

Johannes Kepler Universität Linz

Innovation in Strategischen Netzwerkorganisationen an Hand von Strategischen Unternehmungsnetzwerken und Open-Source-Software-Systemen

Diplomarbeit zur Erlangung des akademischen Grades eines Magisters
der Sozial- und Wirtschaftswissenschaften

eingereicht am Institut für Unternehmensführung: Forschungsschwerpunkt
Organisation

Betreuer: a. Univ.-Prof. Mag. Dr. Robert Bauer

Linz, August 2003

Leonhard Dobusch
Teistlergutstraße 1a
4040 Linz
Matr. Nr.: 9851296

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Leonhard Dobusch

Linz, August 2003

Abstract

Strategische Unternehmungsnetzwerke und Open-Source-Software-Systeme lassen sich unter den gemeinsamen Typus der Strategischen Netzwerkorganisation subsumieren. Neben den Vorteilen dieser Strategischen Netzwerkorganisationen im Bereich der Komplexitätsbewältigung gegenüber herkömmlichen Formen der Organisation des Leistungserstellungsprozesses ergeben sich besondere Vorteile im Bereich des Innovationsprozesses. Und hier wiederum insbesondere für den Fall, dass in der strategischen Netzwerkorganisation ein Wissensaustausch auf horizontaler Ebene der Wertschöpfungskette ermöglicht wird, wie das teilweise in Strategischen Unternehmungsnetzwerken, immer jedoch im Bereich der Open-Source-Software-Systeme der Fall ist.

Inhaltsverzeichnis

EIDESSTATTLICHE ERKLÄRUNG	2
ABSTRACT.....	3
INHALTSVERZEICHNIS.....	4
TABELLENVERZEICHNIS.....	5
1 EINLEITUNG	6
1.1 IDEE	6
1.2 ÜBERSICHT	6
2 DIE STRATEGISCHE NETZWERKORGANISATION.....	8
2.1 STRATEGISCHE NETZWERKORGANISATION ALS META-ORGANISATIONSFORM.....	8
2.2 CHARAKTERISTIKA VON NETZWERKTEILNEHMERN (KNOTEN)	9
2.3 CHARAKTERISTIKA DER NETZWERKBEZIEHUNGEN (KANTEN).....	9
2.4 CHARAKTERISTIKA DES GESAMTNETZWERKS	10
2.4.1 <i>Komplexitätsbewältigung durch Selbstorganisation.....</i>	<i>10</i>
2.4.2 <i>Verschiedene Grade der Zentralität.....</i>	<i>12</i>
3 STRATEGISCHE UNTERNEHMUNGSNETZWERKE	14
3.1 BEGRIFFSEXPLIKATION.....	14
3.1.1 <i>Knoten</i>	<i>14</i>
3.1.2 <i>Kanten</i>	<i>14</i>
3.1.3 <i>Gesamtnetzwerk</i>	<i>15</i>
3.2 ABGRENZUNG ZU MARKTLICH UND HIERARCHISCH ORGANISIERTER LEISTUNGSERSTELLUNG	16
3.2.1 <i>Komplexität und Modularität von Produkt und Prozess.....</i>	<i>16</i>
3.2.2 <i>Grenzen klassischer organisationaler Eingliederung (Integration).....</i>	<i>17</i>
3.2.3 <i>Grenzen klassischer zwischenbetrieblicher Arbeitsteilung (Outsourcing).....</i>	<i>18</i>
3.2.4 <i>Strategische Unternehmungsnetzwerke im Bereich klassischer Koordinationsdilemmata.....</i>	<i>19</i>
3.3 EMPIRISCHE BEISPIELE FÜR STRATEGISCHE UNTERNEHMUNGSNETZWERKE.....	22
4 OPEN-SOURCE-SOFTWARE-SYSTEME.....	23
4.1 DEFINITION UND BESCHREIBUNG.....	23
4.1.1 <i>Allgemeines und historische Entwicklung</i>	<i>23</i>
4.1.2 <i>Knoten</i>	<i>24</i>
4.1.3 <i>Kanten</i>	<i>26</i>
4.1.4 <i>Gesamtnetzwerk</i>	<i>28</i>
4.2 ABGRENZUNG ZU HERKÖMMLICH-PROPRIETÄRER SOFTWAREPRODUKTION	29
4.2.1 <i>Komplexität und Modularität von Produkt und Prozess.....</i>	<i>29</i>
4.2.2 <i>Grenzen klassischer Softwareproduktion</i>	<i>30</i>
4.2.3 <i>Open-Source-Software-Systeme als alternative Form der Softwareprojekt- Koordination.....</i>	<i>33</i>
4.3 EMPIRISCHE BEISPIELE FÜR OPEN-SOURCE-SOFTWARE-SYSTEME	35
5 STRATEGISCHE NETZWERKORGANISATION UND INNOVATION.....	36

5.1 INNOVATIONSPROZESSE IN STRATEGISCHEN UNTERNEHMUNGSNETZ-WERKEN.....	36
5.1.1 <i>Parallelität des Innovationsprozesses in Strategischen Unternehmensnetzwerken</i>	36
5.1.2 <i>Redundantes Wissen als Grundlage für Innovationsvorteile in Strategischen Unternehmensnetzwerken</i>	37
5.1.3 <i>Freigabe und Austausch von Wissen in Strategischen Unternehmensnetzwerken</i> .	39
5.1.4 <i>Ebenen des Wissensaustauschs im Strategischen Unternehmensnetzwerk</i>	41
5.2 INNOVATIONSPROZESSE IN OPEN-SOURCE-SOFTWARE-SYSTEMEN	42
5.2.1 <i>Koordinationskapazität als Innovationsvorteil</i>	42
5.2.2 <i>Offener Quellcode als Basis für Innovationsvorteile von Open-Source-Software-Systemen</i>	43
5.2.3 <i>Freigabe und Austausch von Wissen in Open-Source-Software Systemen</i>	44
5.2.4 <i>Ebenen des Wissensaustauschs in Open-Source-Software-Systemen</i>	46
5.3 HORIZONTALER WISSENSAUSTAUSCH ALS EIN WESENTLICHER GRUND FÜR INNOVATIONSVORTEILE IN STRATEGISCHEN NETZWERKORGANISATIONEN	46
LITERATUR	48

Tabellenverzeichnis

TAB.1: UNTERSCHIEDE UND GEMEINSAMKEITEN IN DEN ORGANISATIONSMERKMALEN UND KOORDINATIONSMECHANISMEN DER OPEN-SOURCE-SOFTWARE- UND KLASSISCHER SOFTWARE-ENTWICKLUNG (NACH SCHNEIDEWIND ET. AL. 2002, S 227)	30
TAB. 2: MOTIVATIONEN FÜR BEITRÄGE AN OPEN-SOURCE-SOFTWARE-PROJEKTE (NACH SCHNEIDEWIND ET AL. 2002, S 229)	45

1 Einleitung

1.1 Idee

Ausgangspunkt dieser Arbeit ist die besondere Fähigkeit von Strategischen Netzwerkorganisationen zur Komplexitätsbewältigung in Verbindung mit netzwerkinternem Wissensaustausch. Die Wahl von Strategischen Unternehmungsnetzwerken und Open-Source-Software-Systemen als Analyseebenen und der spezielle Fokus auf Innovationsprozesse in Strategischen Netzwerkorganisationen resultiert aus zahlreichen offensichtlichen Parallelen zwischen den zwei wohl prominentesten empirischen Beispielen der beiden Netzwerktypen:

Auf Seiten der Strategischen Unternehmungsnetzwerke ist dies der von Dyer/Nobeoka (2000) insbesondere unter Berücksichtigung des Wissensaustauschs beleuchtete Fall des „Toyota Knowledge-Sharing Network“. Aber schon vor Dyer/Nobeoka haben zahlreiche Autoren auf das Toyota-Beispiel im Zusammenhang mit Strategischen Unternehmungsnetzwerken verwiesen (vgl. u.a. Sydow 1992, S 40 / Jarillo 1993, S 117).

Auf Seiten der Open-Source-Software-Systeme ist dies „Linux“, als wohl bekanntestes und größtes Beispiel für ein Open-Source-Software-Projekt. Und auch Linux hat schon Eingang in organisationstheoretische Forschung (u.a. Morner 2002) gefunden und wurde beispielsweise von Gulati et al. (2000) im Zusammenhang mit Strategischen Netzwerkorganisationen erwähnt.

Beide Beispiele weisen nun zahlreiche Gemeinsamkeiten auf, die es rechtfertigen, sie als empirische Ausprägungen demselben organisationstheoretischen Konstrukt, nämlich der Strategischen Netzwerkorganisation, zuzuordnen. Die interessantesten Erkenntnisse lassen sich aber bei Betrachtung der Unterschiede zwischen eben diesen beiden verschiedenen Formen von Strategischen Netzwerkorganisationen gewinnen und hier wiederum im Bereich des Innovationsprozesses: So ist es Ausmaß und Gestaltung des Wissensaustauschs zwischen den einzelnen Netzwerkmitgliedern, insbesondere ob vertikal oder horizontal bezüglich der Wertschöpfungskette, der wesentlichen Einfluss auf Geschwindigkeit und Qualität des Innovationsprozesses hat.

1.2 Übersicht

Am Beginn der Arbeit steht in der Folge die allgemeine und abstrakte Beschreibung und Definition des organisationstheoretischen Konstrukts „Strategische Netzwerkorganisation“.

Anhand der klassischen Bestandteile von Netzwerken – Knoten, Kanten und Gesamtnetzwerk – werden die besonderen Eigenschaften Strategischer Netzwerkorganisationen beschrieben.

Die folgenden beiden Kapitel beschäftigen sich mit den beiden Ausprägungen Strategisches Unternehmungsnetzwerk und Open-Source-Software-Systeme. Wobei, neben der einleitenden Beschreibung nach denselben Kriterien, mit denen ersten Kapitel Strategische Netzwerkorganisationen im allgemeinen charakterisiert wurden, der Vergleich zu den jeweils herkömmlichen Organisationsformen des Leistungserstellungsprozesses (integrierte Unternehmung und Funktionsexternalisierung auf Seiten der Unternehmungsnetzwerke bzw. proprietäre Closed-Source-Softwareproduktion) im Vordergrund steht.

Im fünften und letzten Kapitel stehen dann Wissensaustausch innerhalb der beiden Netzwerktypen und die damit verknüpften Innovationsprozesse im Zentrum. So wird die Bedeutung des Wissensaustauschs (und insbesondere dessen verschiedene Ebenen entlang der Wertschöpfungskette) für innovative Prozesse erläutert und es werden die strukturellen Voraussetzungen für sein Funktionieren beleuchtet.

2 Die strategische Netzwerkorganisation

Die Erbringung einer integrierten Gesamtleistung in einem arbeitsteiligen Prozess setzt grundsätzlich noch keine strategische Organisationsform voraus. Die Verbindung einzelner Teilleistungen bedarf aber jedenfalls einer Koordination(sform), die die Komplexität des Leistungserstellungsprozesses bewältigt.

Als gegenpolige Koordinationsformen in der klassischen Organisationsforschung finden sich hier einerseits Koordination von Leistungen im Rahmen einer integrierten Organisation mit innerbetrieblicher Arbeitsteilung oder andererseits Koordination von Leistungen über Märkte mit zwischenbetrieblicher Arbeitsteilung. Während einer integrierten Organisation eine strategische Komponente – zumindest in Form eines Organisationsziels - immanent ist, fehlt sie notwendigerweise „dem Markt“. Dessen „Ziel“ ist nur ein Gleichgewicht zwischen Angebot und Nachfrage.

2.1 Strategische Netzwerkorganisation als Meta-Organisationsform

Strategische Netzwerkorganisationen lassen sich nun keiner dieser beiden gegenpoligen Koordinationsformen eindeutig zuordnen. Für den Fall der Strategischen Unternehmungsnetzwerke zeigt dies schon die Diktion Sydows (1992), der von „Quasi-Externalisierung“ auf der einen Seite und „Quasi-Internalisierung“ auf der anderen Seite als Phänomene ein und derselben Organisationsform spricht.

Die Koordination (zumindest formal, in den verschiedenen Typen Strategischer Netzwerkorganisationen aber im zunehmenden Maße auch real) wirtschaftlich und organisatorisch selbstständiger bzw. autonomer Leistungserbringer ist eigentlich die klassische Aufgabe von Märkten.

Der Austausch und Transfer von Wissen zwischen den einzelnen Leistungserbringern wiederum, die in Strategischen Netzwerkorganisationen vorhandenen Regelsysteme (vgl. z.B. Toyotas „network rules“ in Dyer/Nobeoka 2000, S 357f) und die Zielgerichtetheit des Leistungserstellungsprozesses auf die integrierte Gesamtleistung hin, sind genauso Merkmale integrierter Unternehmungen.

Die primäre Erkenntnis aus diesen beiden Beobachtungen ist die Betrachtung der Strategischen Netzwerkorganisation als Meta-Organisation: Selbstständige und autonome Leistungserbringer werden im Rahmen eines verbindlichen Regelwerks koordiniert.

Darüber hinaus lassen sich schon an diesem Punkt strategische Netzwerkorganisationen von anderen interorganisationalen Netzwerken, die keine integrierte (Gesamt-) Leistung erbringen, wie insbesondere (branchenübergreifende) regionale Netzwerke und (branchenbezogene) Cluster, abgrenzen (vgl. Bauer 2002, S 17). Es ist aber durchaus denkbar, dass ein strategisches Netzwerk beispielsweise den Kern eines Clusters bildet.

Nicht nur auf Grund der gemeinsam erbrachten integrierten Leistung, sondern auch ausgehend von der Definition als Meta-Organisation unterscheiden sich strategische Netzwerkorganisationen letztlich auch von Sozialen Netzwerken (z.B. Boos/Exner 1992).

2.2 Charakteristika von Netzwerkteilnehmern (Knoten)

Die Bezeichnung der Strategischen Netzwerkorganisation impliziert aber auch die Beschränktheit des Teilnehmerkreises auf Organisationen. Dies stimmt in der grundlegenden Tendenz, wie aber in der Folge das Beispiel der Open-Source-Software-Systeme zeigt, sind es eher die Eigenschaften einer Organisation, die von den Teilnehmern erfüllt werden müssen.

Wesentlichstes Charakteristikum ist die autonome und (wirtschaftlich) selbstständige Erbringung einer Teilleistung. Und eben dieses Charakteristikum erfüllen nun beispielsweise auch einzelne Personen bzw. Entwickler in Open-Source-Software-Systemen (als quasi Ein-Mensch-Organisationen).

Einhergehend mit der arbeitsteiligen Erbringung der integrierten Gesamtleistung ist auch die Spezialisierung der verschiedenen Netzwerkteilnehmer auf einzelne Aufgabenbereiche. Diese Spezialisierung erfordert aber keineswegs Ausschließlichkeit in der Form, dass gleiche Aufgaben nicht auch von verschiedenen Netzwerkteilnehmern erfüllt werden können. Im Gegenteil, redundante Kompetenzen bzw. Teilnehmer mit gleichen Spezialisierungsgebieten sind ein Merkmal Strategischer Netzwerkorganisationen.

