestellt werden.

*Redundanz*

Mithilfe von Redundanzen kann man eine Vielzahl von Problemen lösen. Bei einem Ausfall einer redundant ausgelegten Komponente des Systems springen zuerst die übrigen unter Verlust der vollen Leistungsfähigkeit ein. Die Handlungsfreiheit für Fehlerbehebung ist gegeben. Die Herausforderung bei der Infrastrukturgestaltung besteht darin, die Stellen zu erkennen, an denen Redundanzen wirkungsvoll eingesetzt werden.

*Skalierbarkeit*

Die Skalierbarkeit bezieht sich in diesem konkreten Projekt auf die Anzahl der ausgelieferten Webseiten (in Abhängigkeit von den betriebenen Domains), die Anzahl der bereitgestellten Homepages und verfügbaren E-Mail-Postfächer. Die Herausforderung besteht darin, eine Architektur so zu gestalten, dass sie für höhere quantitative Anforderungen modular durch weitere Exemplare bereits bestehender Komponenten ergänzt werden kann.

*Performanz*

Performanz im Bereich von Infrastrukturen beschreibt das Verhalten eines Bestandteils genauso wie des Gesamtsystems. Es geht darum, dass die In-

Infrastruktur im Eindruck des Anwenders zügig reagiert. Performanz kann durch Redundanz erhöht werden.

#### *Abstützung auf Standards*

Die Abstützung auf Standards beim Aufbau der Infrastruktur ist essentiell zur Vermeidung von Eigenentwicklungen. Die Herausforderung besteht darin, relevante Standards zu (er)kennen und im Projekt einzusetzen. Dies betrifft sowohl die nach außen hin sichtbaren Standards wie auch intern genutzte.

#### *Erweiterbarkeit*

Das ist die Möglichkeit, auf sich ändernde Anforderungen zu reagieren, die insbesondere eine Vergrößerung des Funktionsumfangs zur Folge haben. Die Herausforderung besteht darin, ein weiterhin ausgewogenes System zum Betrieb der Infrastruktur zu besitzen, das in seiner Qualität in Bezug auf die anderen Herausforderungen gleich geblieben ist.

#### *Nutzbarkeit*

Nutzbarkeit ist ein evidentes Problem bei jeder großen Infrastruktur. Die Herausforderung ist, alle Teile des Systems dem Anwender zur Verfügung zu stellen, d. h. die Funktionen zu kommunizieren und in einfacher Art und Weise erschließen zu können.

#### *Systemvielfalt*

Systemvielfalt ist ein Problem bei der Errichtung von Infrastruktur, das durch die beteiligten Personen definiert wird. Je größer die Systemvielfalt wird, umso schwerer wird es, im Team ausreichend ausgebildete Personen zu haben. Die Herausforderung besteht in der Freiheit, alle gewünschten Produkte einzusetzen und gleichzeitig die Komplexität möglichst niedrig zu halten.

#### *Aktualisierbarkeit, Wartbarkeit*

Eine Infrastruktur muss von Zeit zu Zeit auf den aktuellen Stand gebracht oder gepflegt werden. Für diese Updates bzw. die Pflege müssen in der Infrastruktur Mechanismen vorgesehen sein (vergleiche Sicherungskasten, Hauptventil usw.). Akzeptiert wird ein partieller Ausfall für einen kurzen Zeitraum. Bei informatischen Infrastrukturen erwartet man darüber hinaus, dass diese im Dauerbetrieb gepflegt werden können.

## - C -

# Empirieprojekt *Community System*

Dieses Kapitel stellt den zeitlichen Verlauf der *CommSy*-Entwicklung im Zeitraum Frühjahr 1999 bis Frühjahr 2001 dar. Es wird ein Überblick über den Funktionsumfang gegeben, die notwendigen technischen Voraussetzungen zum Betrieb werden betrachtet und die organisatorischen Rahmenbedingungen für die jeweiligen Projekte aufgefächert.

Das *CommSy*-Projekt entwickelte sich aus einer Arbeitsgruppe zum Thema *Knowledge Management*, die sich als *Knowledge Network* oder kurz *KnowNet* bezeichnete. Diese Arbeitsgruppe begann ihre Arbeit im Frühjahr 1999 als eine extracurriculare Veranstaltung, initiiert von Professor Arno Rolf. Die Gruppe bestand aus drei Wissenschaftlichen Mitarbeitern und vier Studierenden aus unterschiedlichen Arbeitsbereichen und einem externen Promovenden.

Die *KnowNet*-Gruppe traf sich etwa alle vier Wochen über einen Zeitraum von ca. 10 Monaten. Man wollte das eigene Knowledge Management technisch unterstützen. Zuerst wurden gemeinsame Dokumente und allgemeine Informationen auf einem Webserver in einem geschützten Bereich abgelegt. Um diese technische Unterstützung bereitzustellen, wurde ein Netscape-Enterprise-Server 3.6 mit der Funktionalität zum Hochladen von Dokumenten, die im Netscape Composer erstellt wurden, eingesetzt. Die Teilnehmenden mussten hierfür die Struktur und die Verlinkung der Dokumente selber pflegen.

Nach kurzer Zeit wurde die Frage einer weitergehenden Software-Unterstützung diskutiert. Weil einige Mitglieder der Gruppe besonders interessiert an dynamischen Webseiten und der dafür benötigten Technik waren und weil die anderen keine Standardprodukte mit proprietärer Software (wie z. B. Lotus Notes) verwenden wollten, begannen ein Student und ein Mitarbeiter einen Prototypen zu bauen, der intensiv zwischen einer kleinen Gruppe der Beteiligten diskutiert wurde.

Während des Sommers 1999 wurde dieser Prototyp zu einem geschlossenen System weiterentwickelt, das von mehreren Mitgliedern der *KnowNet*-Gruppe verwendet wurde. Während der Planung des Wintersemesters

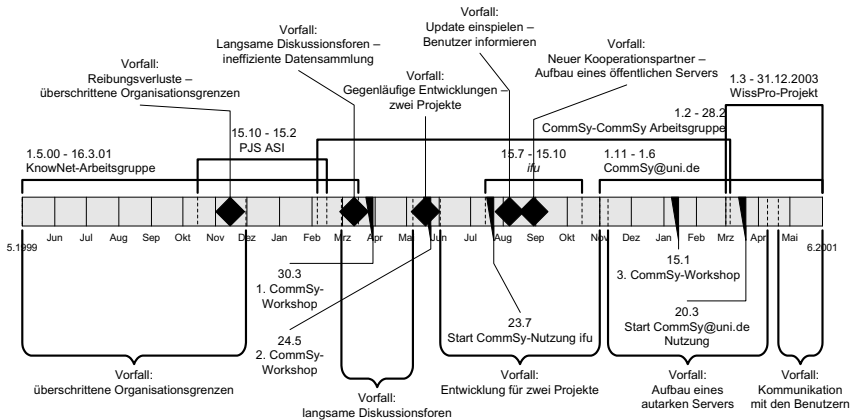


Abbildung 18: Der Zeitverlauf des *CommSy*-Projektes

und besonders eines Projektseminars wurde die Frage erörtert, ob das System in Lehrekontexten eingesetzt werden könnte. Eine Anzahl von Nachteilen bzw. fehlenden Funktionen wurde identifiziert und einer der Studierenden wurde als Studentische Hilfskraft eingestellt, um die essentiellen Änderungen einzubauen. In diesem Zusammenhang wurde ich gefragt, ob ich die notwendige Unterstützung in Bezug auf Server-Administration und Kontenpflege leisten würde. Hieraus entwickelte sich eine längerfristige Zusammenarbeit bei der Entwicklung dieses Community Systems über Organisationsgrenzen hinweg.

Einen Überblick über die verschiedenen Projekte und deren zeitliche Einordnung bietet die Abbildung 18. Im Folgenden werden diese Projekte in ihrer Reihenfolge skizziert und Entwicklungsschwerpunkte, Entwurfsentscheidungen und Probleme benannt. Die Bestandteile der technischen Architektur werden im Zusammenhang dargestellt. Zuvor wird im direkt folgenden Abschnitt eine thematische Einführung in das Gebiet der Community Systems gegeben und der Funktionsumfang des aktuellen Systems motiviert.

Zum Zeitpunkt der Erstellung dieses Textes wird weiterhin an dem Projekt *CommSy* entwickelt. Die damit verbundenen Aktivitäten finden im Rahmen des WissPro-Projektes (s. u.) statt. In diesem Abschnitt kann nur auf die Projekte bis Anfang 2001 eingegangen werden.



Die empirische Basis für dieses Projekt bilden eine große Anzahl von Interviews mit Studierenden, die einige Wissenschaftliche Mitarbeiter im Rahmen von Projekten geführt haben, die Beiträge in den Diskussionsforen der entsprechenden virtuellen Projekträume und meine Sammlung aller E-Mails, die im Laufe der Projekte ausgetauscht wurden. Eine ausführliche Aufstellung des vorliegenden Materials wird in Kapitel 4.1.1 gegeben.

## C.1 Funktionsumfang und Basisdefinition

Die *CommSy*-Entwicklung fand seit Mai 1999 in einer interdisziplinären Arbeitsgruppe aus etwa fünfzehn Studierenden und Wissenschaftlichen Mitarbeitern im Fachbereich Informatik der Universität Hamburg (Gumm et al. 2000) statt. *CommSy* steht für *Community System* und ist eine web-basierte Anwendung zur Unterstützung der Kommunikation und Koordination in Lern- und Arbeitsgruppen.

Mit dem Begriff „Community System“ lehnen wir uns dabei an die Definition von Schatz an:

“An electronic community system is a computer system which encodes the knowledge of a community and provides an environment which supports manipulation of that knowledge. Different communities have different knowledge but their environment has great similarities. The community knowledge might be thought of as being stored in an electronic library.”

(Schatz 1991, S. 88)

*CommSy* unterstützt vernetzte Projektarbeit durch seine Funktionalitäten in Bezug auf verschiedene Kommunikationsarten und für den Umgang mit unterschiedlichen Arbeitsmaterialien. Als Projekt wird das gemeinsame Arbeitsziel aller Beteiligten bezeichnet. Projektgruppen oder kurz Gruppen bezieht sich auf Untergruppen der im Projekt Mitwirkenden. Die hier dargestellte Funktionsbeschreibung bezieht sich auf einen Stand des Systems, der für etwa die zweite Hälfte des betrachteten Zeitraums gilt (Bleek und Pape 2001).

- Neuigkeiten und Termine können in den entsprechenden Bereichen „Neuigkeiten“ und „Termine“ bekannt gegeben werden. In diesen Bereichen kann jedes Mitglied eventuell für das gesamte Projekt oder für einzelne Projektgruppen interessante Neuigkeiten und Termine

einstellen. Autoren können Einträge einfach als für nur eine Projektgruppe relevant kennzeichnen, indem sie eine der im System bekannten Projektgruppen referenzieren. Aktuelle Informationen werden auf der Einstiegsseite eines *CommSy*-Projektraums gezeigt (siehe Abbildung 19 auf Seite 256). Durch Anwählen einer Rubrik gelangt der Benutzer in den entsprechenden Bereich, in dem alle jemals eingestellten Einträge in Form einer archivartigen Übersicht dargestellt werden. Es kann dann ein einzelner Eintrag ausgewählt werden, um sich detailliertere Informationen zu dem Eintrag anzeigen zu lassen.

- Neben dem Einstellen von Ankündigungen (mittels Neuigkeiten und Terminen) unterstützt *CommSy* eine reichhaltigere Kommunikation, weil es jedem Mitglied erlaubt ist, an jeden Eintrag in jedem Bereich Anmerkungen zu hängen. Dies kann auf unterschiedliche Weise genutzt werden, z. B. um eine Rezension eines Buches zu publizieren, um eine Tagesordnung zu einer Sitzung zu veröffentlichen, um weitere Informationen zu einer Neuigkeit anzubieten oder um einen Hyperlink zu einem Eintrag bereitzustellen.
- Kontroversere Kommunikation kann in Diskussionsforen stattfinden. Dieser Bereich ist eine einfache Variante von Internet-Newsgroups. Jedes Mitglied des *CommSy*-Projektraums kann sich an jeder Diskussion beteiligen bzw. eine neue Diskussion initiieren.
- Eine eher indirekte Unterstützung der Kommunikation stellt *CommSy* mit den Bereichen „Personen“ und „Gruppen“ zur Verfügung. Im Bereich „Personen“ hat jedes Mitglied eine eigene „Mini-Homepage“, um sich mit Namen, Bild und weiteren Kontaktdaten zu präsentieren. Diese Kontaktinformationen können verwendet werden, um außerhalb des Systems Kommunikation zu etablieren. Im Bereich „Gruppen“ können neue Projektgruppen eingerichtet werden und man kann existierenden beitreten. So werden die Interessen der Mitglieder deutlich und die Struktur des Gesamtprojektes wird transparent.
- Im Bereich „Quellen“ können alle Mitglieder Informationen zu Büchern oder anderen Arbeitsmaterialien wie Artikeln, CDs, Videos, Hyperlinks usw., die sie für ihre Arbeit brauchen, sammeln. Mit dieser Möglichkeit können das Projekt, aber auch die Projektgruppen gemeinsame Literaturlisten bzw. Quellenlisten pflegen. Jeder kann

natürlich zu einer Quelle Arbeitsmaterial in Form einer Datei hängen (Bleek et al. 2000).

- Jedes Mitglied hat im gesamten Community System die Möglichkeit, Dateien an fast alle Einträge (Neuigkeiten, Termine, Gruppen, Anmerkungen in Diskussionsforen, Quellen) zu heften. Mit dieser Datei-Funktionalität unterstützt *CommSy* den Umgang mit verschiedensten Arbeitsmaterialien: Textdokumente genauso wie Multimediateien. Diese Funktionalität zum Heraufladen und Herunterladen von Dateien kann gut zum Austausch von Arbeitsmaterialien innerhalb des Projektes bzw. innerhalb der Projektgruppen genutzt werden.
- Darüber hinaus gibt es den Bereich „Workspaces“, in dem Gruppen von Mitgliedern zusammen asynchron HTML-Dokumente schreiben und anschließend präsentieren können. So unterstützt *CommSy* nicht nur den Umgang mit „fremden“ Arbeitsmaterialien, sondern alle Mitglieder haben die Möglichkeit, eigene HTML-Dokumente in *CommSy* zu gestalten.
- Ein Suchmechanismus unterstützt den schnellen Zugriff auf alle im System gespeicherten Einträge. Zurzeit bietet *CommSy* nur eine simple Volltextsuche, da bisher kein Mitglied den Bedarf eines verfeinerten Suchtools geäußert hat.

Wie bereits oben erwähnt, war die Entscheidung für die Eigenentwicklung durch zwei Faktoren bestimmt. Als erster Faktor ist das Bedürfnis zu nennen, keine proprietären Programme zu verlangen, die von den Anwendern erst installiert werden müssen. Vielmehr sollten Standardprodukte, die bereits jedem bekannt und für jeden verfügbar sind, angewendet werden (hierzu zählte damals der Netscape Communicator mit seinem Composer). Als zweiten Faktor gab es bei den Beteiligten ein großes Interesse, die zu dem Zeitpunkt verfügbare Technik zur dynamischen Generierung von Webseiten zu untersuchen und bewerten zu lernen.

Der erste Faktor stellt klar, dass ausschließlich minimale Webseiten für das System verlangt werden können. Diese müssen dem HTML-Standard genügen und dürfen sich nicht auf Java Applets, DHTML, Javascript oder Ähnliches verlassen. Jeder Browser muss genügen, um das System zu betreten, Dinge zu lesen und eigene Eintragungen einzustellen. Am Fachbereich Informatik der Universität Hamburg ist der Netscape Communicator

**WissPro CommSy**

Home | Events | News | People | Discussion forums | Sources | Groups | Files | Search | Internal | Logout

Home

| Events (coming up)                   | Date           | Duration          | Location                               | Relevant for          | new |
|--------------------------------------|----------------|-------------------|--|-----------------------|-----|
| 3. WissPro-Workshop                  | Fri 20.07.2001 | 9:30              | Redderkoppel 43                        | FBI-WissPro           |     |
| Gartenparty                          | Fri 13.07.2001 | 17:00             | WissPro Räume                          | Technik               |     |
| CommSy Sprint                        | Thu 12.07.2001 | 2 days            | Vorschläge willkommen!                 | Technik               |     |
| CommSy Dinner                        | Thu 12.07.2001 | 19:00 - 22:00     | F-224, Informatikum, HH-Stellingen     | Technik               |     |
| Arbeitsgruppe Technik                | Wed 11.07.2001 | 14:00 - 18:00     |  |                       |     |
| Arbeitsgruppe Kooperationsplattform  | Wed 11.07.2001 | 10:00 - 12:30     | F-224, Informatikum, HH-Stellingen     | Kooperationsplattform |     |
| CommSy Architekturgruppe             | Mon 09.07.2001 | 14:30 - 17:00     |  | Technik               |     |
| Philosophie (I) und Essen in WissPro | Thu 05.07.2001 | 12 - 14           | Besprechungsraum Rathaus Stellingen    | alle in WissPro       |     |
| Arbeitsgruppe Mediendidaktik         | Wed 04.07.2001 | 13:00 - 15:00     | (Koppelstr. gegenüber Ecke Bassehlweg) | Mediendidaktik        |     |
| Arbeitsgruppe Wissensarchiv          | Wed 04.07.2001 | 13:00 - 16:00     | Pf Raum 513                            | WissPro Wissensarchiv |     |
| Präsentation Logo-Entwürfe           | Tue 03.07.2001 | 09:30 - ca. 10:30 | Besprechungsraum, HH, FBI              | alle in WissPro       |     |
| Kunstgenuss: Requiem                 | Sat 30.06.2001 | 20:00 - 21:30     | F-224 Hauptkirche St. Petri, Hamburg   | FBI-WissPro           |     |

| News (of the last 7 days)                     | Date           | Time     | Entered by      | Relevant for    | new |
|---|----------------|----------|-----------------|-----------------|-----|
| Erfahrungsseminar Audio?                      | Wed 27.06.2001 | 01:13:43 | Volker Gnann    | alle in WissPro |     |
| Neues Diskussionsforum "Auditive Medien"      | Wed 27.06.2001 | 01:02:25 | Volker Gnann    | alle in WissPro |     |
| PubSy is online again                         | Mon 25.06.2001 | 18:59:26 | Iwer Jackewitz  | alle in WissPro |     |
| Wisspro in Lübeck                             | Mon 25.06.2001 | 16:24:40 | Barbara Kleinen |                 |     |
| Zusammenfassung (oder Dossier?) Wissensarchiv | Wed 20.06.2001 | 17:02:27 | Dorina Gumm     | alle in WissPro |     |

**Members (contact information)**

Okday Akkaya | Sonja Anand | Alexander Auch | Stefan Aufenanger | Günter Binge | Wolf-Gideon Bleek | Christiane Floyd | Volker Gnann | Jose Manuel Gonzalez Vazquez | Dorina Gumm | Peter Harder | Michael Herczeg | Iwer Jackewitz | Michael Janneck | Marli Jennecke | Oliver Jessauke | Helge Kahler | Herbert Klaren | Barbara Kleinen | Dede Kruse | Joeran Muuss-Merholz | Ulrike Najm | Horst Oberquelle | Katja Priem | Kai Raudzus | Anica Richardt | Markus Rohde | Arno Rolf | Bernd Schatz | Christina Schryer | Anne Stephan | Monique Strauss | Ralf D. Tscheuschner | Carmen Ueckert | Florian Wanning | Ursula Woerz | Ole Wotschach | Bernd Wolf

**Discussion forums** (interact with other members on specific topics)

**Reference Sources** (relevant books, papers, links, etc.)

**Groups** (Overview and administration of all groups used within the system)

**Files** (All files stored in this CommSy)

**Search** (search features for files and reference literature)

**Internal** (Project room configuration and administration)

Home | Events | News | People | Discussion forums | Sources | Groups | Files | Search | Internal | Logout

**comm<sup>3</sup>sy**  
Development \$18 footer.php.v.1.55 2001/05/05 10:27:05 jakewit Exp \$

logged on: Michael Janneck (Janneck) - 27.06.2001, 12:54

Abbildung 19: Die Einstiegsseite<sup>1</sup> des *CommSy*

populär, weil er ein freies Produkt ist, eine gute Reputation hat und für verschiedenste Betriebssysteme zur Verfügung steht. Zum Startzeitpunkt des *CommSy* wurden die Versionen 4.5x bis 4.7x benutzt. Einige benutzten Windows-Arbeitsstationen mit Microsoft Internet Explorer 3.x bis 4.x. Darüber hinaus nutzten einige Studierende zu Hause modifizierte Browser ihrer ISPs.<sup>2</sup>

Der zweite Faktor reflektiert das Interesse an den frei verfügbaren Software-Bestandteilen. Der Arbeitsbereich SWT hatte einen Netscape-Enterprise-3.6-Webserver seit ca. zwei Jahren funktionsfähig im Einsatz. Dieser

<sup>1</sup> Dank an Michael Janneck für die Abbildung.

<sup>2</sup> ISP = internet service provider (z. B. Deutsche Telekom, T-Online, AOL Deutschland, CompuServe)

Server wurde in einer Vielzahl von Einsatzgebieten (Lehrveranstaltungen und Forschungsprojekten) verwendet. Die technische Herausforderung in diesem Zusammenhang war es, in derselben Konfiguration zusätzlich die Skriptsprache PHP zu verwenden, um dynamische Webseiten herzustellen. PHP als Sprache erlaubt einen besonders komfortablen Zugriff auf die relationale Datenbank MySQL. Die Datenbank konnte leicht installiert werden. Allerdings ergaben sich Schwierigkeiten bei der Verbindung zwischen Netscape Enterprise Server und dem Skriptinterpreter. Konkret haben drei Personen an dem Problem fast zwei Wochen gearbeitet. Nach ihrem Erfolg haben sie die Ergebnisse auf einer „HowTo“-Webseite dokumentiert und sie in einigen Newsgruppen zur Verfügung gestellt. Darauf gab es noch viele Monate später Resonanz anderer Hilfe Suchender.

Um sicherzustellen, dass nur Mitglieder der Arbeitsgruppe den Zugriff auf die dynamischen Webseiten haben, wurde der bereits vorhandene Netscape-Directory-Server 4.0 verwendet. Der Directory-Server wurde so eingerichtet, dass er dem Webserver zur Kontenverwaltung diene. So genannte *Configuration Styles* konnten dafür verwendet werden, auf bestimmten Pfaden Zugriffsbeschränkungen einzuführen, die es nur ausgewählten Gruppen erlaubten, auf die Seiten zuzugreifen. Somit wurde für das *Comm-Sy* eine Gruppe angelegt und die Konten wurden in die Gruppe aufgenommen.

Damit hatten wir folgende Konfiguration für den ersten Server zur Verfügung:

1. Gerät: Sun Enterprise 5, 320 MB, 9 GB (Ausstattung des Arbeitsbereichs);
2. Internet-Verbindung: 10-MBit-Netzanschluss (Ausstattung des Fachbereichs);
3. Webserver: Netscape Enterprise 3.6;
4. Directory-Server: Netscape Directory-Server 4.0;
5. Script-Interpreter: PHP 3;
6. Datenbank: MySql 3.22.

## C.2 Entwicklung des *CommSy*-Projektes

Das *CommSy* ist aus der Infrastruktur-Perspektive besonders interessant mit Blick auf seine Entwicklung über die Zeit. Durch den Einsatz des Systems in unterschiedlichen Lehr-Lern-Kontexten wurde sein Einsatzgebiet immer schärfer definiert und gleichzeitig immer konkreteren Anforderungen ausgesetzt. Der Schwerpunkt der Anforderungen verschob sich ebenso mit jedem Projekt, sodass wiederholt Entscheidungen anstanden, die das Leistungsspektrum und die Flexibilität der Lernumgebung bestimmten. Die Tabelle 12 auf Seite 272 zeigt die relevanten Lehre-Projekte an der Universität Hamburg, die in direktem Zusammenhang mit der *CommSy*-Entwicklung standen. Veranstaltungen vor Herbst 1999 wurden mit anderer Software-Unterstützung bestritten und dienten als Vorlage für die *CommSy*-Entwicklung. Später wurde das *CommSy* regelmäßig eingesetzt bzw. mit anderen ähnlich ausgerichteten Systemen verglichen. Darüber hinaus gibt es mittlerweile durch den öffentlichen *CommSy*-Server bei der Firma uni.de (vergleiche Kapitel C.2.5) eine große Anzahl weiterer Projekte, die sich allerdings unserer direkten Begleitung und Auswertung entziehen. Von den Projekten an der Universität Hamburg werden im Folgenden einige ausgewählt beschrieben.

### C.2.1 KnowNet

KnowNet war eine Arbeitsgruppe im Kontext des Arbeitsbereichs ASI, die mit Software des Arbeitsbereichs SWT arbeitete: Die Konten zur Zugriffssteuerung wurden für alle Teilnehmenden im Webserver verwaltet, der zusammen mit der Kontenverwaltung auf einem Rechner des Arbeitsbereichs SWT lief. Ein erstes technisches Problem, das sich stellte, war die von dem Professor geforderte Verschiebung des Webbereichs unter der Adresse `www.knownet.de`. Dies war nicht möglich, da der gewählte Dienstanbieter zu diesem Zeitpunkt keine dynamisch generierbaren Webseiten unterstützte. Wir behelfen uns mit einer Lösung, bei der die Anwender von dort auf eine Universitätsadresse umgeleitet wurden.

Die (Weiter-)Entwicklung des Systems war schwierig, da wir an organisatorische Grenzen stießen. Die Studierenden hatten ihre Unix-Konten in Unix-Projekte beim Arbeitsbereich ASI eingetragen, der Webserver lief allerdings beim Arbeitsbereich SWT und nutzte dort Verzeichnisse im Unix-Filesystem. Studierende und Mitarbeiter von ASI konnten als Mit-

glied ihrer Unix-Projekte weder auf das Dateisystem zugreifen noch an den Rechnern des Arbeitsbereichs SWT arbeiten. Die Organisationsform des Fachbereichs sah es nicht vor, dass Lehre-Projekte zwischen Arbeitsbereichen gemeinsame Ressourcen nutzten, die über Unix-Gruppen organisiert wurden. Die Beteiligten behelfen sich damit, ein weiteres Projekt (d. h. eine Unix-Gruppe) zu beantragen, das bei SWT gültig war und somit allen den Zugriff ermöglichte. Dies führte regelmäßig zu Verwirrung bei anderen Fachbereichsangehörigen, da ASI-Lehrveranstaltungen mit SWT-Projekten liefen.

#### *Problem C.1 (Entwicklung – Zugriff über Abteilungsgrenzen)*

Die Trennung des Fachbereichs in Arbeitsbereiche und die damit verbundenen Grenzen setzten dem Projekt wiederholt organisatorische Schranken, die erkannt und umgangen werden mussten.