Wie hoch der Grad dieser Spezialisierung dann in der konkreten Ausprägung der strategischen Netzwerkorganisation ist, kann von Typus zu Typus variieren, wie auch im folgenden die Beispiele Strategisches Unternehmungsnetzwerk (sehr hoher Spezialisierungsgrad) und Open-Source-Software-Systeme (eher geringer Spezialisierungsgrad) zeigen werden.

2.3 Charakteristika der Netzwerkbeziehungen (Kanten)

Schon die Bezeichnung als *Netzwerkorganisation* legt Beziehungen zwischen den einzelnen Netzwerkteilnehmern nahe, die über die idealtypischen marktlichen „arm's length ties“ hinausgehen. In diesen Kontext fällt auch Uzzi (1996, S 677 ff) Einteilung

interorganisationaler Beziehungen in enge und lose („weak“ bzw. „strong ties“), wobei Uzziis „strong ties“ mit intensivem interorganisationalem Austausch die Netzwerkbeziehungen in einer Strategischen Netzwerkorganisation schon sehr gut beschreiben. Dies vor allem auch deshalb, weil der mit „strong ties“ verbundene höhere Koordinationsaufwand charakteristisch für die Koordiniertheit der Beziehungen in einer Strategischen Netzwerkorganisation ist.

Dies schließt andererseits aber nicht aus, dass in einem Netzwerk „weak“ und „strong ties“ durchaus nebeneinander vorkommen, da ja auch interorganisationaler Austausch nicht notwendig zwischen allen Teilnehmern des Netzwerks stattfindet. Umgekehrt ist aber eine strategische Netzwerkorganisation ohne enge Beziehungen zumindest zwischen Teilen der Netzwerkmitglieder nicht denkbar.

Was die konkrete Ausformung interorganisationaler Beziehungen innerhalb des Netzwerks betrifft, so unterscheidet Bauer (2002, S 23)) drei Ebenen: Einen latenten Markttest, der in einem einheitlichen Bewertungsmaßstab mit netzwerk-externer Vergleichbarkeit besteht (z.B. Geld in Unternehmungsnetzwerken), interorganisationale Standardisierung (z.B. durch Aus- und Weiterbildung in Unternehmungsnetzwerken, gemeinsamer Programmiersprache und einheitlichen Programmierstandards in Open-Source-Software-Systemen) und inter(organisations)kulturelle Kommunikation (z.B. gezielt organisierte zwischenmenschliche Kommunikation in Unternehmungsnetzwerken).

2.4 Charakteristika des Gesamtnetzwerks

Die Erfassung von strategischen Netzwerkorganisationen in ihrer Gesamtheit erfordert eine kurze Betrachtung der Basisaufgabe (menschlicher) Organisationen: Der Bewältigung von Komplexität.

2.4.1 Komplexitätsbewältigung durch Selbstorganisation

Folgt man Probst (1987, S 13), dann hat man es bei (sozialen bzw. aus Menschen bestehenden) Organisationen mit chaotischen Systemen hoher Komplexität zu tun, die oftmals klassischen, auf monokausalen Ursache-Wirkungs-Annahmen basierenden Lenkungssystemen unzugänglich sind. Umso mehr muss diese Charakterisierung nun auf strategische Netzwerkorganisationen zutreffen, in der zahlreiche, schon für sich hoch komplexe, Organisationen eine gemeinsame Leistung erbringen. Die Erbringung der gemeinsamen Leistung (in der spezifischen Qualität) erfordert nun gerade die Bewältigung eben dieses hohen Ausmaßes an Komplexität.

Komplexität "zu bewältigen" bedeutet aber nun, wie Probst treffend formuliert "ein System hoher Komplexität unter Kontrolle zu bringen" (Probst 1987, S. 40). Mit anderen Worten ausgedrückt, um das soziale System „Organisation“ zu lenken, bedarf es der Bewältigung der vorhandenen Komplexität, was im Umkehrschluss wiederum bedeutet, dass jede Form der Lenkung Komplexitätsbewältigung impliziert. In weiterer Folge sind Strukturen, die der Lenkung einer Organisation dienen, nichts anderes als bestimmte Formen der Komplexitätsbewältigung.

Ausgehend von dieser Überlegung stellt sich nun die Frage, welche Vorteile sich aus Netzwerkstrukturen im allgemeinen und aus zwischenbetrieblichen bzw. virtuellen Netzwerkstrukturen im speziellen für die Komplexitätsbewältigung ergeben. Dies ist vor allem hinsichtlich des Vergleichs mit der klassischen Form einer integrierten Organisation zur Leistungserbringung von Interesse: Folgt man Ashby's Überlegungen, dass Varietät (im Sinne von Komplexität) nur mit Varietät selbst bewältigt werden kann (im englischen Original heißt es "...only variety can destroy variety.") (Probst 1987, S 40), so stoßen klassische Formen der Organisation mit ihren beschränkten Kontroll- und Koordinationskapazitäten schnell an ihre Grenzen. Diese Grenzen lassen sich zwar durch Monopolgewinne oder sonstige Formen von Quersubventionierung verschieben, an irgendeinem Punkt aber stehen in der Folge Märkte, die mit der ihnen innewohnenden Komplexität die Koordination der Leistungen übernehmen.

Strategische Netzwerkorganisationen ermöglichen nun die Bewältigung eines größeren Ausmaßes an Komplexität, indem wesentliche Voraussetzungen für die Entstehung und das Funktionieren selbstorganisierender Prozesse erfüllt sind (nach Probst 1987):

- Komplexität: Eine strategische Netzwerkorganisation ist, wie bereits ausgeführt, eine hoch komplexe Organisationsform. Diese Komplexität erfordert nun selbstorganisierende Prozesse zu ihrer Bewältigung, ist aber auch gleichzeitig Voraussetzung für deren Entstehen und Funktionieren.
- Autonomie: Autonome Entscheidungsspielräume der Mitglieder einer Strategischen Netzwerkorganisation sind, da es sich ja definitionsgemäß um autonome und wirtschaftlich selbstständige Einheiten handelt, in hohem Maße vorhanden.
- Redundanz: Der Wissenstransfer und Austausch innerhalb der Strategischen Netzwerkorganisation sorgt automatisch für redundantes Wissen und bildet damit die Basis für selbstorganisierende Prozesse.

Wenn – wie bei Probst - die Selbstorganisationsfähigkeit einer einzelnen Organisation als Ausgangspunkt für die Beschränkung auf ein Meta-Management der Selbstorganisation nun als universeller Imperativ für einzelne Organisationen (wie z.B. mittelgroße Unternehmen) noch zweifelhaft ist, so scheint eben dieser Imperativ auf strategische Netzwerkorganisationen doch anwendbar zu sein: Das hohe Ausmaß an Komplexität in Verbindung mit dem hohen Ausmaß an Redundanz unter den einzelnen Netzwerkteilnehmern in einem gemeinsamen sozialen System, stellt geradezu den Paradefall einer Selbstorganisation begünstigenden und erfordernden Organisationsform dar.

Schließlich ist ein Merkmal von Netzwerkstrukturen, dass die einzelnen Beziehungen der Netzwerkmitglieder untereinander gerade nicht völlig durchstrukturiert, geplant und/oder zentral vorgegeben sind, sondern zu einem guten Teil selbstorganisierend entstehen. Niemand, auch eine allfällige fokale Instanz nicht, hat einen totalen Überblick über das Gesamtnetzwerk. Umgekehrt wäre der Koordinationsaufwand (zumindest wirtschaftlich) auch nicht bewältigbar. Zusammenfassend ergibt sich daraus die Unterstützung der Selbstorganisationsfähigkeit als wichtigste (Meta-)Koordinationsaufgabe einer zentralen Netzwerkinstanz.

2.4.2 Verschiedene Grade der Zentralität

Das Ausmaß, in dem selbstorganisierende Kräfte die Netzwerkstruktur bestimmen bzw. organisieren, hängt nun von Stärke und (Eingriffs-)Intensität eben jener zentralen Koordinationsinstanz ab. Abgesehen von der Frage, in wie weit Selbstorganisation – durch Meta-Management - „organisiert“ werden kann und muss, auf die hier nicht näher eingegangen werden soll, lassen sich verschiedene Grade der Zentralität von Strategischen Netzwerkorganisationen unterscheiden. So beschreibt auch Bauer (2002, S 27) Netzwerk-Organisationen von niedriger, mittlerer und hoher Zentralität:

- Niedrige Zentralität: Der hypothetisch niedrigste Zentralitätsgrad herrscht in einer Strategischen Netzwerkorganisation, die ihre gemeinsam erbrachte, integrierte Gesamtleistung ohne zentrale Koordinationsinstanz erbringt. Jegliche von außen „erkannte“ Ordnung wäre in diesem Netzwerk auf selbstorganisierende Kräfte zurückzuführen, die Idee der integrierten Leistung das koordinierende Element.

Auch wenn sich zahlreiche empirische Beispiele von strategischen Unternehmungsnetzwerken rund um starke Zentralinstanzen entwickelt haben (z.B. Toyota, Benetton), so erscheinen Netzwerke mit viel niedrigerer Zentralität durchaus vorstellbar. Dies insbesondere, als bei

immer komplexeren Aufgabenstellungen koordinierende Instanzen immer mehr gezwungen sind, sich auf Meta-Management der Selbstorganisation zu beschränken.

- Mittlere/Mäßige Zentralität: In Strategischen Netzwerkorganisationen mit mäßiger Zentralität übernehmen einzelne Netzwerkmitglieder als zentrale Dienstleister Aufgaben für zahlreiche oder alle anderen Mitglieder, um Skalenerträge und andere Effizienz und Effektivitätsvorteile zu nutzen.

- Hohe Zentralität: Ist die Komplexität der Leistung noch durch eine hierarchisch integrierte Organisation bewältigbar und ergibt sich trotzdem ein (Effizienz-)Vorteil aus einer Netzwerkstruktur, so ist auch direkte zentrale Koordination des Netzwerks denkbar. Klassisches empirisches Beispiel für diesen Fall ist das strategische Unternehmungsnetzwerk von Benetton (vgl. Sydow 1992, Jarillo 1993).

Je nach Grad der Zentralität ermöglicht eine Strategische Netzwerkorganisation in verschiedenem Ausmaß Kooperation, wie sie sonst nur innerhalb einer Organisation möglich ist, bei gleichzeitig hoher Komplexitätsbewältigungskapazität durch Integration marktlicher Koordinationsmechanismen in die Netzwerkorganisation.

3 Strategische Unternehmungsnetzwerke

3.1 Begriffsexplikation

Strategische Unternehmungsnetzwerke, wie sie ausführlich von Sydow (1992) und Jarillo (1993) beschrieben wurden, stellen nun den klassischen Fall einer strategischen Netzwerkorganisation dar: Rechtlich selbstständige Betriebe erbringen gemeinsam in enger (über rein marktliche Austauschbeziehungen weit hinausgehender) Beziehung zueinander eine komplexe, integrierte (Gesamt-) Leistung (vgl. auch Bauer 2002).

3.1.1 Knoten

Die integrierte Gesamtleistung eines Strategischen Unternehmungsnetzwerks wird von rechtlich und zumindest formal wirtschaftlich unabhängigen Unternehmungen mit relativ hohem Spezialisierungsgrad erbracht (Sydow 1992, S 79 / Bauer 2002, S 17). Gerade diese Spezialisierung einzelner Netzwerkteilnehmer auf ihre jeweiligen Aufgaben stellt nun eine Basis für hohe Qualität und Effizienz bei Erstellung der Teilleistung dar: Einerseits, da die häufige Wiederholung gleicher bzw. ähnlicher Arbeitsschritte organisationales Lernen ermöglicht. Andererseits, da durch die Konzentration auf wenige spezifische Aufgaben weniger Kompromisse beim organisationalen Design der Unternehmung eingegangen werden müssen, als dies bei hierarchisch integrierten Organisationen mit zahlreichen verschiedenen Aufgaben der Fall ist (vgl. Bauer 2002, S 20f)

Schließlich werden durch Erfüllung der Teilleistungen in autonomen Betrieben innerbetriebliche und zwischenbetriebliche Arbeitsteilung als Quellen für Produktivitätsfortschritte gleichermaßen genutzt.

3.1.2 Kanten

Wie bei der innerbetrieblichen ergibt sich auch bei der zwischenbetrieblichen Arbeitsteilung die Frage der Zusammenführung der Einzelleistung zur finalen integrierten Teil- bzw. Gesamtleistung. Während die dafür notwendige Koordinationsleistung dem Gesamtnetzwerk zugeschrieben und hier nicht erläutert wird, so sind auch die Beziehungen der einzelnen Netzwerkunternehmen untereinander von besonderer Qualität.

Diese besondere Qualität lässt sich nicht nur an den Merkmalen festmachen, die Uzzi (1996, S 677f) für sogenannte „strong ties“ expliziert hat: „Trust, fine-grained information transfer, and joint problem-solving arrangements.“ Denn über diese Merkmale hinaus findet zwischen

einzelnen Netzwerkunternehmungen idealtypisch noch stärkerer Wissenstransfer statt. So umfasst Uzzi's „fine-grained information transfer“ zwar über das normale Maß hinausgehende Informationen über die eigenen Leistungen und den eigenen Betrieb, nicht jedoch betriebliche Geheimnisse wie Fertigungstechniken oder ähnliches. Gerade aber der Austausch bzw. die Freigabe von derart sensiblem Know-how findet in strategischen Netzwerkorganisationen statt (vgl. Dyer/Nobeoka 2000).

Die allgemeinen Feststellungen in Kapitel 2.3, insbesondere das Nebeneinander von „weak“ und „strong ties“ innerhalb desselben Netzwerks und die verschiedenen Ausformungen interorganisationaler Beziehungen (Bauer 2003) haben logischerweise auch für Unternehmungsnetzwerke Gültigkeit und sollen hier nicht wiederholt werden.

3.1.3 Gesamtnetzwerk

Wie bereits ausgeführt, ist die Bewältigung hoher Komplexität gleichzeitig Merkmal und Aufgabe strategischer Netzwerkorganisationen. Die konkrete Ausprägung insbesondere des Zentralisierungsgrades bei strategischen Unternehmungsnetzwerken hängt nun wenig überraschend von der Komplexität der zu erbringenden Leistung ab. Je komplexer die Leistung und/oder der Leistungserstellungsprozess desto niedriger ist der idealtypische Zentralisierungsgrad und die Beschränkung auf das Meta-Management der Selbstorganisation.

Real existierende Unternehmungsnetzwerke und deren Koordinationsstruktur richten sich aber nicht (nur) nach Effizienz- und Effektivitätsgesichtspunkten sondern auch nach Rentabilitätsgesichtspunkten. So sind die meisten Beispiele großer strategischer Netzwerke evolutiv aus traditionellen, vertikal integrierten Unternehmungen hervorgegangen. Diese schon allein historisch hierarchisierte Struktur lässt eine (zumindest leichte) Dominanz von Profitinteressen der fokalen Unternehmung manifestiert in klassischen Kontrollmechanismen auf Kosten einer idealtypischen Koordinationsstruktur erwarten.