### **C.2.2 Projektseminar im Wintersemester 1999/2000**

Das Projektseminar „Intranets, virtuelle Gemeinschaften und Knowledge Networks“ im Wintersemester 1999/2000 fand am Arbeitsbereich ASI unter der Leitung von Arno Rolf und Bernd Pape statt. Es wird im Folgenden kurz als PJS ASI bezeichnet. Etwa 25 Studierende nahmen daran teil und arbeiteten in 5 Projektgruppen, die sich je einem speziellen Thema widmeten. Plenarsitzungen und Gruppenarbeit wechselten dabei einander ab. In den Plenarsitzungen wurde Raum für allgemeine Informationen zum Thema sowie für Präsentationen gegeben. Die Gruppenarbeit diente zur freien Arbeitsgestaltung innerhalb der fünf Themengruppen und wurde von den Teilnehmenden selbstständig geplant. Zur Unterstützung der Koordination und Kommunikation wurde das CommSy eingesetzt. Darüber hinaus stand eine E-Mail-Liste zur Verfügung.

Das PJS ASI war wiederum eine ASI-Veranstaltung, bei der die notwendigen Konten zur Nutzung des CommSys bei SWT verwaltet wurden. Der Festplattenspeicherplatz im für das CommSy notwendigen Unix-Bereich musste „quer“ zu den Organisationsgrenzen beantragt werden. Die Veranstalter mussten wiederholt den Administrator von SWT und mich bitten, bestimmte Konfigurationen durchzuführen, ohne dass sie direkt organisatorisch etwas mit der Lehrveranstaltung zu tun hatten.

*Problem C.2 (CommSy – Konten anlegen)*

In den ersten Versionen des Community Systems wurden die Konten für den Zugriff „von Hand“ mit der vom Directory-Server bereitgestellten Oberfläche erzeugt und danach in die Gruppe, die dem betreffenden *CommSy* zugeordnet war, eingetragen. Die relevante Gruppe war an einer Namenskonvention zu erkennen. Beide Tätigkeiten erfordern Kontextwissen und gewisse Berechtigungen, die nur wenige Personen hatten.

*Problem C.3 (CommSy – Konten freischalten)*

Für den Einsatz in der ersten Lehrveranstaltung war die Kontengenerierung bereits insofern automatisiert, als dass die Studierenden und andere Teilnehmende sich durch Ausfüllen einer Webseite ein Konto selber anlegen konnten. Das Freischalten – die Zuordnung des Kontos zu einer Gruppe – musste allerdings immer noch „von Hand“ passieren. Es gab nur wenige Personen, die dafür die Berechtigung besaßen.

Neben der *CommSy*-Nutzung wurde ergänzend ein so genannter „freier Webbereich“<sup>3</sup> verwendet, der in vorangegangenen Lehrveranstaltungen üblich war. Es war den Lehrenden von Anfang an klar, dass das *CommSy* nicht alle technischen kommunikativen Möglichkeiten abdecken würde. Diese könnten besonders gut durch einen freien Webbereich ausgeglichen werden. Der Webbereich wurde von den Teilnehmenden während des Projektseminars zum Austausch von Zwischenergebnissen verwendet und am Schluss zur Präsentation der Endergebnisse.

Bereits die erste Lehrveranstaltung, in der das *CommSy* eingesetzt wurde, ist empirisch ausgewertet worden. Die Studierenden haben in begleitenden Interviews neben positiven Aussagen über die innovative technische Unterstützung angemerkt, dass das System seine Grenzen besitzt und gegebenenfalls andere Kommunikationsmittel (Telefon, E-Mail etc.) verwendet werden müssten (Medienkompetenz). Die Eignung für Kommunikation wurde unterstrichen, die für Dokumentation angezweifelt. Insbesondere wurden Probleme in der Nutzung der Diskussionsforen konstatiert (vergleiche Vorfall 4.4.2):

---

3 Ein Bereich auf dem Dateisystem, der über den Webserver zugreifbar war. Der Bereich war für alle Teilnehmenden des Seminars zugreifbar. Alle Dateien, die hier abgelegt wurden, konnten im Web abgerufen werden.



*Problem C.4 (Diskussionsforen – langsame Erzeugung)*

Aufgrund schlechter Programmierung war die Generierung der Webseiten von Diskussionsforen besonders langsam. Für jeden Beitrag wurden mehrere einzelne Abfragen an der Datenbank durchgeführt (Beleg Interview JE-5-36).

*Problem C.5 (Diskussionsforen – lange Ladezeiten)*

Lange Diskussionen mit vielen Beiträgen erfordern die Übertragung von umfangreichen Webseiten. Einige Webbrowser können Tabellen, die hier zum Layout der Diskussionsforen verwendet wurden, aber erst anzeigen, wenn sie vollständig empfangen wurden. Dies führte zu Beschwerden der Anwender (Beleg Interview KR-x-32).

In einer mehrstündigen Sitzung mit dem Programmierer der Diskussionsforen wurde der Algorithmus zur Präsentation eines Forums analysiert und die damit verbundenen Datenbankabfragen untersucht. Eine Lösung bestand in drei gleichzeitigen Änderungen. Zur Verbesserung des Laufzeitverhaltens wurden die Datenbankzugriffe optimiert. Zur Minderung der übertragenen Informationen wurden für die Formatierungen Cascading Style Sheets verwendet, um für die in jeder Zeile notwendigen Formatangaben weniger Zeichen zu verbrauchen. Gleichzeitig wurde die Tabelle in mehrere Tabellen zerlegt, damit der Seitenaufbau sukzessive und damit als subjektiv schneller wahrgenommen werden konnte.

Im Vordergrund der Projekterfahrungen stand hier die Einführung technischer Unterstützung in einem bereits etablierten Lehr-Lern-Kontext. Der zusätzliche Aufwand für das Erlernen der neuen „Lern-Infrastruktur“ wurde kritisch beurteilt und im Kontext der gewünschten Lernerfolge eingeordnet:

*Erlernbarkeit*

Die Lehr-Lern-Infrastruktur muss in Bezug auf ihre *Erlernbarkeit* bewertet werden, d. h.: Wie leicht können Neulinge sich die Infrastruktur erschließen, um sie für ihre Zwecke einzusetzen? Damit der Lernaufwand möglichst gering gehalten werden konnte und die Lerninhalte nur durch Elemente ergänzt wurden, die zu den spezifischen Inhalten passten, wurde das CommSy dementsprechend gestaltet.

### *Kommunizierbarkeit*

Das *CommSy* unterstützt das Sprechen über das *CommSy* selbst, indem es sich jedem Benutzer in gleicher Weise darstellt. Für die *Kommunizierbarkeit* wird somit eine gemeinsame Basis gelegt.

### **C.2.3 CommSy-CommSy – ein CommSy für die Entwickler**

Das CommSy-CommSy stellt ein außergewöhnliches Projekt dar, welches im Anschluss an die Lehrveranstaltung PJS ASI entstand. Unter den Mitarbeitern und Studierenden wurde ein großes Potenzial in der Weiterentwicklung des Systems gesehen und dadurch die Bereitschaft signalisiert, außerhalb von Lehrveranstaltungen Zeit und Arbeit in das System zu investieren. Die Gemeinschaft organisierte sich selbst mit einem *CommSy*, wodurch der Name der Gruppe motiviert ist.

Insgesamt haben mehr als 25 Studierende und Mitarbeiter an der Gruppe teilgenommen und sich unterschiedlich engagiert. In unregelmäßigen Abständen wurden Treffen organisiert, Lehrveranstaltungen beraten und Workshops veranstaltet. Zwischen diesen Terminen koordinierten sich die Beteiligten mithilfe des *CommSys*. Diese Aktivität wird z. B. durch die insgesamt 27 Diskussionsforen mit mehreren tausend Einträgen dokumentiert.

### *Übertragbarkeit*

Der ursprüngliche Einsatz des *CommSy* war in einer Gruppe für Knowledge Management. Der weitere Einsatz fand in einer Lehrveranstaltung statt. In der Gruppe CommSy-CommSy wurde das System erneut für eine extracurriculare Arbeitsgruppe verwendet. Arbeitsschwerpunkte der Gruppe waren eine fachliche und technische Konsolidierung. Dies ging einher mit Reflexionen über Benutzbarkeit und die Übertragung auf unterschiedliche Anwendungskontexte: Damit wurde die Eigenschaft *Übertragbarkeit* auf unterschiedlichen Ebenen für die Lehr-Lern-Infrastruktur erkundet.

### **C.2.4 Internationale Frauenuniversität – der CommSy-Einsatz bei der ifu**

Bei der *Internationalen Frauenuniversität* stellte das *CommSy* einen Bestandteil der medialen Lernumgebung dar. Ziel war die Bereitstellung einer Lernumgebung, die zeitlich und örtlich außerhalb von Hamburg erreichbar bleibt. Dabei bot das *CommSy* für die *ifu* zwar eine Basis, indem es

die oben beschriebenen Unterstützungen für die Projektarbeit zur Verfügung stellte, bedurfte aber mehrerer essentiell notwendiger Weiterentwicklungen, die hier beschrieben werden sollen. Somit löste der Einsatz einen Entwicklungsschub aus. Notwendige Erweiterung war erstens die Mehrsprachigkeit und insbesondere die Unterstützung der englischen Sprache, um überhaupt in einem internationalen Kontext eingesetzt werden zu können. Zweitens war die Anbindung an eine Kontenverwaltung (hier konkret der Directory-Server in Berlin) notwendig, um neben dem Komfort für die Studentinnen auch eine Vereinfachung in der Verwaltung durchzuführen. Die dritte Weiterentwicklung stellt die Bereitstellung mehrerer Projekträume mit einer Installation sowie einer Datenbank für mehrere Projekträume dar, die ebenfalls für die Verwaltung und Administration der virtuellen Projekträume essentiell war.

Im Rahmen von Studien- und Diplomarbeiten wurden darüber hinaus die Workspaces entwickelt, die zur Präsentation von Endergebnissen gedacht waren, und die Quellenverwaltung verbessert (Jeenicke 2001), sodass Teilnehmende Quellen unterschiedlichen Typs in angemessener Art erfassen und sortieren können.

Als eine Anforderung aus der *ifu* heraus ergab sich der Wunsch, zwischen verschiedenen *CommSy* Dokumente austauschen zu können. Diese Anforderung konnte im Rahmen des Projektes nicht umgesetzt werden und wurde in das Forschungsprojekt *Wisspro* weitergegeben.

### *Übertragbarkeit*

In Bezug auf *Übertragbarkeit* hat dieser Anwendungskontext gezeigt, dass der Infrastruktur-Bestandteil *CommSy* auf einen weiteren Kontext übertragen werden konnte. Durch diese Übertragung sind neue Anforderungen artikuliert worden, die in die Infrastruktur-Entwicklung zurückgeflossen sind.

## **C.2.5 *CommSy*@uni.de – ein öffentlicher *CommSy*-Server**

Die Firma uni.de AG ist im Laufe des Jahres 2000 mit der Frage an die Entwickler des *CommSy* herangetreten, ob es möglich ist, das *CommSy* auf ihrem öffentlichen Server zu betreiben. Sie suchten zur Stärkung ihrer fachlichen Orientierung ein Produkt mit entsprechender Ausrichtung. Die Herausforderung, das *CommSy* als Produkt öffentlich anzubieten, wurde von dem *CommSy*-Team angenommen.

Tabelle 9: Fachliche Arbeitspakete

| Paket  | Titel   | Aufwand |
|--------|---|---------|
| FAP01* | Erster Kontakt                                  | 34 PT   |
| FAP02  | Öffentliches <i>CommSy</i>                      | 29 PT   |
| FAP03  | Absichtserklärung                               | 3 PT    |
| FAP04* | Einsatzvorbereitungen                           | 6 PT    |
| FAP05* | Einführung in die Benutzung und ihre Begleitung | 17 PT   |
| FAP06† | Meta- <i>CommSy</i>                             | 28 PT   |
| FAP07† | Evaluation der <i>CommSy</i> -Nutzung           | 20 PT   |
| FAP08  | Beratervertrag                                  | 20 PT   |

\* – ein Auftrag hierfür wurde erteilt

† – ohne einen Auftrag durchgeführt

Die bisherige Entwicklung des *CommSys* war im Rahmen von studentischen Projekten, Lehrveranstaltungen und Freizeitaktivitäten geschehen. Für den öffentlichen Betrieb fehlten technische und organisatorische Voraussetzungen.

Aus der Erfahrung mit unterschiedlichen Einsatzkontexten wurden in der Art Konsequenzen gezogen, als dass systematisch nach zusätzlichen Anforderungen gesucht wurde. In den Verhandlungen wurde die Möglichkeit genutzt, fachliche und technische Arbeitspakete zu definieren. Dadurch gewann das Entwicklerteam einen Überblick über die Entwicklungen, die essentiell durchzuführen waren, und konnte gleichzeitig die Aufgaben näher benennen, die vertraglich diskutiert werden mussten. Die Tabelle 9 und die Tabelle 10 auf der nächsten Seite geben einen Überblick.<sup>4</sup> Man einigte sich auf ausgewählte Arbeitspakete und setzte diese im universitären Kontext in kleinen Arbeitsprojekten um.

Ein Großteil der „Reifung“ des *CommSys* als Infrastrukturbestandteil bestand in diesem Projekt darin, dass die Administration und die Nutzung vollständig aus den Händen der initial Beteiligten gegeben wurden. Sowohl die Administration für den Betreiber, die Firma uni.de, als auch für den

4 Die mit einem Stern gekennzeichneten Arbeitspakete wurden im Rahmen der vertraglichen Zusammenarbeit bearbeitet. Die mit † markierten, wurden kostenlos bzw. im Rahmen des Beratervertrags durchgeführt. Alle mit ¶ markierten Arbeitspakete wurden in anderen Projekten (z. B. *ifu*) bearbeitet.

Tabelle 10: Technische Arbeitspakete

| <b>Paket</b> | <b>Titel</b>                          | <b>Aufwand</b> |
|--------------|---------------------------------------|----------------|
| TAP01        | Mailanbindung                         | 15 PT          |
| TAP02*       | Userverwaltung                        | 10 PT          |
| TAP03        | Sourcecode Management                 | 17 PT          |
| TAP04*       | Speicherung in <i>einer</i> Datenbank | 6 PT           |
| TAP05¶       | Nutzung <i>eines</i> Code-Baumes      | 6 PT           |
| TAP06        | Templates                             | 30 PT          |
| TAP07        | Sicherheitskonzept                    | 4 PT           |
| TAP08        | Quota für Projekträume                | 6 PT           |
| TAP09*       | Konfiguration in DB                   | 5 PT           |
| TAP10        | Verschiedene Raumtypen                | 8 PT           |
| TAP11*       | User/Gruppe                           | 7 PT           |
| TAP12        | Admin Level                           | 9 PT           |
| TAP13        | weitere Sprachen                      | 9 PT           |
| TAP14*       | Werbefbanner                          | 2 PT           |
| TAP15        | Brücken zu anderen Systemteilen       | 5 PT           |
| TAP16        | Mailinglisten                         | 6 PT           |
| TAP17        | Personalisierung                      | 7,5 PT         |
| TAP18        | Online-Hilfe                          | 12 PT          |
| TAP19        | Überblick über Rubriken               | 8 PT           |
| TAP20¶       | Übergreifende Diskussionsforen        | 3 PT           |
| TAP21*       | web-gestützte Installation            | 6 PT           |
| TAP22*       | web-gestützte Konfiguration           | 6 PT           |
| TAP23†       | Session Management                    | 10 PT          |
| TAP24        | CD-Abzug                              | 2 PT           |

\* – ein Auftrag hierfür wurde erteilt

† – ohne einen Auftrag durchgeführt

¶ – in anderem Projektkontext bearbeitet

Projektraumleiter, jeden einzelnen Kursleiter, musste unabhängig von den Entwicklern durchführbar sein. Für beide Bereiche wurden entsprechende Web-Schnittstellen bereitgestellt.

## C.2.6 WissPro

Das WissPro-Projekt ist die konsequente Fortführung der *CommSy*-Entwicklung. Es stellt einen institutionellen Einsatz am Fachbereich Informatik der Universität Hamburg sicher. Der Projektantrag fasst dies folgendermaßen zusammen:

„In dem Vorhaben wird das Studieren in gestaltungs- und IT-orientierten Studiengängen wie in der Weiterbildung in seiner Gesamtheit als Wissensprojekt mit interdisziplinären Anknüpfungen verstanden und unterstützt. Inhaltlich stehen die Teile des Studiums im Vordergrund, in denen es um die Wechselwirkungen zwischen Informatiksystemen und Anwendungskontexten (Menschen, Aufgaben und Organisationen) geht. Es geht um Sozial- und Medienkompetenz der Gestalter.

Zu dem angestrebten integrierten Projektstudium gehören innovative Lehr- und Lernformen und die dazu notwendige web-basierte Infrastruktur, bestehend aus vernetzten gemeinschaftsorientierten Lernumgebungen sowie multiperspektivisch erschließbaren Wissensarchiven. Die organisatorische Umsetzung erfolgt an den einzelnen Standorten im Rahmen von Studienreformbemühungen durch die Neugestaltung einzelner Lehrveranstaltungen, die Integration zwischen Lehrveranstaltung sowie begleitende Qualitätssicherungsmaßnahmen.

Die Antragsteller beabsichtigen mit dem hier vorgestellten Projekt Methoden zu entwickeln, die es den Studierenden und Lehrenden erlauben, die im Rahmen des Projektvorhabens entwickelte Infrastruktur optimal und nachhaltig zu nutzen. Integriertes, projektorientiertes Studieren kommt wegen seiner vielfältigen sozialen Bezüge und der Verzahnung von Theorie und Praxis insbesondere auch den Lehr- und Lerninteressen von Frauen entgegen.“  
(Oberquelle et al. 2001)

Aus infrastruktureller Sicht geht es dabei darum, ein Bündel von Maßnahmen zu etablieren, welches auf didaktischer, fachlicher, organisatorischer und technischer Ebene sicherstellt, dass das *CommSy* im Regelbetrieb der Lehre eingesetzt werden kann. Dazu zählt z. B. ein Hosting-Konzept, mit dem der technische Betrieb sichergestellt und eine organisatorische Form gefunden werden soll, die eine klare Aufgabenteilung gepaart mit Qualitätssicherung sicherstellt.

Das Projekt und die damit verbundenen Maßnahmen zeigen, dass aufgrund des infrastrukturellen Charakters des *CommSys* auf verschiedenen Ebenen Handlungsbedarf besteht. Diese Bedarfe stehen in engem Zusammenhang und müssen deshalb aufeinander abgestimmt werden. Im Mittel-

punkt des Projektes steht deshalb das Ziel, dem Fachbereich Informatik der Universität Hamburg nach Ablauf der Projektlaufzeit ein lauffähiges System zu übergeben, welches auf den o. g. Ebenen konsolidiert und etabliert ist.

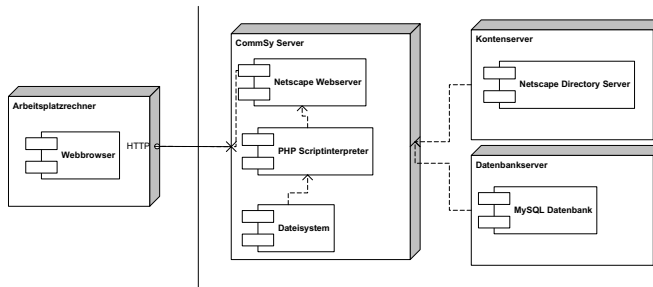
Um auf den unterschiedlichen Ebenen eine Konsolidierung und Etablierung herzustellen, werden unterschiedliche Maßnahmen gewählt. Für die Entwicklung der Anwendung wird eine Open-Source-Strategie anvisiert, um Effekte dieser Art der Entwicklung auszunutzen. Für den Betrieb der Infrastruktur wird eine Hosting-Strategie untersucht, mit der Anforderungen aus dem Betrieb angemessen erfüllt werden können. Auf der Ebene der didaktischen und fachlichen Einbettung werden bisher experimentelle Lehrveranstaltungen in das regelmäßige Curriculum aufgenommen. Fragen bezüglich der organisatorischen Einbettung sollen durch konkrete Benennung und Beschreibung von funktionellen Rollen geschehen.

### **C.3 Technische Infrastruktur**

Zum Betrieb eines *CommSy*-Projektraums ist eine Vielzahl von technischen, organisatorischen und sozialen Rahmenbedingungen zu schaffen. In diesem Abschnitt wird der Schwerpunkt auf die ersten beiden (Bleek und Pape 2001) gelegt. Wie bereits angedeutet, benötigt man einige Standardprodukte zum Betrieb. Die Abbildung 11 auf Seite 271 gibt dazu nähere Auskunft.

Kriterien für die Auswahl dieser technischen Bestandteile lassen sich aus dem Einsatzkontext ableiten. Die verwendete Software ist kostenfrei verfügbar, sodass hieraus weder für den Betrieb noch bei der Weiterentwicklung Kosten entstehen. Die Software ist auf Unix-Maschinen (Sun SPARC, Intel) genauso lauffähig wie auf Windows-Rechnern (Intel). Dadurch lässt sie sich nicht nur einfach am Fachbereich Informatik betreiben, sondern kann von den Studierenden, die in Projekten Weiterentwicklungen durchführen, an eigenen Rechnern betrieben werden.

Die Integration in bestehende technische Infrastrukturen stand und steht bei der Entwicklung des Systems in Vordergrund. Diese Integration kann auf mehreren Ebenen betrachtet werden. Zum einen muss sich die Entwicklung und der Betrieb auf bestehenden Infrastrukturen abstützen. Dies nimmt die Entscheidung für die Wahl der Hardware und des Betriebssystems vorweg. Im konkreten Einsatz muss sich die Technik ebenfalls in die gegebenen technischen Rahmenbedingungen eingliedern. Dies wird durch

Abbildung 20: Software-Architektur des *CommSys*

die Entwicklung für das Web adressiert, indem als minimale Voraussetzung zur Nutzung des Systems lediglich ein Webbrowser vorausgesetzt wird.

## C.4 Zusammenfassende Betrachtung

Die Entwicklung des *CommSy*-Systems hat als Anwendungsentwicklung für Knowledge-Management begonnen. Durch den Schwerpunkt auf einer web-basierten Schnittstelle und die Integration in bestehende technische und organisatorische Einrichtungen wurde das *CommSy* zu einem Infrastrukturbestandteil und hatte selbst Auswirkungen auf die bestehende Infrastruktur. Das System selbst stellt keine eigene Infrastruktur dar, bildet aber im Lehr-Lern-Kontext zusammen mit anderen Bestandteilen eine Lern-Infrastruktur. Darüber hinaus offenbart der Einsatz eines *CommSy*-Systems, an welchen Stellen eine bestehende technische oder organisatorische Infrastruktur gebraucht wird und angepasst werden muss.

Hierbei sind zuerst die sichtbar gewordenen Organisationsgrenzen zu nennen. Die bestehende technische Infrastruktur des Fachbereichs stützt sich auf die Gliederung in Fachbereichseinheiten ab. Damit werden umgekehrt Projekte, die über Grenzen hinweg stattfinden, signifikant behindert. Sowohl das Projekt KnowNet als auch die *CommSy*-Entwicklung haben dies gezeigt.



### C.4.1 Rückwirkungen

Die beobachteten Rückwirkungen beim Einsatz des *CommSys* können im Bereich der Lehre eingeordnet werden. Durch den erfolgreichen Einsatz in einzelnen Lehrveranstaltungen wurden andere darauf aufmerksam. Sie wollten das System ebenfalls einsetzen und haben ihre Lehre daraufhin angepasst. Insofern ist die Nutzung eines web-basierten Community Systems in der Lehre fortan selbstverständlicher geworden.

Die wohl stärkste Rückwirkung des *CommSys*-Einsatzes ist die Beantragung eines Forschungsprojektes für die institutionalisierte Fortführung des Projektes. Hier wirkt die Infrastruktur verändernd auf die Ursprungsorganisation. Aus- und Rückwirkungen betreffen dabei die bereits aufgeführten Ebenen des Curriculums, der Didaktik, der technischen und der organisatorischen Infrastruktur.

### C.4.2 Wahrgenommene Problemfelder

Im Bereich der *CommSys*-Entwicklung wurde eine signifikante Zahl von Problemfeldern wahrgenommen. In Bezug auf Infrastruktur sind hier insbesondere zu nennen: die starren technischen Grenzen der Infrastruktur des Fachbereichs sowie die dem Anwendungskontext widersprechende Personalausstattung und -ausbildung. Dies sind im Einzelnen:

- Grenzen der Ausstattung: Es gab keine dedizierten Rechner für den Betrieb des *CommSys*. Aufgrund von Verfügbarkeit wurden vorhandene Geräte benachbarter Arbeitsbereiche verwendet.
- Grenzen der technischen Infrastruktur: Die vorhandenen Geräte hatten nicht genügend Leistung, um viele *CommSys* zu betreiben. Die technischen Restriktionen durch Unix-Gruppen, die die organisatorischen Grenzen abbilden, stellten wiederholt unnötige Engpässe dar.
- Grenzen der personellen Ausstattung: Es gab und gibt derzeit keine Stellen für Personen, die das System administrieren, die Verwendung schulen, die laufende Nutzung betreuen, das System systematisch weiterentwickeln etc.

### **C.4.3 Infrastrukturelle Herausforderungen**

Im Mittelpunkt der unterschiedlichen Herausforderungen, die dieses Projekt an Infrastruktur gestellt hat, steht das Bedürfnis, die technische Infrastruktur so weit es geht in den Hintergrund zu drängen und das notwendige Maß an technischer Wahrnehmung im Bewusstsein der Beteiligten zu verdeutlichen. Dazu waren die folgenden Merkmale gefordert worden:

#### *Integrierbarkeit*

Mit Integrierbarkeit wird die Möglichkeit bezeichnet, die neue Infrastruktur (hier die Lernplattform) in die bestehende Infrastruktur eingliedern zu können. Die Herausforderung besteht sowohl bei der Gestaltung der Lernplattform als auch beim Prozess der Integration.

#### *Mandantenfähigkeit*

Eine Infrastruktur kann von einem Betreiber (hier der Projektgruppe *Comm-Sy-CommSy*) in mehreren Kontexten eingesetzt werden, wenn die Infrastruktur sich gegenüber den Kunden als eine Einheit darstellt, die abgeschlossen ist. Die Herausforderung besteht darin, durch wenig Mehraufwand die Infrastruktur in diese Lage zu versetzen.

#### *Übertragbarkeit*

Eine Infrastruktur kann von einem Kontext auf einen neuen Kontext übertragen werden. Die Herausforderung besteht darin, auf die neu erwachsenen Anforderungen einzugehen und die dafür notwendigen Änderungen in die bestehende Infrastruktur zu integrieren.