Der Grad der Zentralität hat aber auch Einfluss auf das Ausmaß an Vertrauen in die Zentralkoordination, das für das Funktionieren des Unternehmungsnetzwerks notwendig ist. So ist beispielsweise beim in der Autoindustrie zu beobachtenden Rückgang der Netzwerkzentralität, bei der sich bisherige „Hersteller“ verstärkt auf Marketing und Absatz konzentrieren und ihre „Zulieferer“ an Bedeutung gewinnen (Bauer 2002, S 29), das Vertrauen auf die Kompetenz des als Koordinationsinstanz fungierenden „Herstellers“ immer entscheidenderes Legitimationsmerkmal. Die Stärke der Legitimation kann sich so beispielsweise mit der Stärke der Marke und des Vertriebskanals quantifizieren lassen. Als

praktisches Beispiel wäre hier der „Zulieferbetrieb“ Magna zu nennen, der bereits ein komplettes Auto („X3“) für einen anderen „Hersteller“ (BMW) - nach gemeinsamer Entwicklung - fertigt (Automotive Intelligence News, 2001).

3.2 Abgrenzung zu marktlich und hierarchisch organisierter Leistungserstellung

Bevor strategische Unternehmungsnetzwerke analysiert und bezüglich ihrer Koordinationseffizienz und –effektivität bewertet werden können, ist es notwendig, die organisationalen Alternativkonzepte vertikale Integration und Funktionsexternalisierung (Outsourcing) sowie die allgemeine Aufgabenstellung sämtlicher drei Organisationskonzepte, nämlich die Bewältigung von Komplexität in modularen Leistungserstellungsprozessen, darzustellen.

3.2.1 Komplexität und Modularität von Produkt und Prozess

Nicht zufällig stehen Komplexität und Modularität am Beginn des Vergleichs von strategischen Unternehmungsnetzwerken mit marktlichen oder hierarchischen Formen der Organisation des Leistungserstellungsprozesses: Komplexität und deren Bewältigung steht hinter jeglicher Leistungserstellung und Modularität von Leistung und Leistungserstellung ist ein wesentlicher Komplexitätstreiber. Dies folgt schon allein aus der banalen Tatsache, dass mit steigender Modularität und Arbeitsteilung (Lawrence/Lorsch 1967) und mehr noch mit steigender Modulinterdependenz (Augier/Simon 2003) die Komplexität im Leistungserstellungsprozess und damit auch der erforderliche Koordinationsaufwand zunimmt.

Jarillo (1993) erklärt die Komplexitätserhöhung durch steigenden Aufwand für die Koordination der einzelnen „Aktivitäten“, die sowohl einzelne Arbeitsschritte im Sinne klassischer Arbeitsteilung als auch Module im Sinne ganzer Abschnitte des Wertschöpfungsprozesses umfassen können. Bei näherer Betrachtung ist die Einteilung, was als Aktivität angesehen wird, eine willkürliche, die der Entscheidung der Tiefe der (inner- und zwischenbetrieblichen) Arbeitsteilung entspricht.

Das Ausmaß dieser Komplexitätssteigerung mit dem Modularisierungsgrad wiederum hängt wesentlich von Anzahl und Art der Interdependenzen der einzelnen Module bzw. Aktivitäten untereinander ab. So führt die rein sequentielle Unterteilung des Fertigungsprozesses (z.B. Fließband) klarerweise zu einem geringeren Komplexitätszuwachs als bei rückgekoppelten und parallelen Arbeitsschritten (z.B. „concurrent engineering“ im modernen Automobilbau) oder

gar ständig notwendigen Konsistenzprüfungen bei wechselseitig aufeinander bezogenen Schnittstellen (z.B. Softwareproduktion). (Zu den verschiedenen Ausprägungen organisationaler Komplexität: Moldoveanu/Bauer 2003)

Der mit steigender Modularität (wie beispielsweise in der Automobilbranche) verbundene Koordinationsaufwand ist nun oftmals, wie die folgenden Kapitel zeigen werden, weder in integrierten Unternehmen noch durch klassisches Outsourcing effizient zu bewältigen.

3.2.2 Grenzen klassischer organisationaler Eingliederung (Integration)

Jarillo (1993) hat in seinem Standardwerk „Strategic Networks“ ausführlich die Defizite sowohl hoch integrierter Unternehmen als eben auch des „Over-Subcontractings“ als Form marktlicher Organisation für den Unternehmensbereich herausgearbeitet. Wichtig bei Jarillos Analyse der Defizite der beiden herkömmlichen Organisationsformen ist aber, dass er die Defizite im Vergleich eben dieser beiden herkömmlichen Organisationsformen zueinander herausarbeitet. Mit der Konsequenz, dass Effizienz Nachteile bei der Erfüllung einer Aufgabe In-House im Vergleich zur Fremdvergabe derselben nicht auch bzw. nicht im gleichen Ausmaß beim Vergleich In-House vs. Strategisches Unternehmungsnetzwerk bestehen.

Das Problem hierarchischer Koordination von Leistungserbringungsprozessen ist eines der ältesten in der betriebswirtschaftlichen Forschung, da es sich der Grundfrage nach der Zusammenführung der arbeitsteiligen Leistungen innerhalb einer Organisation widmet (vgl. Schreyögg 1999, S 154). So ist es nicht weiter überraschend, dass Trends in der Organisationsgestaltung wie beispielsweise Divisionalisierung oder Outsourcing als Reaktion auf Ineffizienzen einfacher hierarchischer Integration von Teilleistungen entstanden sind.

Diesen zahlreichen (in der Regel auch bei divisionaler Struktur bestehenden) potentiellen Effizienz Nachteilen vertikal integrierter Unternehmungen, widmet sich die folgende kurze Übersicht:

- Economies of Scale eines Zulieferbetriebes: Ein Zulieferbetrieb kann mehr als nur einen Kunden beliefern. Das ermöglicht ihm in der Regel höhere Stückzahlen und damit verbundene Einkaufs- und Produktionsersparnisse sowie Lernkurveneffekte.
- Unerfahrenheit bei der konkreten Aktivität, die durch Nachbesserungen, erhöhten Entwicklungsaufwand etc. Kosten verursacht. So weist auch Jarillo (1993, S 59) auf diesen Punkt hin: „... each activity of the business system is in fact a different business, in the sense that the technology, optimum size, distribution methods etc. may be different.“

- Entstehung interner Monopole mit den ähnlichen Zielverschiebungen und den damit verbundenen Effizienzachteilen wie sie auch in externen Monopolen erfolgen.
- „Mismatch of Cultures“: Abgesehen von den Vorteilen unbürokratischer Unternehmenskulturen in kleineren Firmen sind in größeren Unternehmen verschiedene, für die konkrete Aufgabe passende, Kulturen und Lohnniveaus (z.B. nach Firmenübernahmen) meist schwer aufrecht zu erhalten.
- Flexibilitätseinbußen in Verbindung mit einer Erhöhung der Fixkostenbasis. So geht mit dem Outsourcing von Leistungen gleichzeitig auch eine Variabilisierung von Kosten einher.

3.2.3 Grenzen klassischer zwischenbetrieblicher Arbeitsteilung (Outsourcing)

In Kenntnis der Probleme vertikaler Integration verfolgen nun zahlreiche Unternehmen Outsourcing- bzw. Funktionsexternalisierungs-Strategien. Neben der Begegnung obiger Effizienzprobleme vertikaler Integration entspricht Outsourcing auch einem klassischen Rationalisierungsverständnis, das auf Reduzierung der Personalkosten und des Gewerkschaftseinflusses abzielt (Sydow 1992, 108).

Zahlreiche empirische Beispiele (vgl. IBM, General Electrics, General Motors in Jarillo 1993, S 83ff) zeigen aber, dass auch Outsourcing mit Risiken und potentiellen Effizienzachteilen verbunden ist. So im speziellen folgende:

- Die Gefahr der Auslagerung von Kernkompetenzen, insbesondere beim Outsourcing kompletter Subsysteme/Module bzw. der Konzentration auf unprofitable Module/Teilleistungen (z.B. der IBM-PC, bei dem Intel und Microsoft die profitablen Bereiche Prozessor und Software übernahmen).
- Hohe Transaktionskosten bei der Umgehung dieser Gefahren durch kurze Verträge, maximale Anzahl an Zulieferern und ausschließlicher Fremdvergabe von Sub-Modulen. Denn schließlich erfordert dieses Vorgehen in der Regel große Lagerbestände und ist durch die große Anzahl an Schnittstellen (u.a. wegen der ausschließlichen Vergabe von Sub-Modulen) fehleranfällig.
- Die Fehleranfälligkeit sowohl der Komponenten als auch des Gesamtsystems und mit ihr die Transaktionskosten (durch Aus- und Nachbesserungen, gesteigerten Koordinationsbedarf über Organisationsgrenzen hinweg etc.) steigen nun weiter mit der Interdependenz der einzelnen Teilleistungen (vgl. Augier/Simon 2003, S 41).

- Verhinderung oder Erschwerung von Lerneffekten einerseits in den Zulieferbetrieben durch regelmäßige Neuvergabe der Aufträge und andererseits zwischen den einzelnen Betrieben, da der starke Wettbewerb Wissensaustausch untereinander unmöglich macht und keinerlei Anreize für Wissensaustausch bestehen. Im Gegenteil, geben einzelne Zulieferbetriebe ihr spezifisches und für andere nützliches Wissen preis, machen sie sich dadurch selbst überflüssig.

3.2.4 Strategische Unternehmungsnetzwerke im Bereich klassischer Koordinationsdilemmata

Strategische Unternehmungsnetzwerke stellen nun eine alternative Möglichkeit dar, Ineffizienzen im Bereich der Zusammenführung bzw. Koordination inner- und zwischenbetrieblicher Arbeitsteilung, zu begegnen.

Am Beginn steht die Erkenntnis über den Konnex zwischen Komplexität und Koordinationsaufwand, da die Lösung der Koordinationsfrage unmittelbar durch die Wahl der Organisationsform erfolgt. So jedenfalls auch Jarillo (1993, S 14), der den Wertschöpfungs- und Kostencharakter der Lösung der Koordinationsaufgabe hervorstreicht: „Thus there are two kinds of costs or sources of value: doing the things, (...) and coordinating those activities. This second set of costs (...) is directly affected by the choice of the organizational form.“ bzw. (ebd. S 42) „The coordination of the different activities that end up producing the final product or service is also a costly, value-adding activity, and it can be performed in a more or less efficient (and defensible) way, ...“.

Schon rein abstrakte Überlegungen, die Struktur von strategischen Netzwerkorganisationen und deren Komplexitätsgrad betreffend, weisen nun – in Einklang mit den angeführten Eigenschaften die Selbstorganisationsfähigkeit betreffend – auf eine gesteigerte Komplexitätsbewältigungs- und damit gleichzeitig Koordinationsfähigkeit, verglichen mit marktlichen oder hierarchischen Organisationsformen, hin.

Der Umkehrschluss, nämlich, dass Strategische Unternehmungsnetzwerke sich im Gegensatz nicht für wenig komplexe Leistungen mit geringem Koordinationsaufwand eignen, ist nicht ohne weiteres zulässig. Diese Überlegung verbietet sich allein auf Grund der Tatsache, dass eine gesteigerte Koordinationsfähigkeit einen wesentlichen, aber nicht den einzigen Vorteil der Netzwerkorganisation gegenüber klassischen Organisationskonzepten darstellt.

Die weiteren potentielle Effizienzvorteile von strategischen Unternehmungsnetzwerken lassen sich am besten von zwei Standpunkten aus erläutern, wie Sydow (1992) mit seinen beiden Begriffen der „Quasi-Internalisierung“ für die Kooperation mit selbständigen Firmen ohne vollständige organisatorische Eingliederung und der „Quasi-Externalisierung“ für die Auslagerung von Aktivitäten in selbstständige Unternehmen in Beibehaltung enger kooperativer Beziehungen zeigt.

Folgt man der einschlägigen Literatur (Sydow 1992, Jarillo 1993 et al.) treten nun strategische Unternehmungsnetzwerke den beschriebenen potentiellen Effizienzvorteilen sowohl der vertikalen Integration als auch der Funktionsexternalisierung entgegen. Das Ausmaß, in dem ein potentieller Effizienzvorteil trotzdem bzw. immer noch zum Tragen kommt, hängt im konkreten Fall wesentlich von der Art des Netzwerks in Verbindung mit der Art der produzierten Leistung ab. So gibt es – und es kann je nach erbrachter Leistung auch durchaus sinnvoll sein – strategische Netzwerke, die tendenziell größere Ähnlichkeiten mit integrierten Unternehmen aufweisen und den einzelnen Mitgliedern nur geringe Autonomie zugestehen (z.B. McDonalds). Diesen gegenüber steht die teilweise Entwicklung (beispielsweise in der Automobilbranche) hin zu größerer Autonomie der Mitglieder – bei gleichzeitig engen Netzwerkbeziehungen - , was auf die Komplexität der zu erbringenden Leistung zurückzuführen sein dürfte.