Tabelle 11: Technische Infrastruktur für das *Community System*

|                  |  |
|------------------|--|
| Betriebssystem   | Es existieren Erfahrungen mit Solaris, Linux und Windows. Solaris 2.8 auf SunSPARC-Plattform wurde initial benutzt und für die Projekte KnowNet, PJS ASI und CommSy-CommSy verwendet (siehe Tabelle 12 auf der nächsten Seite). Bei der <i>ifu</i> wurde anfangs ausschließlich auf Windows NT 4.0 SP5 auf IntelPC-Plattform gesetzt. Aus Gründen der Performanz ist schließlich aber wieder Solaris 2.8 auf SunSPARC verwendet worden. Die Anwendungen im Rahmen des WissPro-Projektes finden mit Linux (Kernel 2.3) auf IntelPC-Basis statt. |
| Datenbank        | Zur Speicherung der Daten für das <i>Community System</i> wird ein relationaler Datenbankserver verwendet. Derzeit ist ein MySQL-2.32-Server im Einsatz. Dieser wurde sowohl unter Solaris und Windows als auch unter Linux verwendet.   |
| Webserver        | Ein laufender Webserver wird zur Auslieferung der dynamisch erzeugten Seiten benötigt. Prinzipiell können alle Webserver benutzt werden. Angefangen hat das Projekt mit einem Netscape-Enterprise-3.6-Webserver (jetzt iPlanet). Derzeit wird ein Apache-Webserver der Version 1.3.19 in den meisten Projekten eingesetzt. Es gibt ebenfalls Erfahrungen mit dem Microsoft Internet Information Server (IIS); dies wird aber derzeit nicht weiterverfolgt.   |
| Skriptsprache    | Dynamische Webseiten werden mit einer Skriptsprache erstellt. Hierfür wurde PHP wegen seiner einfachen Syntax gewählt. PHP ermöglicht das Zusammensetzen von festgelegten Texten (z. B. für HTML) und Ergebnissen aus (Datenbank-)Abfragen zu Webseiten. Mittlerweile existieren Module für unterschiedliche Webserver sowie eine große Auswahl von Anschlüssen an andere Systeme (Datenbanken, E-Mail, Directory-Server usw.).  |
| Autentifizierung | Zur Anmeldung von Anwendern an ein <i>CommSy</i> stehen verschiedene Authentifizierungsmechanismen zur Verfügung. Dies ist notwendig, um die gewünschte geschlossene Benutzergruppe innerhalb eines <i>CommSys</i> herzustellen. Derzeit wird neben einem proprietären Verfahren die Verwendung des LDAP-Standards unterstützt (lightweight directory access protocol, Yeong et al. 1995, Howes et al. 1995). Dies ist notwendig für die Integration in bestehende Umgebungen.   |
| Zugriff          | Für den Zugriff auf das <i>Community System</i> wird lediglich ein Webbrowser benötigt, der Cascading Stylesheets (CSS) unterstützt. Dies ist mittlerweile bei allen Browsern der Fall, sodass es keinerlei Einschränkung bedeutet. Geeignet sind z. B. Netscape Communicator ab Version 4.x oder Microsoft Internet Explorer ab Version 4.x.  |

Tabelle 12: Liste relevanter Projekte, die an der Universität Hamburg mit Software-Unterstützung gehalten wurden

| <b>Jahr</b> | <b>Projekttitel</b>   | <b>Lehrende</b>               | <b>Thema</b>                     | <b>Software</b>   |
|-------------|---|-------------------------------|----------------------------------|-------------------|
| 1997        | Organisations- und Wirtschaftsinformatik  | Rolf, Pape                    | Information Systems              | freier Webbereich |
| 1998        | Softwareunterstützung für Kooperation (Wetzel et al. 1998)                                    | Bleek, Klischewski, Wetzel    | CSCW                             | Lotus Notes       |
| 1998        | Softwareentwicklung in Organisationen (Bleek und Mack 1999)                                   | Bleek, Floyd, Mack            | Software-technik                 | freier Webbereich |
| 1998        | Objektorientierte Softwareentwicklung (Bleek et al. 1999b)                                    | Bleek, Züllighoven            | Software-technik                 | freier Webbereich |
| 1999        | Workflow-Management-Systeme   | Klischewski, Wetzel           | CSCW                             | freier Webbereich |
| 1999        | Intranets, virtuelle Organisationen, Knowledge Networks (Bleek et al. 2000, Gumm et al. 2000) | Rolf, Pape                    | Information Systems              | <i>CommSy</i>     |
| 1999        | Objektorientierte Softwareentwicklung   | Wolf, Züllighoven             | Software-technik                 | freier Webbereich |
| 2000        | <i>CommSy-CommSy</i>  | Pape, Jackewitz, Bleek et al. | <i>CommSy</i> Weiterentwicklung  | <i>CommSy</i>     |
| 2000        | Softwareentwicklung und Wissensmanagement   | Bleek, Floyd, Wulf            | Software-technik                 | <i>CommSy</i>     |
| 2000        | Schnittstellengestaltung  | Oberquelle, Janneck           | MMK                              | Swiki             |
| 2000        | Intranets und virtuelle Gemeinschaften  | Rolf, Jackewitz               | Information Systems              | <i>CommSy</i>     |
| 2000        | Objektorientierte Softwareentwicklung   | Lippert, Züllighoven          | Software-technik                 | <i>CommSy</i>     |
| 2000        | Serviceflow   | Klischewski, Wetzel           | CSCW                             | <i>CommSy</i>     |
| 2000        | Internationale Frauenuniversität, Projektbereich Information (Bleek et al. 2000)              | Floyd et al.                  | interdisziplinär                 | <i>CommSy</i>     |
| 2001        | Admina  | Najmi et al.                  | Selbstlernen                     | <i>CommSy</i>     |
| seit 2001   | WissPro   | diverse                       | verschiedene Lehrveranstaltungen | <i>CommSy</i>     |

# Literaturverzeichnis

- (Abbate 1994) Abbate, Jane: The Internet Challenge: Conflict and Compromise in Computer Networking. In: (Summerton 1994a), S. 193-210
- (Balzert 1996) Balzert, H.: *Lehrbuch der Software-Technik*. Heidelberg: Spektrum Akademischer Verlag, 1996
- (Bangemann 1997) Bangemann, Martin: *Information Infrastructure*. <http://www.ispo.cec.be/infosoc/promo/speech/geneva.html>. 8. September 1997. – Speech at Telecom Inter@ctive '97 in Geneva
- (Bansler et al. 2000) Bansler, Jørgen P.; Damsgaard, Jan; Scheepers, Rens; Havn, Erling; Thommesen, Jacob: Corporate Intranet Implementation: Managing Emergent Technologies and Organizational Practices. In: *Journal of the Association for Information Systems* 1 (2000), December, Nr. 10
- (Bateson 1972) Bateson, Gregory: *Steps to an Ecology of Mind*. New York: Ballantine, 1972
- (Büchner et al. 2001) Büchner, Heino; Zschau, Oliver; Traub, Dennis; Zahradka, Rik: *Web Content Management – Websites professionell betreiben*. Galileo Business, 2001
- (Becker 1982) Becker, Howard S.: *Art Worlds*. Berkeley, CA: University of California Press, 1982
- (Beyer und Holtzblatt 1997) Beyer, Hugh; Holtzblatt, Karen: *Contextual Design – Defining Customer-Centered Systems*. San Francisco: Morgan Kaufmann Publishers Inc., 1997
- (Beyer und Holtzblatt 1999) Beyer, Hugh; Holtzblatt, Karen: Contextual Design. In: *interactions* 6 (1999), January + February, Nr. 1, S. 32-42

- (Bijker et al. 1987) Bijker, Wiebe E. (Hrsg.); Hughes, Thomas P. (Hrsg.); Pinch, Trevor J. (Hrsg.): *The Social Construction of Technological Systems – New Directions in the Sociology and History of Technology*. Cambridge, Massachusetts: The MIT Press, 1987
- (Bijker und Law 1992) Bijker, Wiebe E. (Hrsg.); Law, J. (Hrsg.): *Shaping Technology / Building Society: Studies in Sociotechnical Change*. Cambridge, Massachusetts: The MIT Press, 1992
- (Bleek 1997) Bleek, Wolf-Gideon: *Technisch eingebettete Anwendungssysteme*, Universität Hamburg, Fachbereich Informatik, Diplomarbeit, August 1997
- (Bleek 2001) Bleek, Wolf-Gideon: *Situations in Life to Support the Use and Modeling of Municipal Information Systems*. In: (Remenyi und Bannister 2001), S. 49-60
- (Bleek 2002) Bleek, Wolf-Gideon: Lebenslagen als Unterstützung bei der Benutzung und Modellierung von städtischen Portal-Webseiten. In: Herczeg, Michael (Hrsg.); Prinz, Wolfgang (Hrsg.); Oberquelle, Horst (Hrsg.): *Mensch und Computer 2002: Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten* Bd. 2. Stuttgart: B. G. Teubner, 2002, S. 75-84
- (Bleek et al. 2000) Bleek, Wolf-Gideon; Dittrich, Yvonne; Jeenicke, Martti: *Classifications as Tools*. December 2000. – CSCW 2000 Workshop Paper on Groupware and Classification
- (Bleek und Floyd 2000) Bleek, Wolf-Gideon; Floyd, Christiane: Lebenslagen – Evaluation und Konzeption / Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik. 2000. – Forschungsbericht. unter Mithilfe von Timmy Blank
- (Bleek und Floyd 2001) Bleek, Wolf-Gideon; Floyd, Christiane: Lebenslagen – Redaktionskonzept / Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik. 2001. – Forschungsbericht
- (Bleek et al. 1999a) Bleek, Wolf-Gideon; Görtz, Thorsten; Lilienthal, Carola; Lippert, Martin; Roock, Stefan; Strunk, Wolfgang; Wolf, Henning: Interaktionsformen zur flexiblen Anbindung von Fenstersysteme-

- men / Universität Hamburg, Fachbereich Informatik. 1999 (FBI-HH-M-285/99). – Forschungsbericht
- (Bleek et al. 1999b) Bleek, Wolf-Gideon; Gryczan, Guido; Lilienthal, Carola; Lippert, Martin; Roock, Stefan; Wolf, Henning; Züllighoven, Heinz: Von anwendungsorientierter Softwareentwicklung zu anwendungsorientierten Lehrveranstaltungen – der Werkzeug & Material-Ansatz in der Lehre. In: Dreher, B. (Hrsg.); Schulz, Ch. (Hrsg.); Weber-Wulff, D. (Hrsg.): *Software Engineering im Unterricht der Hochschulen – SEUH*. Wiesbaden: Teubner, Februar 1999b, S. 9-20
- (Bleek et al. 2000) Bleek, Wolf-Gideon; Jackewitz, Iver; Janneck, Michael; Wolff, Bernd: *Community System CommSy*. <http://www.commsy.de>. 2000. – besucht am 16. September 2002
- (Bleek et al. 2003) Bleek, Wolf-Gideon; Jackewitz, Iver; Pape, Bernd: Matching Needs – Application Service Providing for Asynchronous Learning Networks. In: *Proceedings of the 36th Hawaii International Conference on System Sciences*. Big Island, Hawaii, Januar 2003
- (Bleek et al. 2001) Bleek, Wolf-Gideon; Jeenicke, Martti; Klischewski, Ralf: e-Prototyping. In: *EMISA* (2001), Nr. 1, S. 11-18
- (Bleek et al. 2002) Bleek, Wolf-Gideon; Jeenicke, Martti; Klischewski, Ralf: Framing Participatory Design Through e-Prototyping. In: Binder, Thomas (Hrsg.); Gregory, Judith (Hrsg.); Wagner, Ina (Hrsg.): *Proceedings of the Seventh Biennial Participatory Design Conference*. Malmö, 23.-25. Juni 2002, S. 300-310
- (Bleek et al. 2000) Bleek, Wolf-Gideon; Kielas, Wiebke; Malon, Katharina; Otto, Torsten; Wolff, Bernd: Vorgehen zur Einführung von Community Systemen in Lerngemeinschaften. In: (Engelien und Neumann 2000), S. 97-113
- (Bleek und Mack 1999) Bleek, Wolf-Gideon; Mack, Julian: Some Reflections about Virtual Organisations. In: Käkölä, Timo (Hrsg.): *Proceedings of the 22th Information Systems Seminar in Scandinavia* Bd. 1 IRIS, 1999, S. 135-143

- (Bleek und Pape 2001) Bleek, Wolf-Gideon; Pape, Bernd: Application Service Providing für vernetzte Projektarbeit – am Beispiel von CommSy@uni.de. In: Engelen, Martin (Hrsg.); Homann, Jens (Hrsg.): *Gemeinschaft in neuen Medien*. TU Dresden: Josef Eul Verlag, September 2001 (Telekommunikation @ Mediendienste), S. 349-371
- (Boehm 1976) Boehm, B.: Software Engineering. In: *IEEE Transactions on Computers* 25 (1976), S. 1226-1241
- (Boehm 1988) Boehm, B.: The Spiral Model of Software Development and Enhancement. In: *Computer* 21 (1988), Nr. 5, S. 61-72
- (Bowker 1994) Bowker, Geoffrey: Information Mythology and Infrastructure. In: (Bud 1994), S. 231-247
- (Bowker und Star 1994) Bowker, Geoffrey; Star, Susan L.: Knowledge and infrastructure in international information management: Problems of classification and coding. In: (Bud 1994), S. 187-213
- (Bowker 1994) Bowker, Geoffrey C.: *Science on the Run: Information Management and Industrial Geophysics at Schlumberger, 1920-1940*. Cambridge, MA: MIT Press, 1994
- (Bowker und Star 1998) Bowker, Geoffrey C.; Star, Susan L.: Building Information Infrastructures for Social Worlds – The Role of Classifications and Standards. In: Ishida, Toru (Hrsg.): *Community Computing and Support Systems*. Springer, 1998 (Lecture Notes in Computer Science 1519), S. 231-248
- (Bowker und Star 1999) Bowker, Geoffrey C.; Star, Susan L.: *Sorting Things Out – Classification and its Consequences*. Cambridge, Massachusetts; London, England: The MIT Press, 1999
- (Brockhaus 1931) Brockhaus (Hrsg.): *Der Große Brockhaus – Handbuch des Wissens in zwanzig Bänden*. Bd. Neunter Band: I – Kas. fünfzehnte, völlig neubearbeitete Auflage von Brockhaus' Konversations-Lexikon. Leipzig: F.A. Brockhaus, 1931
- (Brockhaus 1989) Brockhaus (Hrsg.): *Brockhaus-Enzyklopädie: in 24 Bd.*. Bd. 10: HERR – IS. 19. völlig neubearb. Aufl. Mannheim: F.A. Brockhaus GmbH, 1989