Zusammengefasst und verallgemeinert haben Strategische Unternehmungsnetzwerke vom Standpunkt einer „Quasi-Internalisierung“ aus folgende potentiellen Vorteile gegenüber klassisch vertikal integrierten Unternehmen:

- Economies of Scale der Zulieferbetriebe sind grundsätzlich möglich. Auch wenn in zahlreichen empirischen Beispielen (insbesondere der Sonderfall Benetton) die überwiegende Mehrheit der Betriebe im Netzwerk nur innerhalb des Netzwerks liefert, so ist dies keineswegs notwendigerweise so.
- Spezialisierung der Betriebe und Fokussierung auf die jeweiligen, relativ engen Aufgaben ermöglichen auf Wiederholung gleicher bzw. ähnlicher Arbeitsschritte basierendes Lernen (Bauer 2002, S 20).
- Die simple wirtschaftliche Selbstständigkeit und auch nur stark beschränkter Autonomie spielt eine nicht zu unterschätzende Rolle, wie das Beispiel des McDonalds-Franchisesystems beweist: In McDonalds-Filialien ist vom Sortiment über die Ausstattung bis hin zu den einzelnen Arbeitsvorgängen der Mitarbeiter alles standardisiert. Und

trotzdem sind die Filialen, die von wirtschaftlich selbstständigen Franchisenehmern betrieben werden, profitabler als die direkt von McDonalds gehaltenen Filialen, was Jarillo zu folgendem Schluss kommen lässt (1993, S 17): „If the technology and the basic operations are the same, then the motivation of individual ownership must make the difference.“

- Neben dem permanent vorhandenen latenten Markttest und der wirtschaftlichen Selbstständigkeit der Betriebe sorgt auch das Nebeneinander verschiedener Betriebe mit gleichen oder ähnlichen Aufgabenbereichen für effizienzfördernden Wettbewerbsdruck. Dieser ist allerdings nicht so ausgeprägt wie bei klassischem Outsourcing, weil sich aus den engen Beziehungen innerhalb des Netzwerks in der Regel weniger Unsicherheit über zukünftige Partizipation am Leistungserstellungsprozess einstellt.
- Die kleineren selbstständigen Einheiten ermöglichen eine der Wertschöpfung entsprechende Entlohnung und der Aufgabenstellung angemessene Organisationskultur. Zu diesen Vorteilen kommt noch die allgemein bessere Überschaubarkeit kleinerer und unbürokratischerer Einheiten.
- Der Gewinn an Flexibilität im klassischen Sinn der Weitergabe von Umsatzrückgängen an die Zulieferer durch eine fokale Unternehmung ist zwar vorhanden, sollte aber nicht überbewertet werden. Nicht nur in Anbetracht des für enge Beziehungen und Wissensaustausch notwendigen Vertrauensverhältnisses ist eine faire Verteilung (unter Berücksichtigung der individuellen Tragfähigkeit) negativer Entwicklungen auf die Teilnehmer des Netzwerks sinnvoll. Denn eine zu starke Schwächung der einzelnen Betriebe schadet mittelfristig dem gesamten Netzwerk. Sehr wohl stellt aber die Struktur einen Flexibilitätsgewinn gegenüber einer vertikal integrierten Unternehmung dar, wenn es um technologische Weiterentwicklungen geht: Hier können Strategische Netzwerkorganisationen schneller und einfacher reagieren. (Jarillo 1993, S 66)

Vom Standpunkt der „Quasi-Externalisierung“ aus ergibt sich gegenüber klassischer Funktionsexternalisierung folgendes Bild:

- Die beschriebene Gefahr der Auslagerung von Kernkompetenzen beim Outsourcing kompletter Subsysteme/Module bzw. der Konzentration auf unprofitable Module/Teilleistungen ist auch im strategischen Unternehmungsnetzwerk nicht völlig ausgeschlossen. Dies insbesondere aus Sicht einer allfälligen fokalen Unternehmung und unter deren Rentabilitätsgesichtspunkten. Zumindest aber bleiben die Kernkompetenzen im

Gesamtnetzwerk vorhanden und der Verlust an Know-how sollte durch ständigen Wissenstransfer auch gering gehalten werden.

- Die engen Beziehungen zwischen den einzelnen Betrieben ermöglichen eine geringere Anzahl an Betrieben die gleiche oder ähnliche Aktivitäten wahrnehmen., wobei trotzdem aus Innovations- und Ausfallsicherheitsüberlegungen mehrere redundante Betriebe sinnvoll sind. Zusätzlich ergibt sich aus den besonderen Netzwerkbeziehungen ein besserer Informationsfluss, der auch bei größerer Schnittstellenanzahl Fehlerhaftigkeit reduziert.
- Mit der geringeren Anzahl und den „strong ties“ innerhalb des Netzwerks einher geht in Strategischen Unternehmungsnetzwerken eine erhöhte – teilweise durch Regelwerke abgesicherte (vgl. Dyer/Nobeoka 2000) – Existenzsicherheit. Die wiederum erlaubt noch stärkeren Wissensaustausch zwischen den einzelnen Betrieben, der auch - abgesehen vom gesondert zu behandelnden Innovationsbereich – der Fehlerhäufigkeit im Leistungskordinationsprozess entgegenwirkt.
- Neben der Verminderung von Barrieren für interorganisationalen Wissenstransfer ermöglicht die erhöhte Existenzsicherheit schließlich den Betrieben langfristige Lerneffekte bzw. Innovationsprozesse und stellt einen Investitionsanreiz dar.

3.3 Empirische Beispiele für strategische Unternehmungsnetzwerke

Die Anzahl und Diversität der in der Literatur unter dem Phänomen der Strategischen Netzwerkorganisation angeführten empirischen Beispiele ist vielfältig und zur genaueren Betrachtung wird auf die Literatur im Anhang verwiesen.

Neben den bereits erwähnten Beispielen des „Toyota Knowledge-Sharing Networks“ (u.a. Dyer/Nobeoka 2000), McDonalds (Sydow 1992, S 30f) und Benetton (ebenda, S 33f) soll hier nur noch gesondert das Dienstleistungsnetzwerk der Flughafen Frankfurt/Main AG (FAG) (Duschek 2001) erwähnt werden. Dies deshalb, weil es die Vorteile von Strategischen Unternehmungsnetzwerken auch abseits vom Fertigungsnetzwerken demonstriert und gleichzeitig der Komplexität im Dienstleistungsbereich völlig anders begegnet, als dies etwa das McDonalds-Franchisnetzwerk mit seiner Standardisierungsstrategie tut. So steht im Netzwerk der FAG der Wissensaustausch zur temporären Nutzung von (Kern-)Kompetenzen der Netzwerkmitglieder im Sinne einer „lerninduzierten Kooperation“ im Vordergrund (Duschek 2001, S 180).

4 Open-Source-Software-Systeme

4.1 Definition und Beschreibung

4.1.1 Allgemeines und historische Entwicklung

Open-Source-Software-Systeme sind im Rahmen eines (virtuellen) Netzwerks organisierte Softwareprojekte mit dem Grundprinzip der freien und kostenlosen Weitergabe des Quellcodes der Programme (engl. „Sourcecode“) und der Erlaubnis, diesen zu ändern (Tippmann 2001). Der Quelltext ist die Befehlsabfolge eines Computerprogrammes, dessen Kenntnis die Erzeugung („Kompilierung“) beliebig neuer bzw. verbesserter Versionen des gleichen Programmes ermöglicht (Morner 2002, S 219f).

Im Unterschied zu Open-Source-Software bleiben bei proprietärer Software sowohl das Ergebnis der Programmierarbeit in Form des Quellcodes als auch die Arbeitsprozesse, -fortschritte und Zwischenergebnisse nicht offen zugänglich. Vervielfältigung, Weiterverbreitung und Modifizierung sind untersagt und Eigentümer der Software bleibt der Hersteller, der das Urheberrecht und die vollständige Kontrolle über das Produkt hat (Leiteritz 2002, S 4).

Im Gegensatz dazu nennt www.opensource.org (letzter Besuch: 02.08.2003) die wesentlichen (formalen) Eigenschaften von Open-Source-Software:

- Jeder hat das Recht, die Software nach eigenem Ermessen zu nutzen.
- Der Quelltext muss jedem Benutzer offengelegt werden oder es muß auf eine frei zugängliche Stelle verwiesen werden, wo er erhältlich ist.
- Der Benutzer hat das Recht, die Software zu modifizieren und in modifizierter Form weiterzuverteilen.
- Die Lizenz darf niemanden im Verkauf oder der Weitergabe der Software in Form einer Softwarezusammenstellung einschränken.

Open-Source-Software-Entwicklung gleicht unter diesen Voraussetzungen in vielen Punkten einem wissenschaftlichen Prozess, wie DiBona/Ockman/Stone (1999, S 2) mit der Formulierung „Science, after all, is ultimately an Open Source enterprise“ andeuten. Mit den Prozessstufen Entwicklung/Entdeckung, Dokumentation und Validierung/Rechtfertigung in Form eines öffentlichen Dialogs wird die prinzipielle Nachvollziehbarkeit und Überprüfbarkeit des Quellcodes ermöglicht. DiBona et al. (1999, S 2) stellen genau an dieser Stelle fest: "But ultimately the process of discovery must be served by sharing information: enabling other

scientists to go forward where one cannot; pollinating the ideas of others so that something new may grow that otherwise would not have been born."

Die Ursprünge der Open-Source-Bewegung liegen bereits in den 60er Jahren des 20. Jahrhunderts, als Software regelmäßig freien Zusatz zur (teuren) Hardware darstellte. Die Entwicklung hin zur ausschließlichen kostenpflichtigen Lizenzierung von Software in Verbindung mit Unzulänglichkeiten insbesondere im Bereich der Hardware-Unterstützung führte in der Folge zu ersten Open-Source-Software-Projekten. Eine große Rolle spielte in diesem Zusammenhang Richard Stallman mit der Gründung der Free Software Foundation (FSF) und der Entwicklung der General Public License (GPL)¹, der damit einen institutionellen und rechtlichen Rahmen für Entwicklung und Vertrieb von Open-Source-Software schuf. (Morner 2002, 219f / Stallman 1998)

4.1.2 Knoten

Teilnehmer und somit Knoten eines Open-Source-Software-Systems sind grundsätzlich all jene Menschen und Firmen, die Beiträge zur Software und deren Entwicklung liefern. Diese Beiträge können entweder durch Schreiben und Veröffentlichen von Quellcode für die Software geschehen, oder durch Unterstützung bei der Fehlersuche, dem sogenannten „Debugging“. Ausschließliche Konsumenten bzw. Benutzer von Open-Source-Software sind in diesem Sinne noch nicht als Teilnehmer zu sehen, auch wenn der Übergang (beispielsweise wenn Bug-Reports erstattet werden) sicherlich ein fließender ist.

Ein wesentliches Merkmal aller Teilnehmer eines Open-Source-Software-Systems ist ihre relative Autonomie im Bezug auf das Gesamtnetzwerk. Jede Partizipation soll und kann nur freiwillig erfolgen, es besteht aber keine wirtschaftliche Abhängigkeit vom Open-Source-Software-System. Dass teilweise Programmierer von partizipierenden bzw. kontribuierenden Firmen wirtschaftlich abhängig sind, schadet nicht, da ja in diesem Fall die Firma als Organisation Bestandteil des Netzwerks ist und als solche wiederum freiwillig partizipiert. Das

¹ Lizenzen bilden auch bei Open-Source-Software den rechtlichen Rahmen der Nutzung. Wie bei proprietärer Software regeln sie die Nutzungsrechte am Programm und am Quellcode. Bei Open-Source-Software-Lizenzen steht aber im Gegensatz zu proprietären Lizenzen der Schutz der weiteren freien Verfügbarkeit des Sourcecodes im Vordergrund. Die verschiedenen Lizenzierungstypen erlauben aber bisweilen stark verschiedene Freiheiten im Umgang mit dem Quellcode. GPL (General Public Licence) zwingt beispielsweise bei Verwendung von Quellcode dieser Lizenzierungsform auch zur Freigabe des damit neu erstellten, gesamten Programms und verunmöglicht somit die Integration von GPL-Quellcode in proprietäre Software-Projekte. BSD-Lizenzen (Berkley Software Distribution) im Gegensatz dazu überlassen es weitestgehend dem Urheber des Quellcodes, welche Einschränkungen er für die Benutzung des Quellcodes auferlegen möchte und erlaubt beispielsweise oftmals den Einsatz von BSD-Quellcode in proprietärer Software (Prominentestes Beispiel

in Kapitel 2 geforderte Merkmal der Autonomie der Knoten von Strategischen Netzwerkorganisationen ist für den Fall des Open-Source-Software-Netzwerks jedenfalls in hohem Maße erfüllt.

Dass Open-Source-Software-Systeme hier trotz ihrer mehrheitlich menschlichen Mitglieder nicht als soziale Netzwerke sondern eben als Strategische Netzwerkorganisationen betrachtet werden, liegt daran, dass die Partizipation der einzelnen Mitglieder zum größten Teil in Form der Erbringung einer Teilleistung besteht, die im Rahmen des Open-Source-Software-Systems in einem integrierten (Gesamt-) Softwareprodukt resultiert.

Das Ausmaß an Spezialisierung ist im Softwarebereich bzw. unter den Softwareentwicklern noch bei weitem nicht so stark ausgeprägt wie in herkömmlichen Industrien. Allerdings hat genau diese Tatsache eine hohe Redundanz der Fähigkeiten zur Folge: Große Teile der Mitglieder eines Open-Source-Software-Netzwerks decken einen großen Teil der zur Erstellung insgesamt benötigten Fähigkeiten und Kompetenzen ab. Dies ist aber keineswegs typisch für den Open-Source-Bereich, da auch im Bereich proprietärer Software kleine Entwicklerteams vielfältige Aufgaben wahrnehmen (vgl. Schneidewind/Landsberger/Eggers 2002, S 227). Eine vergleichbare Kompetenzredundanz ist im stark auf Spezialisierungsvorteile setzenden strategischen Unternehmungsnetzwerk unmöglich.

Die Anzahl der an einem Open-Source-Software-Projekt beteiligten Personen schwankt stark in Abhängigkeit von der Größe des Programms und je nach dem, wer noch als Entwickler oder wer nur noch als Anwender gezählt wird. Die Partizipation ist aber keineswegs wahllos und gleichberechtigt, auch wenn prinzipiell jeder über das Internet an der Entwicklung eines Open-Source-Software-Projektes teilnehmen kann. Es findet sich vielmehr das System der sogenannten „Managed Memberships“ (Markus/Manville/Agres 2000, S 21) bei der Auswahl verantwortlicher (Unter-)Projektleiter, bei der nicht nur nach fachlichen sondern auch nach persönlichen Kriterien wie dem potentiellen Entscheidungsverhalten selektiert wird (Morner 2002, S 221).

So spielt in Open-Source-Software-Systemen die Anerkennung einer Verbesserung („patch“) des Programms durch den Projektleiter („Gatekeeper“) eine entscheidende Rolle, er macht ihn mit der Anerkennung zum de facto Standard. Dieser Effekt ist begründet im Vertrauensvorschuss, der zentralen Stellen entgegengebracht wird. Raymond (2001, S 74)

hierfür ist Apples Betriebssystem MacOS X, das auf einem BSD Unix-Kernel basiert.) (Wayner 2000, S 159ff, 343f)

bringt dieses Phänomen auf den Punkt: „...there is a very well-recognized distinction between ‚official‘ patches, approved and integrated into the evolving software by the publicly recognized maintainers, and ‚rouge‘ patches by third parties. Rouge patches are unusual, and generally not trusted.“ Die Vertrauenswürdigkeit einer Person hat im Open-Source-Bereich somit sogar konstitutiven Charakter: Das größere Vertrauen in „offizielle“, von vertrauenswürdiger Stelle anerkannter Verbesserungen schafft in gewisser Hinsicht erst das Produkt als legitimes (immer vorläufiges) Endergebnis der gemeinsamen Entwicklung.

4.1.3 Kanten

Die Beziehungen der Mitglieder eines Open-Source-Software-Netzwerks untereinander bilden eine Mittelstellung in einer immer noch großen Bandbreite zwischen den engen und wirtschaftlich begründeten Dienstgeber-Dienstnehmer-Verhältnissen proprietärer Softwareprojekte einerseits und den reinen Produzenten-Konsumenten-Beziehungen zwischen Herstellern und Käufern proprietärer Systeme andererseits. So steht, wie bereits erwähnt, die Teilnahme an Open-Source-Software-Projekten jedem grundsätzlich frei, solange die offenen Standards für Protokolle, Dateiformate und Schnittstellen eingehalten werden (BWT Deutschland 2001, S 10).

Folgt man wiederum der Einteilung Bauers (2002, S 23) in drei Ebenen, so finden sich sämtliche auch in Open-Source-Software-Systemen:

Im Vordergrund steht interorganisationale bzw. hier besser netzwerkweite Standardisierung: Sowohl die Lizenz des Open-Source-Projektes als auch die gewählte Programmiersprache und sonstige gemeinschaftlich vereinbarte Standards legen die Leitlinien für Form und Grenzen des Leistungsaustauschs innerhalb des Netzwerks fest. Morner (2002, S 222) spricht in diesem Zusammenhang von „Rahmenregelungen“ die in mehr oder weniger demokratischen Verfahren durch die Entwickler festgelegt werden.

Der von Bauer (2002) geforderte latente Markttest findet in Open-Source-Software-Systemen nicht durch monetäre Bewertung sondern durch praktische Anwendung und Vergleich von Programmierlösungen statt. Wettbewerb im Rahmen der Open-Source-Software-Entwicklung erfüllt somit eine direktere Qualitätssicherungsfunktion: Jeder Beitrag zu einem Projekt im Sinne einer Programmierleistung muss sich im Vergleich zu verfügbaren Alternativlösungen sowohl innerhalb als auch außerhalb des Open-Source-Software-Projektes durchsetzen. Fehlerhafte Lösungen eines Softwareproblems werden in der Folge entweder korrigiert oder durch fehlerfreiere ersetzt.

Die Kommunikation zwischen den Netzwerkteilnehmern als dritte Ebene ist in der Regel ausschließlich virtuell, d.h. per Mailing-Liste, Internet-Forum oder Softwareserver, und besteht im Austausch von Information. Die technische Virtualität der Beziehungen der einzelnen Teilnehmer zueinander wird andererseits durch die große Bedeutung von Persönlichkeit und Status innerhalb des Open-Source-Software-Systems kontrastiert, wie sich auch aus der bereits angeführten „Managed Membership“ und dem notwendigen Vertrauen in die Fähigkeit der Gatekeeper ergibt.

Die große Bedeutung des Vertrauens nicht nur zwischen den einzelnen Netzwerkmitgliedern sondern auch zu zentralen Koordinationsinstanzen ist nun in Open-Source-Software-Systemen besonders offensichtlich, als die gesamte Macht der Projektleitung (zur Regelformulierung) allein ihrer Legitimierungsfunktion entspringt. Diese Legitimierungsfunktion wiederum ist aber nun gänzlich vom Vertrauen der Anwender und Entwickler abhängig. Ergebnis eines Vertrauensverlustes bei gleichzeitiger Nachfrage nach dem Produkt/Programm ist denn auch bisweilen ein „Fork“, die Entwicklung einer selbständigen Version mit verschiedener zentraler Koordination (vgl. die verschiedenen BSD-Unix-Versionen OpenBSD, FreeBSD und NetBSD).

Und auch die gemeinsam geteilte Open-Source-Kultur (die sich auch im gemeinsamen Selbstverständnis als „Open-Source-Community“ widerspiegelt, vgl. Morner 2002) spielt für die Kommunikation innerhalb der Entwicklergemeinde eine große Rolle und führt oftmals zu sehr engen Beziehungen untereinander wie auch zu sehr emotional geführten Konflikten (z.B. im Zuge der zahlreichen Spaltungen der BSD-Unix-Entwicklergemeinde).

Ein weiterer Aspekt der kommunikativen Beziehungen zwischen den einzelnen Open-Source-Entwicklern ist die wechselbezügliche Bedeutung von Reputation und Vertrauen: Fachliche Reputation der Entwickler schafft wiederum Vertrauen bei Anwendern und anderen Entwicklern und legitimiert in der Folge Entscheidungen der Entwickler (Morner 2002, S 223). Wie wichtig Reputation für das Funktionieren eines Open-Source-Software-Systems an sich ist, deutet auch Raymonds fundamentale Feststellung „hackerdom runs on reputation“ (2001 „How to become a Hacker“) an, sowie der mit Reputation in der jeweiligen Community verbundene Aufmerksamkeitseffekt („signaling incentive“, Lerner/Tirole 2000, S 15) als wesentlicher motivationaler Faktor für die Partizipation am Netzwerk (vgl. auch Morner 2002, S 223f, Schneidewind et. al. 2002, S 230).

4.1.4 Gesamtnetzwerk

Open-Source-Software-Systeme entwickeln und gestalten sich in überaus großem Maße selbstorganisierend. Dies wird schon bei Morners (2002, S 220) idealtypischer Beschreibung der Entwicklung eines Open-Source-Software-Systems deutlich: Ausgehend vom Problem eines Anwenders oder der Idee eines Entwicklers formiert sich im Rahmen eines ersten Diskussionsprozesses (z.B. in Newsgroups, sonstigen Internet-Foren) ein Projektteam. Findet sich kein bereits existierendes (besseres) Programm und macht ein Entwickler oder eine Gruppe von Entwicklern einen ersten Vorschlag, der unmittelbar nach der Programmierung der Öffentlichkeit zur Verfügung und Diskussion gestellt wird, kann man bereits von einem Open-Source-Software-System sprechen. Auf ähnliche Weise entstand das derzeit größte Open-Source-Software-Projekt Linux (vgl. Wayner 2000, S 67ff / Moody 2001).

Die Leitung eines Open-Source-Projektes kann sich schon definitionsgemäß nur auf Meta-Management beschränken, da keine weisungsgebundenen Programmierer zur Verfügung stehen. Und selbst jegliche vorhandenen Meta-Management-Strukturen (wie die leitenden Projektteams), die beispielsweise Kommunikations- und Schnittstellenstandards festlegen, entstehen selbstorganisierend und stehen zur freien Disposition innerhalb des jeweiligen Projekts.

Was den Grad der Zentralität von Open-Source-Software-Systemen angeht, so fällt eine Beurteilung des Ausmaßes zwischen niedrig und hoch schwer. Der Entwickler Larry McVoy formulierte, in Anlehnung an die berühmte und von Raymond (1999) geprägte Bezeichnung der freien Softwareentwicklung als „Bazar“, die Notwendigkeit einer zentralen Koordinationsinstanz blumiger mit den Worten: „Wenn man genau hinsieht, ist da nirgends ein Bazar. Ganz oben thront immer eine Ein-Mann-Kathedrale.“ (Wayner 2000, S 136) Dass die „Ein-Mann-Kathedrale“ in der Regel aus mehreren Personen besteht, die eine zentrale Koordinationsaufgabe erfüllen, zeigt Apache-Mitbegründer Brian Behlendorf, wenn er die verschiedenen Rollen und deren Aufgaben für die Koordination eines Open-Source-Software-Projektes beschreibt (dazu ausführlich DiBona/Ockman/Stone 1999, S 163f).

Diese zentrale Koordination in Open-Source-Software-Systemen, sei es nun durch eine einzelne Person oder eine Gruppe von Entwicklern, erfüllt in gewisser Hinsicht die Funktion einer „Börsenaufsicht“: Sie ist die Basis dafür, dass sich die Entwickler den von allen geteilten Spielregeln bzw. Kommunikationsstandards unterwerfen. Dadurch erfüllt sie aber eine doppelte Funktion: Einerseits ist die Existenz der zentralen Instanz und ihre

Vertrauenswürdigkeit die Basis für das notwendige Systemvertrauen der einzelnen Systemteilnehmer. Andererseits ergeben sich aus dem kleinsten gemeinsamen Nenner „Systemvertrauen“ Standards, die von allen Teilnehmern des Netzwerks geteilt werden und so einen gemeinsamen, kommensurablen Raum für Austausch und Konfliktbewältigung bilden.

4.2 Abgrenzung zu herkömmlich-proprietärer Softwareproduktion

4.2.1 Komplexität und Modularität von Produkt und Prozess

Das Problem der Komplexitätsbewältigung hat im Softwarebereich, verglichen mit der kurzen Geschichte der Computerprogrammierung an sich, eine schon sehr lange Tradition, hier allerdings vor allem im Bereich der Erstellung von Problemlösungsalgorithmen. Die gestiegene und immer noch stark ansteigende Komplexität im Rahmen der Softwareentwicklung ist hingegen eher jüngeren Datums.

Schinzel (1998, S 3ff) unterscheidet drei Perspektiven der Komplexität im Zusammenhang mit der Erstellung von Software:

- Aus der Perspektive des Software-Engineering: die Komplexität der Systeme, die durch Software teil-modelliert werden sollen, und der im Softwareentwicklungsprozess hervorgerufenen Fehler und Risiken. Systementwicklung, -entwurf und -gestaltung, also Produktion von Software. Hierzu gehört etwa die Komplexitätserhöhung der Aufgabe durch Vorverlegung der Verantwortungsnahe in den Entwicklungsprozess.
- Aus der Perspektive der theoretischen Informatik und der Software-Qualitätssicherung: Komplexität des Produkts Software selbst (statisch: Größe, Modularisierung, Vernetztheit, Schnittstellen, Softwaremetriken; dynamisch: Komplexitätsmaße, Nichtrobustheit des Diskreten)
- Aus der Perspektive der Technikfolgenabschätzung und der Techniksoziologie: Komplexität der Verwendung und der Wirkungen von Software in ihrer Anwendungsumgebung.

Insbesondere aus der Perspektive der Software-Qualitätssicherung und theoretischen Informatik ergibt sich die chaotisch-diskrete Struktur und die damit verbundene irreduzible Komplexität (größerer) Softwareprojekte (Schinzel 1998, S 4f): Aus der Modularisierung der Softwareentwicklung folgt ein gleichzeitiges Arbeiten hunderter Entwickler mit einer unüberschaubaren Anzahl von Schnittstellen. Zusammen mit Änderungen und unterschiedlichen Entwickler-Niveaus kann der Informationsfluss und die Koordination der Beteiligten den vielfältigen Wechselbeziehungen der einzelnen Programmteile nicht gerecht werden. In der Folge führen aber aus Kapazitäts- und Kontrollüberlegungen außer Acht

gelassene, seltene und (scheinbar) unwesentliche Fehler im Zusammenspiel mit der Diskretheit des Mediums Software zu chaotischen, unprognostizierbaren Ergebnissen.

Bei großen Softwareprojekten (wie beispielsweise Betriebssystemen), die eine größere Anzahl von Entwicklern benötigen, stellt diese komplexe Natur von Software und Softwareerstellung große Anforderungen an die Koordinationsinstanz. Bereits 1975 beschäftigte sich Frederick Brooks (DiBona/Ockman/Stone 1999, S 208) mit dem Koordinationsproblem bei Softwareprojekten indem er vorhersagte, dass bei einer Erhöhung der Anzahl der Programmierer um den Faktor N die erbrachte Arbeit zwar auch um Faktor N steigt, aber die Komplexität und die Fehleranfälligkeit um N-Quadrat steigen würde. Er leitete daraus folgendes, nach eigenen Worten „oversimplifying outrageously“, Brooks’s Gesetz ab: „Adding manpower to a late software project makes it later.“ (Brooks 1995, S 25)

Der Lösungsansatz klassisch-proprietärer Softwareerstellung liegt in zentraler Top-Down Modularisierung an Hand eines abstrakten Gesamtsystems. Die sich daraus ergebenden Programmmodule werden von kleinen Entwicklerteams erstellt und anschließend zusammengeführt und getestet (vgl. Tab. 1, Schneidewind et. al. 2002, S 228).

	Klassische Softwareprodukte	Open-Source-Entwicklung
Zugänglichkeit des Quellcodes	Nur für Entwickler / Unternehmensmitglieder	Offen für jeden Interessierten
Größe der Entwicklergruppe	Kleine Entwicklungsteams	Kleine bis sehr große Entwicklergemeinden (z.B. über 3000 Entwickler bei Linux)
Steuerung des Entwicklungsprozesses	Zentral, Top-Down	Dezentral, Bottom-Up
Zeitpunkt des Release	Erst nach umfassenden Tests	Möglichst unmittelbar nach Programmierung
Modularisierung	Modularisierung im Rahmen eines Gesamtsystems	„Bottom-Up“-Modularisierung

Tab.1: **Unterschiede und Gemeinsamkeiten in den Organisationsmerkmalen und Koordinationsmechanismen der Open-Source-Software- und klassischer Software-Entwicklung (nach Schneidewind et. al. 2002, S 227)**

Dass diese herkömmliche Form der Softwareerzeugung und deren finale Softwareerzeugnisse zahlreichen Ansprüchen immer weniger gerecht werden können, weil sie die zunehmende Komplexität multipler und sich ständig verändernder Hard- und Softwareumgebungen immer schwerer bewältigen können, zeigt der folgende Abschnitt.

4.2.2 Grenzen klassischer Softwareproduktion

Versucht man die Qualität und Effizienz und damit auch gleichzeitig die Grenzen einer Softwareerstellungsmethode oder auch nur eines fertigen Softwareprodukts zu beurteilen, muss zuerst ein geeigneter Maßstab hierfür gefunden werden. Selbst wenn nur proprietäre Softwareerstellung untersucht wird, die ja im Unterschied zu Open-Source-Software-Systemen monetäre Kennzahlen aufweist, muss im Vorhinein die Frage, was qualitativ hochwertige Software ist, beantwortet werden.

Ein allgemeines Abstellen auf den Markterfolg – in gewisser Hinsicht auf die Effizienz des Endprodukts - ist im Softwarebereich problematisch, da der Markterfolg beispielsweise im Betriebssystem- oder Office-Anwendungsbereich wesentlich von der Marktstellung abhängt und somit zirkulär selbstverstärkend wirkt. Dieses Phänomen resultiert einerseits aus den hohen Fix- und gegen Null tendierenden variablen Kosten und andererseits aus dem Faktum, dass das Ausmaß der Verbreitung einer Softwarelösung einen Kaufanreiz darstellt.

Denn eine Konsequenz dieses – keineswegs auf Software beschränkten² - „Lock-In“-Phänomens (Arthur 1989) ist der Einfluss des Marktanteils auf die Entscheidung, was eine „gute“ Lösung darstellt: Auch wenn beim Vergleich der Programme A und B als Insellösungen das Programm A die gestellten Anforderungen bezüglich Nutzungseffizienz besser erfüllt, so kann bei einer Gesamtbetrachtung unter Einbeziehung der Kompatibilität mit Kunden/Lieferanten, der Verfügbarkeit von Erweiterungssoftware, Service und geschulten Mitarbeitern sehr wohl B die bei weitem effizientere, zumindest aber profitablere Lösung für ein Problem darstellen.

Letztendlich ist die Frage, ob eine Softwarelösung qualitativ hochwertig ist, im Einzelfall an Hand der gestellten Anforderungen zu prüfen. So auch Weinberg (1998, S 16f) wenn er festhält: „Indeed, we cannot really measure a program by itself and say whether it is good or bad. (...) Looking honestly at the situation, we are never looking for the best program, seldom looking for a good one, but always looking for one that meets the requirements.“

Ein Kriterium, das noch am ehesten den Anspruch erheben kann, als Vergleichsmaßstab zu dienen, ist die Fehleranfälligkeit eines Softwareprodukts. Denn wenn auch auf Grund der irreduziblen Komplexität im Softwarebereich fehlerlose Programme praktisch nicht realisierbar sind und oftmals auch Fehler als solche nicht klar zu erkennen sind („Is it a bug or is it a

² Prominente Beispiele für Lock-In sind beispielsweise das „QWERTZ-“ bzw. QWERTY-“ Tastaturlayout (z.B. Shapiro/Varian 1999, S 185), das VHS-Videorekordersystem und allgemeiner für Industriestandards Sun Microsystems Open System Strategie (Garud/Kumaraswamy 1993)

feature?“ ist ein unter Entwicklern beliebter Spruch), so lassen sich Fehler im Sinne von Abweichungen von der geforderten Funktionsweise sowie deren Häufigkeit feststellen.