- (Bud 1994) Bud, L. (Hrsg.): *Information Acumen: The Understanding and Use of Knowledge in Modern Business*. London: Routledge, 1994
- (Callon et al. 1986) Callon, M.; Law, J.; Rip, A.: *Mapping the Dynamics of Science and Technology: Sociology of Science in the Real World*. Houndmills, Basingstoke: The Macmillan Press Ltd., 1986
- (Carey und The Propel Platform Team 2001) Carey, Michael; The Propel Platform Team: Towards a Scalable Infrastructure for Advanced E-Services / Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 2001. – Forschungsbericht
- (Ciborra 2000a) Ciborra, Claudio U.: A Critical Review of the Literature on the Management of Corporate Information Infrastructure. In: *From Control to Drift* (Ciborra 2000b), S. 15-40
- (Ciborra 2000b) Ciborra, Claudio U. (Hrsg.): *From Control to Drift - The Dynamics of Corporate Information Infrastructures*. 1st Edition. Oxford: Oxford University Press, 2000b.
- (Ciborra und Hanseth 1998) Ciborra, Claudio U.; Hanseth, Ole: Toward a Contingency View of Infrastructure and Knowledge: An Exploratory Study. In: Hirschheim, R. (Hrsg.); Newman, M. (Hrsg.); DeGross, J. I. (Hrsg.): *Proceedings from 19th Annual International Conference on Information Systems (ICIS)*. Helsinki, Finland, 13.-16. Dezember 1998, S. 263-272
- (Clement und Parsons 1990) Clement, Andrew; Parsons, Darrel: Work Group Knowledge Requirements for Desktop Computing. In: *IEEE* (1990), S. 84-93
- (Cockburn 2001) Cockburn, Alistair: *Agile Software Development*. Boston: Pearson Education, Inc, 2001 (The Agile Software Development Series)
- (Crispin 1994) Crispin, M.: Internet Message Access Protocol – Version 4 (IMAP4) / University of Washington. 1994 (1730). – Request for Comment (RFC). <http://ftp.isi.edu/in-notes/rfc1730.txt>
- (Crowther 1995) Crowther, Jonathan (Hrsg.): *Oxford Advanced Learner's Dictionary*. 5th Edition. Oxford University Press, 1995

- (Dahlbom 2000) Dahlbom, Bo: From Infrastructure to Networking. In: (Ciborra 2000b), S. 212-226
- (Damsgaard und Scheepers 1999) Damsgaard, Jan; Scheepers, Rens: A Stage Model of Intranet Technology Implementation and Management. In: Pries-Heje, J. (Hrsg.); Ciborra, C. (Hrsg.); Kautz, K. (Hrsg.); Valor, J. (Hrsg.); Christiaanse, E. (Hrsg.); Avison, D. (Hrsg.); Heje, C. (Hrsg.): *Proceedings of the 7th European Conference on Information Systems*. Copenhagen, Denmark: Copenhagen Business School, June 23-25 1999, S. 100-116
- (Demsetz 1993) Demsetz, H.: The Theory of the firm Revisited. In: Williamson, O. E. (Hrsg.); Winter, S. G. (Hrsg.): *The Nature of the Firm*. New York: Oxford University Press, 1993
- (Denert 1991) Denert, Ernst: *Software-Engineering: Methodische Projektentwicklung*. Berlin, Heidelberg, New York: Springer, 1991
- (Detlor 2000) Detlor, Brian: The corporate portal as information infrastructure: towards a framework for portal design. In: *International Journal of Information Management* 20 (2000), S. 91-101
- (DiB 2000) *Direktes Bürger Informations System – DiBIS*. <http://www.hamburg.de/dibis>. 2000
- (Dittrich et al. 2002) Dittrich, Yvonne (Hrsg.); Floyd, Christiane (Hrsg.); Klischewski, Ralf (Hrsg.): *Social Thinking – Software Practice*. Reading, Mass., USA: MIT Press, 2002
- (Dittrich und Lindeberg 2003) Dittrich, Yvonne; Lindeberg, Olle: Designing for Changing Work and Business Practices. In: Patel, Nandish V. (Hrsg.): *Adaptive Evolutionary Information Systems*. Idea group publishing, 2003, Kapitel VII, S. 152-171
- (Dudenredaktion 1990) Dudenredaktion, Wissenschaftlicher Rat der (Hrsg.): *Duden - Das Fremdwörterbuch*. Dudenverlag, 1990
- (Dudenredaktion 2000) Dudenredaktion, Wissenschaftlicher Rat der (Hrsg.): *Duden - Die deutsche Rechtschreibung*. 22., völlig neu bearbeitete und erweiterte Auflage. Mannheim, Leipzig, Wien, Zürich:

- Bibliographisches Institut & F.A. Brockhaus AG, 2000 (Der Duden in zwölf Bänden 1). – Auf der Grundlage der neuen amtlichen Rechtschreibregeln
- (Eco 1998) Eco, Umberto: *Wie man eine wissenschaftliche Abschlußarbeit schreibt*. 7., unveränderte Auflage der deutschen Ausgabe. Heidelberg: C.F. Müller Juristischer Verlag, 1998. – Ins Deutsche übersetzt von Walter Schick
- (Engelien und Neumann 2000) Engelien, Martin (Hrsg.); Neumann, Detlef (Hrsg.): *Virtuelle Organisation und Neue Medien*. Dresden: Josef Eul Verlag, Lohmar, Köln, 2000
- (Engelschall 2001) Engelschall, Ralf S.: *A Users Guide to URL Rewriting with the Apache Webserver*. <http://www.engelschall.com/pw/apache/rewriteguide/>. 2001. – besucht am 17. September 2001
- (Expo 2000) Expo. *Weltausstellung 2000*. <http://www.expo2000.de.2000>
- (FBI 2000) FBI: *National Infrastructure Protection Center*. <http://www.nipc.gov/>. 2000. – besucht am 5. Dezember 2001
- (Fielding et al. 1997) Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Berners-Lee, T.: *Hypertext Transfer Protocol – HTTP/1.1* / UC Irvine and DEC and MIT/LCS. 1997 (2068). – Request for Comment (RFC). <http://www.w3.org/Protocols/rfc2068/rfc2068>
- (Flick 1998) Flick, Uwe: *Qualitative Forschung – Theorie, Methoden, Anwendung in Psychologie und Sozialwissenschaften*. 3. Auflage. Reinbek: Rowohlt's Taschenbuch Verlag, 1998 (rowohlt's enzyklopädie)
- (Flick et al. 1991) Flick, Uwe; Kardoff, E. von; Keupp, H.; Rosenstiel, L. von; Wolff, S.: *Handbuch Qualitative Sozialforschung. Grundlagen, Konzepte, Methoden und Anwendungen*. München: Psychologie-Verlags-Union, 1991
- (Floyd et al. 1989) Floyd, C.; Reisin, F.-M.; Schmidt, G.: *STEPS to Software Development with Users*. In: Ghezzi, C. (Hrsg.); McDermid, J.A. (Hrsg.): *ESEC '89*. Berlin: Springer-Verlag, 1989 (Lecture Notes in Computer Science 387), S. 48-64

- (Floyd 2002) Floyd, Christiane: Developing and Embedding Autooperational Form. In: (Dittrich et al. 2002), S. 5-28
- (Floyd et al. 1999) Floyd, Christiane (Hrsg.); Dittrich, Yvonne (Hrsg.); Klischewski, Ralf (Hrsg.): *Social Thinking – Software Practice. Relating Software Development, Work and Organizational Change*. Wadern: IBFI, 1999 (Dagstuhl-Report Nr. 99361)
- (Floyd et al. 2000) Floyd, Christiane; Heinsohn, Dorit; Klein-Franke, Silvie: Information als soziale Ressource. In: (Neusel 2000), S. 111-125
- (Floyd et al. 1994) Floyd, Christiane; Krabbel, Anita; Piepenburg, Ulli: Einführung in der Softwaretechnik / Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik. Hamburg, Sommersemester 1994. – Arbeitsunterlagen zur Lehrveranstaltung
- (Floyd et al. 1997) Floyd, Christiane; Krabbel, Anita; Ratuski, Sabine; Wetzel, Ingrid: Zur Evolution der evolutionären Systementwicklung: Erfahrungen aus einem Krankenhausprojekt. In: *Informatik Spektrum* 20 (1997), Februar, Nr. 1, S. 13-20
- (Floyd und Züllighoven 1998) Floyd, Christiane; Züllighoven, Heinz: Softwaretechnik. In: Rechenberg, P. (Hrsg.); Pomberger, G. (Hrsg.): *Informatik-Handbuch*. 2. aktualisierte und erweiterte Auflage. München: Hanser, 1998, S. 763-790
- (Freytag und Ehlers 1999) Freytag, Dr. M.; Ehlers, Jan. *Bericht des Haushaltsausschusses über die Drucksache 16/2520 sowie das Ausschreibungsverfahren hamburg.de*. Drucksache 16/4809. 20. September 1999
- (Gasser und Huhns 1989) Gasser, Les (Hrsg.); Huhns, Michael N. (Hrsg.): *Distributed Artificial Intelligence*. Bd. 2. London: Pitman, 1989
- (Geertz 1983) Geertz, Clifford: *Dichte Beschreibung – Beiträge zum Verstehen kultureller Systeme*. suhrkamp taschenbuch, 1983
- (Glaser und Strauss 1967) Glaser, B.G.; Strauss, A.L.: *The Discovery of Grounded Theory – Strategies for qualitative research*. Chicago, Ill.: Aldine, 1967

- (Gluchowski 1997) Gluchowski, Peter: Das aktuelle Schlagwort: Data Warehouse. In: *Informatik-Spektrum* 20 (1997), Februar, Nr. 1, S. 48-49. – <http://link.springer.de/link/service/journals/00287/papers/7020001/7020%0048.pdf>
- (Gore 1994) Gore, Al: *Building the Information Superhighway*. <http://www.robson.org/gary/captioning/gorespeech.html> (besucht am 13.12.2001). 19. September 1994. – A speech by Vice President Al Gore on September 19, 1994
- (Grimm und Grimm 1877) Grimm, Jacob; Grimm, Wilhelm: *Deutsches Wörterbuch*. Bd. VIERTEN BANDES ZWEITE ABTHEILUNG. Leipzig: Verlag von S. Hirzel, 1877. – Bearbeitet von Moriz Heyne
- (Gumm et al. 2000) Gumm, Dorina; Orlowski, Beate; Jackewitz, Iver; Bestmann, Arne: Kulturelle Merkmale für verteilte Arbeitsgruppen. In: (Engelien und Neumann 2000), S. 23-36
- (Hanseth 2000) Hanseth, Ole: The Economics of Standards. In: (Ciborra 2000b), S. 56-70
- (Hanseth und Monteiro 1998a) Hanseth, Ole; Monteiro, Eric: Changing irreversible networks: Institutionalisation and infrastructure. In: Braa, Kristin (Hrsg.); Monteiro, Eric (Hrsg.): *Proceedings from 20th IRIS*, 1998, S. 21-40
- (Hanseth und Monteiro 1998b) Hanseth, Ole; Monteiro, Eric: *Understanding Information Infrastructure*. 27. August 1998b. – Manuscript
- (Hegfes 1919) Hegfes, Dr. Christ. A.: *Allgemeines verdeutschendes und erklärendes Fremdwörterbuch*. 20. Auflage. Scientia Nobilitas, 1919
- (Heinze und Kill 1988) Heinze, G. W.; Kill, Heinrich H.: The development of the German railroad system. In: (Mayntz und Hughes 1988), S. 105-134
- (Hewitt 1985) Hewitt, Carl: The Challenge of Open Systems. In: *BYTE* (1985), April, S. 223-241

- (Highsmith 2002) Highsmith, Jim: *Agile Software Development Ecosystems*. Boston: Addison-Wesley, 2002 (The Agile Software Development Series)
- (Highsmith III 1999) Highsmith III, James A.: *Adaptive Software Development – A Collaborative Approach to Managing Complex Systems*. New York: Dorset House Publishing, 1999
- (Howes et al. 1995) Howes, T.; Kille, S.; Yeong, W.; Robbins, C.: The String Representation of Standard Attribute Syntaxes / University of Michigan. 1995 (1778). – Request for Comment (RFC). <http://www.innosoft.com/ldapworld/rfc1778.txt>
- (Hughes 1983) Hughes, Th. P.: *Networks of Power: Electrification in Western Society 1890-1930*. Baltimore, Md.: Johns Hopkins University Press, 1983
- (Hughes 1987) Hughes, Th. P.: The Evolution of Large Technological Systems. In: (Bijker et al. 1987), S. 51-82
- (Hughes 1994) Hughes, Thomas P.: Foreword. In: (Summerton 1994a), S. ix-xi
- (ifu 2000a) ifu: *International Women's University*. <http://www.vifu.de>. 2000a. – 15. Juli – 15. Oktober 2000
- (ifu 2000b) ifu: Website of the *Project Area Information*. <http://www.ifu.uni-hamburg.de>. 2000b. – 15. Juli – 15. Oktober 2000
- (IMP 2001) *IMP – Internet Messaging Program*. <http://www.horde.org/imp>. 2001. – Written in PHP and provides webmail access to IMAP and POP3 accounts.
- (ISO 9241–10 1997) ISO 9241–10: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 10: Dialogue principles / International Organization for Standardization (ISO). <http://www.iso.ch/iso/en/CatalogueListPage.CatalogueList?ICS1=13&ICS2=1%80>, 1997 (EN ISO 9241–10). – International Standard. Ersatz für DIN 66234-8: 1988-02

- (Jacobson et al. 1999) Jacobson, Ivar; Booch, Grady; Rumbaugh, James: *The Unified Software Development Process*. First Printing. Reading, MA: Addison-Wesley, January 1999 (Object Technology Series)
- (Janneck und Bleek 2002) Janneck, Michael; Bleek, Wolf-Gideon: Project-based Learning with *CommSy*. In: Stahl, Garry (Hrsg.): *Proceedings of the Conference on Computer Supported Cooperative Learning*, 2002, S. 509-510
- (Jeenicke 2001) Jeenicke, Martti: *Antizipative Softwareentwicklung*, Universität Hamburg, Fachbereich Informatik, Diplomarbeit, 2001
- (Joerges 1988) Joerges, Bernward: Large Technical Systems: Concepts and Issues. In: (Mayntz und Hughes 1988), S. 9-36
- (Kahin und Abbate 1995) Kahin, Brian (Hrsg.); Abbate, Jane (Hrsg.): *Standards Policy for Information Infrastructure*. Cambridge, Massachusetts, and London, England: The MIT Press, 1995
- (Keil-Slawik 2000) Keil-Slawik, Reinhard: Zwischen Vision und Alltagspraxis: Anmerkungen zur Konstruktion und Nutzung typographischer Maschinen. In: Voß, G. G. (Hrsg.); Holly, W. (Hrsg.); Boehnke, K. (Hrsg.): *Neue Medien im Alltag: Begriffsbestimmungen eines interdisziplinären Forschungsfeldes*. Opladen: Leske und Budrich, 2000, S. 199-220
- (Kähler 1999) Kähler, Bettina. *Schriftliche Kleine Anfrage Betr.: Privatisierung von „Hamburg.de“*. Drucksache 16/2081. 12. Februar 1999
- (Kähler 2001) Kähler, Dr. B. *Schriftliche Kleine Anfrage Betr.: Zugang zu „Hamburg.de“*. Drucksache 16/5923. 30. April 2001
- (Kim 2000) Kim, Alfred: MyYahoo. In: *Communications of the ACM* 99 (2000), July, Nr. 77, S. 33-44
- (Klappenbach und Steinitz 1969) Klappenbach, Ruth (Hrsg.); Steinitz, Wolfgang (Hrsg.): *Wörterbuch der Deutschen Gegenwartssprache*. Bd. 3. Akademie-Verlag Berlin, 1969

- (Kling 1987) Kling, Rob: Defining the Boundaries of Computing Across Complex Organizations. In: Boland, Richard (Hrsg.); Hirschheim, Rudolf (Hrsg.): *Critical Issues in Information Systems Research*. John Wiley and Sons, 1987
- (Kling 1992) Kling, Rob: Behind the Terminal: The Critical Role of Computing Infrastructure in Effective Information Systems' Development and Use. In: Cotterman, William (Hrsg.); Senn, James (Hrsg.): *Challenges and Strategies for Research in Systems Development*. London: John Wiley, 1992, Kapitel 10, S. 153-201
- (Kling und Iacono 1989) Kling, Rob; Iacono, Suzanne: The Institutional Character of Computerized Information Systems. In: *Office: Technology and People* 5 (1989), Nr. 1, S. 7-28
- (Kling und Scacchi 1982) Kling, Rob; Scacchi, Walt: The Web of Computing: Computer Technology as Social Organization. In: Yovits, Marshall C. (Hrsg.): *Advances in Computers* Bd. 21. New York, London: Academic Press, 1982, S. 2-90
- (Klischewski 2001) Klischewski, Ralf: Infrastructure for an e-Government Process Portal. In: (Remenyi und Bannister 2001),
- (Koenemann et al. 2000) Koenemann, Jürgen; Lindner, Hans-Günter; Thomas, Christoph: Unternehmensportale: Von Suchmaschinen zum Wissensmanagement. In: *nfd Information Wissenschaft und Praxis* 51 (2000), September, Nr. 6, S. 325-334
- (Krabbel et al. 1997) Krabbel, Anita; Wetzel, Ingrid; Ratuski, Sabine: Anforderungsermittlung für Krankenhausinformationssysteme: Definition von Kernsystem und Ausbaustufen. In: Hasselbring, W. (Hrsg.): *Erfolgsfaktoren Softwaretechnik für die Entwicklung von Krankenhausinformationssystemen*. Krehl Verlag Münster, 1997, S. 1-8
- (Kubicek und Schmid 1996) Kubicek, Herbert; Schmid, Ulrich: Alltagsorientierte Informationssysteme als Medieninnovation – Konzeptionelle Überlegungen zur Erklärung der Schwierigkeiten, „Neue Medien“ und „Multimedia“ zu etablieren. In: Mayntz, Renate (Hrsg.); Meisheit, Bernd (Hrsg.): *Soziale und organisatorische Entwicklungsprozesse von*



- elektronischen Informations- und Kommunikationssystemen – Berichte aus den Verbundprojekten.* Köln: Max-Planck-Institut für Gesellschaftsforschung, April 1996 (Verbund Sozialwissenschaftliche Technikforschung Heft 17), S. 6-44
- (La Porte 1988) La Porte, Todd R.: The United States Air Traffic System: Increasing Reliability in the Midst of Rapid Growth. In: (Mayntz und Hughes 1988), S. 215-244
- (Lamnek 1989) Lamnek, Siegfried: *Qualitative Sozialforschung – Methoden und Techniken.* Bd. Band 2. Weinheim: Psychologie Verlags Union, 1989
- (Latour 1996a) Latour, Bruno: *Aramis or the Love of Technology.* Cambridge, Mass.: Harvard University Press, 1996a
- (Latour 1996b) Kap. Social Theory and the Study of Computerized Work Sites In: Latour, Bruno: *Information Technology and Changes in Organizational Work.* London: Chapman & Hall, 1996b
- (Latour 1999) Latour, Bruno: *Pandora's Hope: Essays on the Reality of Science Studies.* Cambridge, MA: Harvard University Press, 1999
- (Latour und Woolgar 1986) Latour, Bruno; Woolgar, S.: *Laboratory Life: The Construction of Scientific Facts.* Princeton, NJ: Princeton University Press, 1986
- (Lave und Wenger 1991) Lave, Jean; Wenger, Etienne: *Situated Learning: Legitimate Peripheral Participation.* Cambridge, MA: Cambridge University Press, 1991
- (Law 1992) Law, J.: Note on the Theory of the Actor-Network: Ordering, Strategy, and Heterogeneity. In: *Systems Practice* 5 (1992), Nr. 4, S. 379-393
- (Lehman 1980) Lehman, M.M.: Programs, Life Cycles, and Laws of Software Evolution. In: *Proc. of IEEE* 68 (1980), September, Nr. 9, S. 1060-1076

- (Maaß 1994a) Maaß, Susanne: *Strömungen, Leitbilder und Begrifflichkeit der Software-Ergonomie*, Universität Hamburg, Fachbereich Informatik, Habilitation, 1994a
- (Maaß 1994b) Maaß, Susanne: Transparenz – Eine zentrale Software-ergonomische Forderung / Universität Hamburg, Fachbereich Informatik. Vogt-Kölln-Str. 30, D-22527 Hamburg, 1994b (FBI-HH-B-170/94). – Bericht
- (Mathiassen 1998) Mathiassen, Lars: *Reflective Systems Development*, Aalborg University, Diss., 1998
- (Mayntz und Hughes 1988) Mayntz, Renate (Hrsg.); Hughes, Thomas P. (Hrsg.): *The Development of Large Technical Systems*. Frankfurt am Main, Boulder Colorado: Campus Verlag, Westview Press, 1988
- (Mayring 1983) Mayring, Philipp: *Qualitative Inhaltsanalyse*. Weinheim: Deutscher Studien Verlag, 1983
- (Mayring 1990) Mayring, Philipp: *Einführung in die qualitative Sozialforschung*. Weinheim: Deutscher Studien Verlag, 1990
- (McDermid 1991) McDermid, John A.: *Software Engineer's Reference Book*. Oxford: Butterworth-Heinemann, 1991
- (McMaster et al. 1998) McMaster, Tom; Vidgen, Richard T.; Wastell, David G.: Networks of Association and Due Process in IS Development. In: Larsen, Tor J. (Hrsg.); Levine, Linda (Hrsg.); DeGross, Janice I. (Hrsg.): *Information Systems: Current Issues and Future Changes*. Helsinki, Finland, December 10-13, 1998 1998, S. 341-358
- (Merriam-Webster 2001) Merriam-Webster. *Merriam-Webster Online*. <http://www.webster.com>. September 2001
- (Meyers Lexikon 1927) Meyers Lexikon (Hrsg.): *Meyers Lexikon*. Bd. Sechster Band: Hornberg – Korrektiv. Siebente Auflage in vollständiger neuer Bearbeitung. Leipzig: Bibliographisches Institut, 1927
- (Mish et al. 1998) Mish, Frederick C. (Hrsg.); Morse, John M. (Hrsg.); Stevens, Mark A. (Hrsg.): *Merriam-Webster's Collegiate Dictionary*.

- 10th Edition. Springfield, Massachusetts, U.S.A.: Merriam-Webster, 1998
- (Monteiro 2000) Monteiro, Eric: Actor-Network Theory and Information Infrastructure. In: (Ciborra 2000b), S. 71-83
- (Monteiro et al. 1994) Monteiro, Eric; Hanseth, Ole; Hatling, Morten: Developing Information Infrastructure: Standardization vs. Flexibility / University of Trondheim. 1994 (18). – Working Paper. Science, Technology and Society
- (Monteiro et al. 1996) Monteiro, Eric; Hanseth, Ole; Hatling, Morten: Developing Information Infrastructure: The Tension Between Standardizing and Flexibility. In: *Science, Technology and Human Values*, SAGE Periodicals Press 21 (1996), Nr. 4, S. 407-426
- (Monteiro und Hepsø 1998) Monteiro, Eric; Hepsø, Vidar: Diffusion of Infrastructure: Mobilization and Improvisation, 1998, S. 255-274
- (Morrison et al. 1999) Morrison, Henry; Acly, Ed; Byron, Dennis; Garone, Steve; McDonough, Brian; Murray, Gerry; Sweeney, Jacqueline; Villars, Richard: Portal Mania: Who Will Lead the Way to Convergence? / International Data Corporation. 1999 (Document Nr. 19325). – Bulletin
- (Myers und Rose 1988) Myers, J.; Rose, M.: Post Office Protocol – Version 3 (POP3) / TWG. 1988 (1081). – Request for Comment (RFC). <http://www.rfc-editor.org/rfc/rfc1081.txt>
- (Nakano 2001) Nakano, Russell: *Web Content Management – A Collaborative Approach*. First Printing. Boston: Addison-Wesley, Pearson Education, September 2001
- (Netscape Corporation 2000) Netscape Corporation: *Client Customization Kit*. <http://help.netscape.com/products/client/cck/cckcookbook/cckdoc.htm>. 2000. – besucht am 2. Februar 2000
- (Neusel 2000) Neusel, Aylâ (Hrsg.): *Schriftenreihe der Internationalen Frauenuniversität*. Bd. 1: *Die eigene Hochschule – Internationale Frauenuniversität Technik und Kultur*. Opladen: Leske + Budrich, 2000

- (Oberquelle et al. 2001) Oberquelle, Horst; Floyd, Christiane; Rolf, Arno; Herczeg, Michael; Binge, Günter; Aufenanger, Stefan; Klaeren, Herbert: Wissensprojekt „Informatiksysteme im Kontext“ – Vernetzte Lerngemeinschaften in gestaltungs- und IT-orientierten Studiengängen / Universität Hamburg, Fachbereich Informatik, Fachbereich Erziehungswissenschaft; Medizinische Universität zu Lübeck, Institut für Multimediale und Interaktive Systeme; Musikhochschule Lübeck; Eberhard Karls Universität Tübingen, Wilhelm-Schickard-Institut für Informatik. 2001. – Antrag zum Förderprogramm „Neue Medien in der Bildung“
- (OECD 1996) OECD: Information Infrastructure Policies in OECD Countries / Organisation for Economic Co-Operation and Development. Paris, Juli 1996. – Chair's Conclusion at the *Information Society and Development* Conference (May 1996). <http://www1.oecd.org/dsti/sti/it/infosoc/prod/iip.pdf> (besucht am 13.12.2001)
- (Orlikowski 1996) Orlikowski, Wanda J.: Improvising Organizational Transformation Over Time: A Situated Change Perspective. In: *Information Systems Research* 7 (1996), Nr. 1, S. 63-92
- (Orr 1996) Orr, Julian E.: *Talking about Machines – An Ethnography of a Modern Job*. ILR/Cornell Paperbacks, 1996
- (Pantelic und Nohr 2000) Pantelic, Martina; Nohr, Holger; Nohr, Holger (Hrsg.): *Data Warehousing*. Fachhochschule Stuttgart, 2000 (Arbeitspapiere Wissensmanagement Nr. 9/2000). – <http://www.hbi-stuttgart.de/nohr/KM/KmAP/DataWarehouse.pdf>. IS-SN 1616-5349 (Internet), 1616-5330 (Print)
- (Pape et al. 2002a) Pape, Bernd; Bleek, Wolf-Gideon; Jackewitz, Iver; Janneck, Michael: Software Requirements for Project-Based Learning – CommSy as an Exemplary Approach. In: *Proceedings of the 35th Hawaii International Conference on System Sciences*. Big Island, Hawaii, Januar 2002a
- (Pape et al. 2002b) Pape, Bernd; Jackewitz, Iver; Bleek, Wolf-Gideon: Benutzungsbetreuung für Softwareunterstützung in Lehr-Lern-Situationen. In: Bleek, Wolf-Gideon (Hrsg.); Krause, Detlev (Hrsg.);