Und genau diese von Software geforderte Robustheit in der Form geringer Fehleranfälligkeit, insbesondere in sensiblen Bereichen, wo auf Datensicherheit Wert gelegt wird, ist bei komplexen Programmen mit herkömmlichen Produktionsmethoden immer schwieriger und nur mit großem (finanziellem) Aufwand zu erzielen. Der Sprecher des vor allen anderen Zielen auf Sicherheit bedachten BSD-Unix-Projektes „OpenBSD“, Theo de Raadt, formuliert es wie folgt: „Ein sicheres System ist vollkommen fehlerfrei. Und genau das ist nicht machbar.“ (Bonnert 2003, S 106f)

Die Fehleranfälligkeit von Software in der Form „unprognostizierbarer Ergebnisse“ ergibt sich, wie bereits erwähnt, aus steigender Schnittstellenanzahl und durch die Interdependenz der einzelnen Programmmodule. Raymond (2001, S 36) klassifiziert diese unprognostizierbaren Ergebnisse als „complex multi-symptom errors“. Ihm zu Folge haben diese komplexen Softwarefehler multiple Wege („trace paths“) von den Oberflächensymptomen zu den wirklichen Fehlern, die wesentlich von der jeweiligen Hard- und Softwareumgebung abhängen. Grob vereinfacht, führen mehrere kleine Fehler, die oftmals aus spezifischen Hard- oder Softwarekombinationen im konkreten Einsatzfeld erwachsen, zu viel erheblicheren Fehlern.

Proprietäre Entwickler verfolgen nun aber Fehler auf Grund ihrer - von Testern in einem gezwungenermaßen engen Versuchsfeld - gemeldeten Oberflächensymptome zu ihren Wurzeln im Quellcode zurück. Denn proprietäre Softwareprodukte werden erst lange nach Abschluss der Programmierung und einer umfassenden internen Testphase als fertiges und idealerweise stabiles Produkt veröffentlicht (vgl. Schneidewind et. al. 2002, S 227). Auf Grund der Beschränktheit der finanziellen Ressourcen bei kommerziellen Softwareprojekten sind aber dem Umfang und der Diversität der Testphase bzw. Testumgebung vor dem endgültigen Release enge Grenzen gesetzt. Die Entdeckung von „complex multi-symptom errors“ kann unter diesen Umständen nur sehr schwer in ausreichendem Maß erfolgen, wo noch dazu die Zeit der Testphase aus betriebswirtschaftlichen Überlegungen so kurz wie möglich gehalten wird. Nachträgliche Fehlerkorrekturen durch Updates sind in der Folge zwar möglich und in begrenztem Maße auch üblich, allerdings meist mit Image-Schäden und Kosten verbunden.

4.2.3 Open-Source-Software-Systeme als alternative Form der Softwareprojekt-Koordination

Open-Source-Software-Systeme stellen nun einen alternativen Ansatz für die organisatorische Gestaltung von Softwareprojekten dar, der sich von klassisch-proprietären Softwareerstellungsprozessen doch deutlich unterscheidet (vgl. dazu auch die Übersicht Tab. 1).

Folgt man klassischen Überlegungen und Erkenntnissen der Programmierforschung, dürfte es Open-Source-Software-Systeme wie Linux oder Apache mit ihren großen, weltweit verstreuten Entwicklergemeinden überhaupt nicht geben. So zumindest, wenn man der bereits ausgeführten Überlegung des überproportional steigenden Koordinationsaufwands bei steigender Programmiereranzahl folgt und die Unmenge an Entwicklern bei großen Open-Source-Projekten in Betracht zieht.

Die Existenz von Open-Source-Software-Systemen und die Robustheit zahlreicher Open-Source-Software-Produkte beweist nun aber die große Komplexitätsbewältigungskapazität dieser Organisationsform. Besonders deutlich wird dieser Umstand bei der Analyse der Gründe für die relative Robustheit im Sinne geringerer Fehleranfälligkeit von Open-Source-Software.

Ein wesentlicher Grund für die Robustheit von Open-Source-Software liegt in DiBona et al.'s bereits erwähnter Metapher von der „wissenschaftlichen“ Produktionsmethode: „Replication makes scientific results robust. (...) By sharing source code, Open Source developers make software robust.“ und weiter „The scientific method minimizes duplication of effort because peers will know when they are working on similar projects.“

Open-Source-Software-Entwicklung ermöglicht insbesondere im Bereich der complex multi-symptom errors eine vielfältigere Zufallsauswahl an „trace paths“ durch die größere Anzahl und Vielfalt der (weltweiten) Testumgebung. Die komplexe Struktur von Open-Source-Software Netzwerken bietet in diesem Sinne die Möglichkeit, komplexeren Fehlerstrukturen entgegenzutreten. Mit anderen Worten, die Freigabe des Quellcodes und die Einbindung einer maximalen Anzahl von Personen in den Entwicklungsprozess führt zur Erhöhung der Debugging-Kapazität (vgl. „Linus (*Torvald, der Begründer von Linux, Anm. Dobusch*) Law“: „Given enough eyeballs, all bugs are shallow“ Raymond 2001, S 30) und damit der Stabilität des Endprodukts.

Aber auch abseits der geringeren Fehleranfälligkeit weisen Open-Source-Softwareprojekte besondere Eigenschaften auf: Freyermuth (2001, S 176f) weist auf die „den gesunden Menschenverstand verblüffende Einheitlichkeit und historische Kontinuität der Open-Source-Softwareprojekte“ im an sich schnelllebigen Softwarebereich hin. Für dieses Phänomen macht er die Wirkung von Markt- bzw. Selektionsmechanismen bereits im Entwicklungsbereich verantwortlich, die einen sehr starken Standardisierungsdruck bereits in dieser Phase erzeugen. Im Rahmen der Open-Source-Prozesse konkurrieren alle Projekte um eine begrenzte Zahl von talentierten Entwicklern, was in der Folge zu beschleunigter Selektion unattraktiver (z.B. mit anderer Hard- und Software inkompatibler) Projekte führt.

Und selbst wo zwei oder mehr konkurrierende Ansätze überleben, ist Inkompatibilität allein auf Grund des offenen Quellcodes so gut wie ausgeschlossen, da Unverträglichkeiten – ganz im Unterschied zu proprietärer Software – keinerlei Konkurrenzvorteile, sondern vielmehr sogar Akzeptanznachteile mit sich bringen. Auch die Kontinuität der Entwickler ist im Open-Source-Bereich tendenziell höher und trägt somit zur Dauerhaftigkeit der Projekte bei: Denn während in proprietären Softwareprojekten Mitarbeiterwechsel und –fluktuation Ergebnis klassischer Dienstverhältnisse sind, bleiben Entwickler ihren Open-Source-Projekten tendenziell länger treu (Freyermuth 2001). Prominente Beispiele für die Stabilität und Kontinuität von Open-Source-Projekten lassen sich vor allem im Internet finden, das besonders auf Standardisierung und Dauerhaftigkeit angewiesen ist und daher auf die offenen Protokolle TCP/IP, SMTP oder HTTP aufbaut.

Weitere potentielle Vorteile für Anwender von Open-Source-Software (u.a. BWT Deutschland 2001, S 17f) gegenüber proprietärer Software finden sich in folgender Übersicht:

- Keine Abhängigkeit von einem Monopolanbieter und die damit verbundene Vielfalt an Lösungen sowie Vermeidung des Monopolaufschlages auf den Preis der Nutzungslizenz
- Möglichkeit der selbstständigen Fehlerbehebung und Erweiterung
- Tendenziell größere Freiheit bei der Wahl der Hardwareplattform

Demgegenüber stehen aber auch potentielle Nachteile des Einsatzes von Open-Source-Software:

- Kompatibilitätsprobleme mit aktuellen, proprietären Standards
- Keine ausgereiften Produkte für spezielle Einsatzgebiete

- Tendenziell höhere Ansprüche an Anwenderkenntnissen bzw. geringere Auswahl an geschulten Fachkräften in bestimmten Bereichen

4.3 Empirische Beispiele für Open-Source-Software-Systeme

Wie schon im Bereich der Strategischen Unternehmensnetzwerke sollen auch hier die wichtigsten empirischen Beispiele für Open-Source-Software-Systeme nur kurz angeführt werden und für weitere Informationen sei auf die umfangreichen (online-)Ressourcen im Anhang verwiesen.

Neben dem prominentesten und bereits mehrfach genannten Beispiel für Open-Source-Software, dem Betriebssystem Linux (www.linux.org), das einer GPL-Lizenz unterliegt, ist das Open-Source-Softwareprodukt mit dem größten relativen Marktanteil innerhalb seiner Produktgruppe wohl der Apache Webserver (www.apache.org) unter BSD-Lizenz, mit einem Marktanteil von über 50 % (Netcraft Web Server Survey, 2002, DiBona/Ockman/Stone 1999, S 9).

Mit Linux konkurrieren des weiteren im Bereich der Server-Software die ebenfalls bereits erwähnten verschiedenen Varianten von BSD-Unix-Systemen (Wayner 2000, S 345ff): FreeBSD (www.freebsd.org), die populärste Version von BSD-Unix, versucht vor allem ein leicht bedienbares und frei verfügbares Betriebssystem zu entwickeln. NetBSD (www.netbsd.org) wiederum legt ein größeres Augenmerk auf die Unterstützung einer maximalen Anzahl von (bisweilen auch exotischen) Hardware-Plattformen. OpenBSD (www.openbsd.org) letztlich verfolgt das Ziel maximaler Betriebssicherheit.

5 Strategische Netzwerkorganisation und Innovation

Bis hierher wurden Strategische Unternehmungsnetzwerke und Open-Source-Software-Systeme vor allem vom Standpunkt der Organisation bzw. Koordination des Leistungserstellungsprozesses her analysiert und beschrieben. In der Folge sollen hier nun die Auswirkungen der Organisationsform Strategische Netzwerkorganisation auf Innovationsprozesse beleuchtet werden, wieder beginnend mit Strategischen Unternehmungsnetzwerken.

5.1 Innovationsprozesse in Strategischen Unternehmungsnetzwerken

5.1.1 Parallelität des Innovationsprozesses in Strategischen Unternehmungsnetzwerken

Wie bereits ausführlich besprochen, werden im Rahmen eines Strategischen Unternehmungsnetzwerks zahlreiche Teilleistungen zu einer integrierten (Gesamt-)Leistung zusammengeführt. Als wesentlicher Vorteil gegenüber vertikal integrierten Unternehmen stellte sich in diesem Zusammenhang die (wirtschaftliche) Autonomie und die Spezialisierung der einzelnen Leistungserbringer heraus (Bauer 2002, S 20 / Jarillo 1993, S 17).

Diese beiden Faktoren sind es nämlich vor allem, die eine hohe zeitliche Parallelität der Innovationsprozesse innerhalb des Unternehmungsnetzwerks und in der Folge eine hohe Entwicklungsdynamik bzw. -geschwindigkeit ermöglichen: Denn ein gleichzeitiges Forschen an sämtlichen Leistungsmodulen und Leistungsprozessen erscheint in integrierten Unternehmen schwer zu realisieren und zu finanzieren. Insbesondere bei steigender Modularisierung wären Komplexität und Aufwand zu hoch, sich auf jedes einzelne Modul und jeden einzelnen Prozess mit gleicher Aufmerksamkeit zu konzentrieren.

Im Strategischen Unternehmungsnetzwerk können Innovationsprozesse dagegen im überschaubaren Rahmen der einzelnen Aktivitäten in den jeweiligen selbstständigen Netzwerkbetrieben ablaufen. So kommen die oben angeführten Effizienzvorteile kleinerer wirtschaftlicher Einheiten (vgl. Jarillo 1993, S 17) eben gerade auch den Innovationsprozessen zu gute.

Im Unterschied zu einer Outsourcing-Strategie soll aber das gewonnene Wissen der Betriebe im Netzwerk erhalten bleiben. So sieht auch Sydow (1992, S 91) einen wesentlichen Vorteil in

der Erhaltung und Entwicklung des auf Produkt- und Prozessgestaltung bezogenen Innovationspotentials der Zulieferbetriebe in Strategischen Unternehmungsnetzwerken, wenn auch stark aus Sicht einer fokalen Unternehmung: „Im Sinne einer systemischen Rationalisierung soll die Produktivität und Profitabilität auf allen Stufen der Wertkette (...) gesteigert und im Anschluss im Sinne einer systemischen Beherrschung durch die das Netzwerk führende Unternehmung angeeignet werden.“

Parallelität von Innovationsprozessen gibt es aber in Strategischen Unternehmungsnetzwerken nicht nur in zeitlicher sondern auch in qualitativer Hinsicht, sofern mehrere Betriebe gleiche oder ähnliche Aufgaben wahrnehmen. Im Unterschied aber zu reinen Funktionsexternalisierungs-Strategien, bei der durch regelmäßige Neuausschreibungen langfristige Innovationsstrategien nicht unbedingt belohnt werden, schafft die erhöhte Bestandgarantie der Netzwerkumgebung dafür Raum. Diese Form der qualitativen Parallelität des Innovationsprozesses, die aus bewusst geschaffenen Redundanzen innerhalb des Netzwerks resultiert, schafft einerseits netzwerkinternen Wettbewerb und birgt die prinzipielle Voraussetzung für Wissenstransfer auf horizontaler Ebene des Wertschöpfungsprozesses in sich. Die Vorteile redundanten Wissens für Innovationsprozesse innerhalb eines strategischen Unternehmungsnetzwerks sind nun Thema des folgenden Abschnitts.

5.1.2 Redundantes Wissen als Grundlage für Innovationsvorteile in Strategischen Unternehmungsnetzwerken

Neben den Innovationsvorteilen, die mit größerer Ressourceneffizienz in kleineren selbstständigen Einheiten und höherer Komplexitätsbewältigungskapazität der selbstorganisierenden Netzwerkstruktur einhergehen, steht Strategischen Netzwerkorganisationen noch ein weiteres Feld potentieller Innovationsvorteile offen: netzwerkinterner Wissensaustausch.

Konkret besteht die Möglichkeit durch Austausch von Wissen unter den einzelnen autonomen Netzwerkteilnehmern innerhalb des Netzwerks eine „Verwissenschaftlichung“ des Innovationsprozesses zu erreichen, der wiederum einen relativen (Wettbewerbs-) Vorteil gegenüber netzwerkexternen Unternehmen darstellt. Die Freigabe und der Austausch von Know-how führt zur Erhöhung des im Netzwerk redundant vorhandenen Wissens und ermöglicht somit den einzelnen Netzwerkteilnehmern, weitere Innovationsschritte auf höherem Niveau zu beginnen. Wie in (idealtypischen) wissenschaftlichen Forschungsprozessen sorgt die Zugänglichkeit von Innovationen einerseits für bessere Überprüfung bzw. schnellere

Falsifizierung suboptimaler Ergebnisse und andererseits können weitergehende Entwicklungen auch auf fremden Erkenntnissen aufbauen. Die Folge sind schnellere Innovationszyklen und eine höhere Innovationsqualität.