- Oberquelle, Horst (Hrsg.); Pape, Bernd (Hrsg.): *Medienunterstütztes Lernen – Beiträge von der WissPro-Wintertagung 2002* WissPro, Universität Hamburg, Fachbereich Informatik, 2002b, S. 71-90
- (Parnas 1999) Kap. Forward In: Parnas, David L.: *Software Product-Line Engineering. A Family-Based Software Development Process*. Addison Wesley, July 1999, S. xiii-xv
- (Pertsch 1983) Pertsch, Dr. E. (Hrsg.): *Langenscheidts Großes Schulwörterbuch Lateinisch–Deutsch*. Erweiterte Neuausgabe. Langenscheidt, 1983
- (Platon 1994) Platon: *Sämtliche Werke*. Schleiermacher-Übersetzung. Reinbek bei Hamburg: Reclam, 1994
- (Postel und Reynolds 1985) Postel, J.; Reynolds, J.: File Transfer Protocol (FTP) / ISI. 1985 (959). – Request for Comment (RFC). <http://www.w3.org/Protocols/rfc959/Overview.html>
- (Postel 1982) Postel, Jonathan B.: Simple Mail Transfer Protocol – SMTP / Information Sciences Institute University of Southern California. 1982 (821). – Request for Comment (RFC). <http://www.faqs.org/rfcs/rfc821.html>
- (Pressestelle 2001) Pressestelle, Staatliche. *Hamburger Politik 1997 – 2001: Fakten und Ergebnisse*. <http://www.hamburg.de/fhh/behoerden/senatskanzlei/dokumentation/wichtig%edokumente.htm>. April 2001
- (Remenyi und Bannister 2001) Remenyi, Dan (Hrsg.); Bannister, Frank (Hrsg.): *European Conference on e-Government, Dublin*. Trinity College Dublin: MCIL, September 2001
- (Robinson 1993) Robinson, Mike: Design for unanticipated use ..... In: Michelis, Giorgio de (Hrsg.); Simone, Carla (Hrsg.); Schmidt, Kjeld (Hrsg.): *Proceedings of the Third European Conference on Computer Supported Cooperative Work* Bd. 3. Dordrecht, Boston, London: Kluwer Academic Publishers, 1993, S. 187-202

- (Robischon 1994) Robischon, Tobias: Transformation Through Integration: The Unification of German Telecommunications. In: (Summerton 1994a), S. 119-139
- (Rolf 1998) Rolf, Arno: *Grundlagen der Organisations- und Wirtschaftsinformatik*. Berlin, Heidelberg, New York: Springer Verlag, 1998
- (Rowling 1997) Rowling, J. K.: *Harry Potter and the Sorcerer's Stone*. Bd. 1. New York, Toronto, Auckland, Sydney, Mexico City, New Delhi, Hong Kong: Scholastic Inc., 1997
- (Röttger 2000) Röttger, Berndt. *Neuer E-Mail-Service Hamburg.de – offen wie ein Scheunentor*. Hamburger Abendblatt. 30. Juni 2000
- (Schatz 1991) Schatz, Bruce: Building an Electronic Community System. In: *Management Information Systems* 8 (1991), S. 87-107
- (Scheepers 1999) Scheepers, Rens: Key Role Players in the Initiation and Implementation of Intranet Technology. In: Ngwenyama, Ojelanki K. (Hrsg.); Introna, Lucas (Hrsg.); Myers, Michael (Hrsg.); DeGross, Jan (Hrsg.): *New Information Technologies in Organizational Processes*, Kluwer Academic Publisher, August 1999. – Proceedings of IFIP WG 8.2, S. 175-195
- (Schelhowe 2000a) Schelhowe, Heidi: *Projekt Virtual International Women's University*. <http://www.vifu.de>. 2000a. – Gestaltung der Website unter Mitarbeit von Barbara Schelkle
- (Schelhowe 2000b) Schelhowe, Heidi: Virtualität als Teil des Studienreformprojekts Internationale Frauenuniversität. In: (Neusel 2000), S. 175-192
- (Scholze-Stubenrecht und Mang 1999) Scholze-Stubenrecht, Dr. W. (Hrsg.); Mang, Dieter (Hrsg.): *Duden: Das große Wörterbuch der deutschen Sprache*. Bd. 5. Mannheim, Leipzig, Wien, Zürich: Dudenverlag, 1999
- (Schuler und Namioka 1993) Schuler, Douglas (Hrsg.); Namioka, Aki (Hrsg.): *Participatory design: principles and practices*. Hillsdale, NJ: Lawrence Erlbaum Ltd., 1993

- (Schumacher und Schwickert 1999) Schumacher, Mark; Schwickert, Axel C.: Web-Portale Stand und Entwicklungstendenzen / Lehrstuhl für allgemeine BWL und Wirtschaftsinformatik, Johannes Gutenberg Universität. Mainz, 1999 (4/99). – Arbeitspapiere WI. [http://wi.bwl.uni-mainz.de/arbeitspapiere/pdf/ap04\\_99.pdf](http://wi.bwl.uni-mainz.de/arbeitspapiere/pdf/ap04_99.pdf)
- (Senat der Freien und Hansestadt Hamburg 1999a) Senat der Freien und Hansestadt Hamburg: *Hamburg im Internet – Bestandsaufnahme, Zielsetzung, Betreibermodell*. 26. Mai 1999a.
- (Senat der Freien und Hansestadt Hamburg 1999b) Senat der Freien und Hansestadt Hamburg. *Stadtinformationssystem Hamburg – Europaweiter öffentlicher Teilnahmewettbewerb mit anschließendem Verhandlungsverfahren*. <http://www.hamburg.de/stadtinfo/wettbewerb.htm>. Juni 1999b
- (Senat der Freien und Hansestadt Hamburg 2000a) Senat der Freien und Hansestadt Hamburg. *Mitteilung des Senats an die Bürgerschaft – Stand der Verwaltungsmodernisierung in Hamburg*. 5. September 2000a
- (Senat der Freien und Hansestadt Hamburg 2000b) Senat der Freien und Hansestadt Hamburg. *Mitteilung des Senats an die Bürgerschaft – Stellungnahme des Senats zu dem Ersuchen der Bürgerschaft vom 24./25. November 1999 (Drucksache 16/3319) – Bezirke – Service für Bürgerinnen und Bürger*. 5. September 2000b
- (Simpson und Weiner 1989) Simpson, J. A. (Hrsg.); Weiner, E. S. C. (Hrsg.): *The Oxford English Dictionary*. Bd. VII. Second Edition. Oxford: Clarendon Press, 1989
- (Sommerville 2001) Sommerville, Ian: *Software Engineering*. Sixth Edition. Pearson Education Limited, England: Addison-Wesley, 2001
- (Stahlknecht 1990) Stahlknecht, Peter: Standardsoftware. In: Mertens, Peter (Hrsg.): *Lexikon der Wirtschaftsinformatik*. 2., vollst. neu bearb. und erw. Aufl. Berlin, Heidelberg, New York: Springer, 1990, S. 400-402

- (Star 1997) Star, Susan L.: Working Together: Symbolic Interactionism, Activity Theory and Information Systems. In: Engeström, Yrjö (Hrsg.); Middleton, David (Hrsg.): *Communication and Cognition at Work*. Cambridge: Cambridge University Press, 1997. – In einer Vorabversion vom 23.11.1990, S. 296-318
- (Star und Bowker 1995) Star, Susan L.; Bowker, Geoffrey C.: Work and Infrastructure. In: *Communications of the ACM* 38 (1995), Nr. 9, S. 41
- (Star und Ruhleder 1994) Star, Susan L.; Ruhleder, Karen: Steps towards an ecology of infrastructure: complex problems in design and access for large-scale collaborative systems. In: Furuta, R. (Hrsg.); Neuwirth, C. (Hrsg.): *CSCW 94*, Chapel Hill, North Carolina, October 22-26 1994, S. 253-264
- (Star und Ruhleder 1996) Star, Susan L.; Ruhleder, Karen: Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. In: *Information Systems Research* 7 (1996), Nr. 1, S. 111-134
- (Steinberg 1995) Steinberg, S. G.: Addressing the Future of the Net. In: *WIRED* (1995), May, S. 141-144
- (Strauss und Corbin 1996) Strauss, A.; Corbin, J.: *Grounded Theory: Grundlagen qualitativer Sozialforschung*. Weinheim: Beltz Psychologie Verlags Union, 1996
- (Summerton 1994a) Summerton, Jane (Hrsg.): *Changing Large Technical Systems*. Boulder, San Francisco, Oxford: Westview Press, 1994a
- (Summerton 1994b) Summerton, Jane: Introductory Essay: The Systems Approach to Technological Change. In: *Changing Large Technical Systems* (Summerton 1994a), S. 1-21
- (Tanenbaum 1996) Tanenbaum: *Computer Networks*. Third Edition. Vrije Universiteit, Amsterdam, The Netherlands: Prentice-Hall International, Inc., 1996
- (Tuomi 2001) Tuomi, Ilkka: Internet, Innovation, and Open Source: Actors in the Network. In: *First Monday* 6 (2001), January, Nr. 1. – [http://firstmonday.org/issues/issue6\\_1/tuomi/index.html](http://firstmonday.org/issues/issue6_1/tuomi/index.html)



- (Walsham 1997) Walsham, G.: Actor-Network Theory and IS Research: Current Status and Future Prospects. In: Lee, Allen S. (Hrsg.); Liebenau, Jonathan (Hrsg.); DeGross, Janice I. (Hrsg.): *Information Systems and Qualitative Research*. First edition. London, Weinheim, New York, Tokyo, Melbourne, Madras: Chapman & Hall, 1997, Kapitel 23, S. 466-480
- (Weber 1999) Weber, Herbert: *IT Infrastrukturen 2005 – Informations- und Kommunikations-Infrastrukturen als evolutionäre Systeme*. <http://www.continuous-engineering.de>. September 1999. – Fraunhofer Institut Software- und Systemtechnik (ISST)
- (Wegner 2002) Wegner, Holm: *Analyse und objektorientierter Entwurf eines integrierten Portalsystems für das Wissensmanagement*, Technische Universität Hamburg-Harburg, Dissertationsschrift, 2002
- (Weill und Broadbent 1998) Weill, P.; Broadbent, M.: *Leveraging the New Infrastructure: How Market Leaders Capitalize on Information*. Boston: Harvard Business School Press, 1998
- (Welsch 1997) Welsch, Wolfgang: *Unsere Postmoderne Moderne*. 5. Auflage. Akademie Verlag, 1997
- (Wenger 1998) Wenger, Etienne: *Communities of Practice – Learning, Meaning, and Identity*. Cambridge, UK: Cambridge University Press, 1998
- (Wetzel et al. 1998) Wetzel, Ingrid; Klischewski, Ralf; Krabbel, Anita; Lienthal, Carola: Kooperation für Software für Kooperation: Erfahrungen aus einem partizipativen Softwaretechnikprojekt. In: Claus, V. (Hrsg.): *Fachtagung Informatik und Ausbildung*. Stuttgart: Springer Verlag, 1998 (Informatik aktuell), S. 73-81
- (White House 2000) White House, The: *Information Infrastructure Task Force*. <http://iitf.doc.gov/>. 8. Januar 2000. – besucht am 5. Dezember 2001
- (Wixon et al. 1990) Wixon, Dennis; Holtzblatt, Karen; Knox, Stephen: Contextual Design: An Emergent View of System Design. In: Chew,

- J. (Hrsg.); Whiteside, J. (Hrsg.): *CHI'90 Proceedings*. New York, NY, USA: ACM Press, April 1990, S. 329-336
- (Yeong et al. 1995) Yeong, W.; Howes, T.; Kille, S.: Lightweight Directory Access Protocol (LDAP) / University of Michigan. 1995 (1777). – Request for Comment (RFC). <http://www.innosoft.com/ldapworld/rfc1777.txt>
- (Züllighoven 1998) Züllighoven, Heinz: *Das objektorientierte Konstruktionshandbuch – nach dem Werkzeug & Material-Ansatz*. 1. Auflage. d.punkt Verlag, 1998
- (Zschau et al. 2002) Zschau, Oliver; Traub, Dennis; Zahradka, Rik: *Web Content Management – Websites professionell planen und betreiben*. 2., überarbeitete und erweiterte Auflage. Galileo Business, 2002. – siehe auch (Büchner et al. 2001)
- (Zuckerer et al. 1999) Zuckerer, Walter; Stapelfeldt, Dr. D.; Riecken, Jan P.; Bestmann, Tanja; Cords, Ingrid. *Große Anfrage Betr.: Hamburg im Internet*. Drucksache 16/2520. 15. Juni 1999