Die Autonomie der Mitglieder und die Komplexität menschlicher Organisationen haben in Verbindung mit dem mehrfachen Vorhandensein gleicher Informationen in verschiedenen Teilen des Netzwerks vielfältigere Entwicklungsergebnisse zur Folge, weil dadurch verschiedenere Sichtweisen und Entwicklungspotentiale auf eine Information angewendet werden können. Folgt man nun dieser Überlegung, steigert sich das Entwicklungspotential des Gesamtnetzwerks in dem Maße, in dem die Redundanz zunimmt (vgl. für Organisationen Probst 1987, S 81). Der Austausch erhöht somit die Effektivität und das Innovationspotential des Netzwerks in seiner Gesamtheit und stellt mittelfristig durch die Wechselseitigkeit des Austauschs zwischen den Netzwerkmitgliedern auch für die einzelnen teilnehmenden Betriebe einen relativen Vorteil gegenüber der Netzwerkwelt dar.

Dass dieser wissenschaftliche Innovationsprozess und dessen Vorzüge auch in Netzwerken selbstständiger und bisweilen konkurrierender Unternehmen funktionieren kann, zeigt das Beispiel von Toyota. Schon im Titel der Referenzstudie von Dyer/Nobeoka (2000) kommt mit der Bezeichnung des Toyota-Netzwerks als „Knowledge-Sharing Network“ die Bedeutung des Wissenstransfers und –austauschs für den relativen Innovations- und damit Wettbewerbsvorteil zum Ausdruck. Wie entscheidend der Netzwerkcharakter für die Erzielung von Produktivitätsfortschritten durch Innovationsprozesse ist, zeigt, dass nur die Einführung von „Lean Production“ durch US-Automobilhersteller als Antwort auf japanische Produktivitätsfortschritte ohne gleichzeitigen Aufbau einer stabilen Netzwerkwelt nicht den gewünschten Produktivitätsanstieg mit sich brachte (Dyer/Nobeoka 2000, S 346).

In der Terminologie der Spieltheorie betrachtet, bilden in diesem Sinne Strategische Netzwerke den Rahmen - und „erschaffen“ es in gewisser Hinsicht - eines begrenzten Nichtnullsummenspiels (vgl. Axelrod 1991, S 27) für die Netzwerkmitglieder, das Kooperation durch Wissensaustausch ermöglicht und sinnvoll macht.

Die Bedeutung von Wissensfreigabe in Innovationsprozessen lässt sich aber nicht nur innerhalb eines Strategischen Unternehmensnetzwerks beobachten, wie das Beispiel von Sun Microsystems (Garud/Kumaraswamy 1993) im speziellen und das Beispiel breiter Industriestandards im allgemeinen beweisen. Der „Vorteil“ besteht in diesem Sinne aber in der Regel für diejenigen, deren Kooperation zur Durchsetzung eines zumindest Quasi-Standards

bei konkurrierenden Technologien („competing technologies“, Arthur 1989) führt, die also in gewisser Hinsicht ein wiederum gemeinsames, wenngleich auch oftmals loses, Netzwerk bilden. Daher auch der häufig in diesem Zusammenhang erwähnte Begriff der „network externalities“ (u.a. Katz/Shapiro 1985).

5.1.3 Freigabe und Austausch von Wissen in Strategischen Unternehmensnetzwerken

Die eben beschriebenen Vorteile des netzwerkinternen Wissensaustauschs erfordern nun aber in der Praxis erhebliche Anstrengungen, die den Wissensaustausch ermöglichen und forcieren. Denn Wissenstransfer und die Frei- und Weitergabe von explizitem und implizitem Wissen im Rahmen des organisationalen Lernens stellt schon in der klassischen, auf einzelne Unternehmen fokussierten Organisationsforschung ein breites Problemfeld dar (vgl. Schreyögg 1999, S 535f). Von den zwei größten Schwierigkeiten des Wissenstransfers – der Wissensvermittlung einerseits und der Wissensfreigabe andererseits – ist aber im Hinblick auf Netzwerkorganisationen insbesondere zweiteres von entscheidender Bedeutung:

Denn das Wissen soll ja innerhalb des Netzwerks und damit zwischen verschiedenen, teilweise miteinander konkurrierenden, Organisationen ausgetauscht werden, was naturgemäß noch größere Hemmschwellen vermuten lässt als bei Wissenstransfer innerhalb eines vertikal integrierten Unternehmens. Darüber hinaus ist die Freigabe von Wissen in einem Unternehmensnetzwerk nicht (notwendigerweise) mit einer unmittelbaren Gegenleistung und damit einer unmittelbaren wechselseitigen Vorteilhaftigkeit (reciprocal viability) verbunden (Ortmann 1995, S 79).

Die erste angesprochene Schwierigkeit der reinen Wissensvermittlung, also nach der Entscheidung zur prinzipiellen Wissensfreigabe, resultiert aus zahlreichen praktischen Problemen, wie insbesondere wenig kodifiziertem Know-how bzw. oft impliziten Charakter von Wissen und den damit verbundenen Transferschwierigkeiten. Diese Wissenstransferprobleme stellen sich allerdings in gleichem Maße in integrierten, multinationalen Unternehmen und sind keineswegs spezifisch für Strategische Unternehmensnetzwerke. Dementsprechend sind Lösungsansätze wie Personalrotationen, Forschungsvisiten oder gezielte und geförderte Kodifikation von Know-how keineswegs netzwerkspezifisch und es soll hier nicht weiter darauf eingegangen werden.

Die größere Hürde bei der praktischen Umsetzung des Wissensaustauschs im Unternehmensbereich dürfte daher weniger in der Schwierigkeit des tatsächlichen Transfers,

als vielmehr in der weit verbreiteten Abneigung, sowohl gegen Annahme fremden Wissens („not-invented-here Syndrom“, Jarillo 1993, S 69) als auch gegen die Freigabe eigenen Wissens, liegen.

Insbesondere die Weigerung eigenes Wissen mit anderen Unternehmen zu teilen, ist aus der herkömmlichen ökonomischen Sichtweise heraus verständlich: Es ist das spezifische Wissen über die eigenen effizienten Handlungsmuster, die der eigenen Organisation gegenüber Mitbewerbern ihren Vorteil, mehr noch ihre Existenzberechtigung gibt. Wäre dieses Wissen allgemein bekannt, so bestünde keine Notwendigkeit für die Existenz des konkreten Unternehmens mehr.

In Strategischen Unternehmungsnetzwerken stimmt diese Aussage immer noch, allerdings im Idealfall nur noch gegenüber netzwerkexternen Mitbewerbern. Innerhalb des Netzwerks muss sichergestellt werden, dass sich eine Organisation durch Preisgabe ihres spezifischen Wissens nicht selbst ihrer Existenzberechtigung begibt und klarerweise müssen die einzelnen Organisationen auch auf die ausreichende Existenzsicherheit vertrauen können. Notwendig ist ein Vertrauen auf das Gesamtsystem „Strategisches Unternehmungsnetzwerk“ in dem Sinne, dass von gegenseitiger Wissenspreisgabe letztendlich alle im Netzwerk profitieren. Dyer/Nobeoka (2000, S 357) beschreiben dieses Vertrauen im Fall von Toyota als besonderes Selbstverständnis der Netzwerkmitglieder: „Toyota (the nodal firm) uses network-level processes to encourage and educate members to believe that they are part of an interdependent economic community.“

Die Etablierung eines effizienten und effektiven Wissenstransfers innerhalb des Netzwerks hat nun prozesshaften Charakter: Das für die Kooperation notwendige Vertrauen, mit ihm die Kooperation und in der Folge das Netzwerk in seiner Gesamtheit, stabilisieren und entwickeln sich zirkulär in rekursiver Selbstverstärkung (Ortmann 1995, S 78f). Aufgabe des Netzwerkmanagements ist es nun, Strukturen zu schaffen, die diesen rekursiven Prozess ermöglichen und stützen und Opportunismus und Trittbrettfahrertum („free riding“) erschweren und sanktionieren.

Im Beispiel Toyota geschieht dies nun über nach Intensität und Vertrauensgrad abgestufte „network-levels“, die durch konzentrische Kreise versinnbildlicht werden können: Zulieferbetriebe müssen, um von den Vorteilen eines intensiveren und umfassenderen Wissensaustauschs und direkter Hilfestellung von Toyota im innersten „network-level“ (der sogenannten „core group“) zu profitieren, sich zuerst in den beiden äußeren „network levels“

als verlässlich erwiesen haben. Versuchen nun einzelne Unternehmen die Regelungen, beispielsweise die Verpflichtung zur Preisgabe von Fertigungswissen, zu umgehen, stellt der zukünftige Ausschluss von der Partizipation eine bedrohliche Sanktion dar, da einmaliges „free riding“ die Nachteile eines dauerhaften Ausschlusses vom Netzwerk kaum aufwiegen dürfte.

Das erhöht die Kosten für free riding enorm und schafft Raum und Zeit für den angesprochenen rekursiven Kooperationsprozess. Auch Dyer/Nobeoka (2000, S 364) nennen den positiven Effekt dieses Regelwerks auf die Motivation zur Freigabe von Know-how und die wirksame Vermeidung von Trittbrettfahrertum als zwei der drei wesentlichen Leistungen des Toyota-Netzwerks.

So ist für den Fall des Strategischen Unternehmungsnetzwerks von Toyota die Herausbildung einer „Austauschkultur“ unter den Netzwerkunternehmen eine wesentliche Aufgabe des Anreizsystems, das von Toyota implementiert wurde: Der auf den ersten Blick kontraintuitive Austausch von Wissen mit potentiellen Konkurrenzunternehmen erhält durch die Rahmung des strategischen Netzwerks erst seine rationale Basis.

Zusätzlich zum Systemvertrauen auf ein strategisches Netzwerk im obigem Sinne, ist bei Uzzi und seinem Konzept der „strong ties“ vor allem das Vertrauen auf kooperatives Verhalten des konkreten Austauschpartners angesprochen. Wie wesentlich dieses Vertrauen für die Effizienz des Netzwerks ist, streicht auch Jarillo (1993, S 135) heraus: „Being able to generate trust, then, is the key to reducing transaction costs, thus making the existence of a strategic network economically feasible.“ Im Idealfall ist es das Regelwerk des Gesamtnetzwerks, das auch für die individuellen Beziehungen der Netzwerkmitglieder einen vertrauensvollen Rahmen schafft, der die Entwicklung enger Beziehungen ermöglicht und verstärkt.

5.1.4 Ebenen des Wissensaustauschs im Strategischen Unternehmungsnetzwerk

Auch wenn die vorhergehenden Ausführungen über redundantes Wissen in Strategischen Unternehmungsnetzwerken die Vorteile von Wissensaustausch auch oder gerade zwischen Betrieben mit ähnlichen oder gleichen Aufgaben deutlich machen, so stellt gerade dieser Fall in empirischen Beispielen noch die Ausnahme dar. So rekurriert auch Sydow (1992, S 91f) in seinen Ausführungen auf den Begriff der „kontrollierten Autonomie“ und versteht den Wissensaustausch in Strategischen Netzwerkorganisationen vor allem als Mittel der „führenden Unternehmung“ zum Zwecke der Kontrolle des Know-hows „auf allen Stufen der Wertkette“. Und in Jarillos (1993) Beispiel Benetton passiert der Wissenstransfer ebenfalls nicht zwischen

konkurrierenden Netzwerkmitgliedern sondern vor allem auf vertikaler Ebene des Leistungserstellungsprozesses.

In der Regel passiert somit der Wissensaustausch im Rahmen zweier kommensurabler Bereiche (z.B. die jeweiligen Fertigungsabteilungen) von nicht unmittelbar miteinander in Konkurrenz stehenden - weil auf verschiedenen Stufen der Wertschöpfungskette angesiedelten - Betrieben.

Das auch in dieser Hinsicht von Dyer/Nobeoka (2000, S 348, 352ff, 364) sehr detailliert beschriebene Beispiel von Toyota gibt aber Anhaltspunkte dafür, dass diese vertikale Ebene des Wissensaustauschs kein Dogma ist: Durch das bereits beschriebene, ausgeklügelte System der „network-level processes“ in Verbindung mit einer starken, von allen Netzwerkfirmen geteilten, Netzwerkidentität und –kultur wird Austausch auch zwischen Firmen mit sehr ähnlichen Aufgaben forciert. Auch wenn direkte Wettbewerber nicht in den gleichen Austauschgruppen plaziert werden, so ermöglicht Toyotas „Knowledge-Sharing Network“ doch ein sehr hohes Maß an redundantem Wissen mit all den positiven Auswirkungen auf die netzwerkinternen Innovationsprozesse.

5.2 Innovationsprozesse in Open-Source-Software-Systemen

5.2.1 Koordinationskapazität als Innovationsvorteil

Eine Besonderheit bei der Betrachtung von Open-Source-Software-Entwicklung ist die schöpferische Qualität von Softwareprogrammierung an sich. Idealtypisch betrachtet sind Lösungen für Softwareprobleme innovative Akte. Dies entspricht auch DiBona et. al.'s Wissenschaftsmetapher, die Softwareentwicklung als per se innovative Prozesse begreift.

Kann in der Folge eine größere Anzahl an Programmierern aktiv in die Softwareentwicklung eingebunden werden ohne zu einer Überforderung der Projektkoordination zu führen, wie das in Open-Source-Software-Projekten möglich ist (Morner 2002 / Schneidewind et. al. 2002), erhöht sich automatisch das innovative Potential. Gleichzeitig steigt, wie bereits in Kapitel 4.2.3 ausführlich beschrieben, die Robustheit des Softwareprodukts.

Diese Vorteile im Entwicklungsprozess von Open-Source-Software, die sich aus der Koordinationskapazität der Netzwerkstruktur ergeben, treten aber neben dem innovativen Potential zurück, das in der Offenlegung des Quellcodes liegt.

5.2.2 Offener Quellcode als Basis für Innovationsvorteile von Open-Source-Software-Systemen

Nach DiBona et al. (1999, S 6f) ist der größte Kooperationsvorteil bei der Offenlegung von Quellcode bei Open-Source-Software-Systemen die gesteigerte Innovationsfähigkeit. DiBona et al. erklären diese wiederum über den bereits angestrebten Vergleich von Open-Source-Softwareentwicklung mit Wissenschaft: „Where scientists talk of replication, Open Source programmers talk of debugging. Where scientists talk of discovering, Open Source programmers talk of creating.“

Wenn DiBona et al. (1999, S 7) für Open-Source-Software-Entwicklung die klare Parallele zum wissenschaftlichen Forschungsprozess ziehen, ergeben sich drei entscheidende Vorteile des wissenschaftlichen, auf Austausch basierenden Innovationsprozesses:

- Erstens reduziert sich der „reinventing the wheel“-Effekt klassischer proprietärer Softwareentwicklung (und zwar sowohl bei Betrachtung eines einzelnen proprietären Softwareprojekts als auch bezogen auf Closed-Source-Entwicklung als Ganzes) und fokussiert damit die vorhandenen Entwicklungskapazitäten auf „echte“ Innovationsleistungen.
- Zweitens erhöht sich die Innovationsgeschwindigkeit, da durch die austauschbedingte Redundanz eine größere Anzahl von „Forschungseinheiten“ auf höherem Wissensniveau weitere Innovationsschritte setzen kann.
- Drittens erhöht sich die Qualität der Ergebnisse, da sie mehrfach unter verschiedenen Bedingungen getestet (in Popper'schem Sinne „falsifiziert“) werden.

Sämtliche dieser hier aufgezählten Vorteile des wissenschaftlichen Innovationsprozesses haben, wie schnell zu erkennen ist, ihre Entsprechung im Bereich der Strategischen Unternehmungsnetzwerke. Anscheinend sind sie im Open-Source-Bereich aber offensichtlicher, weil die Idee der freien Verfügbarkeit des Quellcodes an sich dieser Überlegung entstammt und wahrscheinlich auch deshalb, weil Quellcode als explizierte ausschließlich geistige Anstrengung den Vergleich mit wissenschaftlichen Erkenntnissen nahelegt.

Gegenüber dem Wissenstransfer in Strategischen Unternehmungsnetzwerken haben Open-Source-Software-Systeme weiters den Vorteil, dass sie sich einer gemeinsamen Sprache, der Programmiersprache des Quellcodes, bedienen. Zusammen mit dem (noch) relativ geringen

Grad an Spezialisierung im Softwarebereich ergibt sich, dass in Open-Source-Software-Systemen über weite Strecken Kommensurabilität herrscht.

Daraus folgt nun, dass bei Open-Source-Softwareprojekten die Frage der Redundanz und deren Erzeugung nicht in vergleichbarem Maße problematisiert ist. Auch wenn der Explizitätsgrad und die Nachvollziehbarkeit des Wissens in Form von Quellcode je nach Ausführlichkeit der Dokumentation schwankt, so ist der Bereich impliziten Wissens im Bereich des Quellcodes sehr eingeschränkt. Klarerweise größer ist der Bereich impliziten Wissens bei den Fähigkeiten der einzelnen Entwickler, wobei aber hier autodidaktische Lernmethoden die Regel sind.

Im allgemeinen liegt das Wissen in expliziter Form (dem Quellcode) vor, ist somit leicht transferier- und erweiterbar. Das Ausmaß an redundantem Wissen in Open-Source-Software-Systemen und den damit verbundenen positiven Auswirkungen auf die im Netzwerk vorhandene Entwicklungskapazität kann also als äußerst groß bezeichnet werden.

5.2.3 Freigabe und Austausch von Wissen in Open-Source-Software Systemen

Nach Analyse der Innovationsvorteile, die in der grundsätzlichen Freigabe und der Möglichkeit des Austauschs von Wissen in der Form von Quellcode bestehen, verdient die Art und Weise dieses Austauschs sowie die Motivation der Entwickler, ihre Arbeitsleistung zur Verfügung zu stellen, nähere Betrachtung.

Ein Vorteil und gleichzeitig die Voraussetzung der globalen Entwicklungszusammenarbeit ist die bereits angesprochene, relativ einfache Möglichkeit des Wissenstransfers im Rahmen virtueller Netzwerke, idR über bestimmte Internetforen und -server. Die Regeln dieses virtuellen Wissensaustausches und in der Folge der Umgang mit dem Trittbrettfahrer-Problem folgen in Open-Source-Software-Systemen aus der Open-Source-Lizenz: Sie gibt vor, welche Nutzungsmöglichkeiten am gemeinschaftlich erstellten Quellcode bestehen, zum Beispiel ob die Verwendung in proprietären Softwareprojekten erlaubt (z.B. BSD) oder untersagt (z.B. GPL) ist. Die Sanktionen für vereinbarungswidriges Verhalten sind neben rechtlichen Konsequenzen vor allem Rufschädigung des Verstoßenden und ein schlechtes Image der von ihm erzeugten Software.

Auf Seiten der individuellen Partizipationsanreize machen Schneidewind et al. (2002, S 229ff) die folgenden vier Hauptanreize aus (vgl. Tab. 2): Das Hoffen auf Reziprozität, die Reputation für die geleisteten Beiträge, der Sinn für Effektivität und die hohe Verbundenheit mit einer Gruppe.

Anreiz	
Hoffen auf Reziprozität	Sowohl unmittelbar (Mitglieder der konkreten Gruppe werden mir bei meinen Problemen helfen) als auch mittelbar (Hilfe aus der Open-Source-Community bei allfälligen Problemen wird erwartet)
Reputation	Hochstehende Beiträge bauen fachliche Reputation mit der Folge unmittelbarer Anerkennung und evtl. späterer Verwertbarkeit im Bereich proprietärer Anstellungsverhältnisse (vgl. auch Lerner/Tirole 2000)
Sinn für Effektivität	Befriedigung durch das Gefühl mit dem eigenen Beitrag viel und sichtbar Wirkungen erzielen zu können
Hohe Verbundenheit mit einer Gruppe	Motivation zum Beitrag durch die Verbundenheit mit der Gruppe

Tab. 2: Motivationen für Beiträge an Open-Source-Software-Projekte (nach Schneidewind et al. 2002, S 229)

Im Bereich der mit den Leistungen verbundenen Reputation in der jeweiligen Community ist nun der mit ihr verknüpfte Aufmerksamkeitseffekt („signaling incentive“, Lerner/Tirole 2000, S 15) nicht zu vernachlässigen. Prestigeträchtig sind nur Lösungen, die von den Nutzern akzeptiert werden. Diese Form der Qualitätskontrolle und des Wettbewerbs ist aber auch einer der Gründe, warum Open-Source-Software bisher vor allem im Profi-Server-Bereich (bspw. Apache- und Linux Server) reüssiert hat: Die Maßstäbe, an denen sich die Beiträge messen müssen, bilden die Ansprüche vor allem der Entwicklergemeinde, die meistens stark von Endbenutzeransprüchen differieren.

Gemeinsam mit den angeführten vier Anreizen bzw. teilweise aus ihnen erwachsend spielt in Open-Source-Software-Systemen die gemeinsame Kultur eine große Rolle, allerdings ist hier die Errichtung und Aufrechterhaltung dieser Kultur keine wesentliche Aufgabe zentraler Koordinationsstellen.. Im Gegensatz zu Strategischen Unternehmungsnetzwerken ist die „Austauschkultur“ selbst - als Teil einer umfassenderen „Hacker-Ethik“ (z.B. Raymond 2001, „How to Become a Hacker?“) - ein wesentlicher Grund für die Partizipation am Netzwerk.

Open-Source-Software-Systeme haben es hier strukturell leichter als Strategische Unternehmungsnetzwerke: Open-Source-Software-Systeme organisieren ja gerade den Teil der Programmierer, der der Idee und der Kultur des „code sharings“ anhängt und auf Erfolg und Effizienz der wissenschaftlichen Methode der Softwareproduktion vertraut. So haben Regelungen in Open-Source-Software-Lizenzen, die auf eine Verhinderung von „free riding“ durch Verwendung von freiem Quellcode in proprietären Programmen abzielen, und die mit ihnen verbundenen rechtlichen Sanktionsmöglichkeiten meist nur deklarativen Charakter und kommen nur selten wirklich zur Anwendung: Die Stärke der Kultur hinter Open-Source-Software, die von der großen Mehrheit der Netzwerkmitglieder akzeptiert und exekutiert wird, befördert die viel effektivere Gefahr der sozialen Ächtung für den Fall des Missbrauchs.

5.2.4 Ebenen des Wissensaustauschs in Open-Source-Software-Systemen

Im Unterschied zu Strategischen Unternehmungsnetzwerken, wo große Teile des Wissenstransfers auf vertikaler Ebene des Wertschöpfungsprozesses stattfinden, erfolgt der Wissenstransfer im Open-Source-Bereich nahezu ausschließlich auf der horizontalen Ebene der Softwareproduktion. Im nachgelagerten Feld des Vertriebs von Open-Source-Softwareprodukten beispielsweise ist Wissenstransfer zwischen den einzelnen Anbietern völlig unüblich.

Dieser radikal horizontale Wissenstransfer ist aber in der konkreten Form vor allem deshalb möglich, weil zwischen den einzelnen Netzwerkteilnehmern kein wirtschaftlicher Wettbewerb besteht. Zwar gibt es einen Wettbewerb um Reputation, da sich die Reputation aber gerade aus dem Vergleich der Programmierleistungen miteinander ergibt, erfordert dieser Wettbewerb ebenfalls den Wissensaustausch. Hier wird eine weitere Parallele zum wissenschaftlichen Forschungsprozess offensichtlich: Schließlich besteht zwischen Forschern ebenfalls ein Wettbewerb um Anerkennung (z.B. in der Form von Veröffentlichungen in Zeitungen), aber zumindest kein unmittelbarer wirtschaftlicher Verdrängungswettbewerb.

5.3 Horizontaler Wissensaustausch als ein wesentlicher Grund für Innovationsvorteile in Strategischen Netzwerkorganisationen

Betrachtet man nun die Innovationsprozesse in Strategischen Unternehmungsnetzwerken und in Open-Source-Software-Systemen gemeinsam, so fällt im Rahmen der zahlreichen Übereinstimmungen – insbesondere die Verwissenschaftlichung des Innovationsprozesses betreffend – die unterschiedlich starke Ausprägung horizontalen Wissenstransfers auf.

Durch die Explizität des Wissens und die totale Horizontalität des Wissensaustauschs kommt die Open-Source-Software-Entwicklung den Prozessen wissenschaftlicher Forschung schon sehr nahe. Dementsprechend ist auch eine bei steigender Komplexität immer stärker zu Tage tretende größere Robustheit von Open-Source-Software im Vergleich zu proprietärer Software zu beobachten. Dieser qualitative Vorteil erlaubt es Open-Source-Software wiederum trotz ihrer auf freiwillige und unentgeltliche Beiträge angewiesenen Struktur gegenüber kommerziellen Mitbewerbern zu reüssieren.

Im Bereich der Unternehmungsnetzwerke ist die Verwissenschaftlichung von Innovationsprozessen erst in ihren Grundzügen zu erkennen. Dies dürfte wohl auch daran liegen, dass Strategische Unternehmungsnetzwerke auch abseits ihres großen (und in der Regel

ungenutzten) Potentials im Bereich der Innovationsprozesse zahlreiche weitere Vorteile gegenüber den herkömmlichen Modellen marktlicher oder vertikal integrierter Leistungserbringung für sich verbuchen können.

In diesem Zusammenhang sticht wiederum die Sonderrolle Toyotas als sehr erfolgreiches Exemplar unter den zahlreichen empirischen Beispielen für Strategische Unternehmungsnetzwerke hervor: Denn es ist das Toyota Knowledge-Sharing-Network, das durch seine ausgeprägte Netzwerkidentität und sein effektives Regelwerk der „network-level processes“ horizontalen Wissenstransfer zumindest ansatzweise ermöglicht. Mit anderen Worten, die empirischen Beobachtungen das Toyota-Netzwerk betreffend, deuten darauf hin, dass es als eines der wenigen empirischen Beispiele dabei ist, die potentiellen Innovationsvorteile von Strategischen Netzwerkorganisationen stärker auszuschöpfen.

Aber auch im Toyota-Netzwerk schreckt man (noch) vor Wissensaustausch zwischen unmittelbaren Konkurrenten bzw. Betrieben mit identen Aufgaben und somit echt horizontalem Austausch zurück (Dyer/Nobeoka 2000, S 355). Bei einem funktionierenden Regelsystem und ausreichender Existenzsicherung für die beteiligten Unternehmungen spricht allerdings zumindest theoretisch nur wenig gegen diesen horizontalen Austausch auch im Bereich von Strategischen Netzwerkorganisationen. Das sich im Bereich der Open-Source-Softwareentwicklung und in Ansätzen bereits im Toyota-Netzwerk abzeichnende innovative Potential eines derartigen horizontalen Wissensaustauschs sollte jedenfalls Anlass genug für Überlegungen in diese Richtung sein.

Abseits dieser konkreten Überlegungen ergeben sich aus der Erkenntnis des Potentials horizontalen Wissenstrfers bzw. redundanten Wissens in wissenschaftlichen oder zumindest wissenschaftsnahen Forschungsprozessen auch allgemeine rechtliche und politische Fragen. Insbesondere die Beantwortung der Fragen nach Umfang, zeitlicher Dauer und Regelungsgegenstand von Patenten und sonstigen Urheberrechten, hat auch wesentliche Auswirkungen auf zukünftige Innovationsprozesse (vgl. die Auswirkungen der Zulassung von Softwarepatenten auf die weitere Entwicklung von Open-Source-Software-Projekten, um nur ein Beispiel zu nennen). Die Betrachtung der Innovationsprozesse in Strategischen Netzwerkorganisationen regt jedenfalls zur Überlegung an, dass Innovationsprozesse durch den Verzicht auf geistiges Eigentum in bestimmtem Umfang durchaus verstärkt werden können.

Literatur

Anderson, Philip W./Arrow, Kenneth J./Pines, David (1988): The Economy as an Evolving Complex System: The Proceedings of the Evolutionary Paths of the Global Economy Workshop, Held September, 1987 in Santa Fe, New Mexico. Addison-Wesley Publishing Company: Redwood City

Arthur, Brian W. (1989): Competing Technologies, Increasing Returns, and Lock-In by Historical Events. In: The Economic Journal, 99, S 116-131

Augier, Mie/Simon, Herbert A. (2003): The Architecture of Complexity (Commentary). In: Garud, Raghu/Kumaraswamy, Arun/Langlois, Richard N. (Hrsg.) (2003): Managing in the Modular Age: Architectures, Networks, and Organizations. Blackwell Publishers Ltd.: Oxford, S 38-44

Automotive Intelligence News, <http://www.autointell.com/News-2001/November-2001/November-2001-1/November-07-01-p02.htm> (letzter Besuch: 15.3.2003)

Axelrod, Robert (1991): Die Evolution der Kooperation. 2. Auflage, R. Oldenbourg Verlag: München

Bauer, Robert (2002): Effektivität und Effizienz in Netzwerk-Organisationen: Unterwegs zu einer epistemologischen Theorie der Organisation. In: Weiskopf, Richard (2003): Menschenregierungskünste: Anwendungen poststrukturalistischer Analyse auf Management und Organisation. Westdeutscher Verlag: Wiesbaden

Boos, F./Exner, A./Heitger, B. (1992): Soziale Netzwerke sind anders. In Organisationsentwicklung, 11. Jahrgang, 1, S 54-61

Bonnert, Erich (2003): „Wir machen keine Politik, wir schreiben Software“: Theo de Raadt über Zukunft und Sicherheit von OpenBSD. In: c't Magazin für Computer und Technik 13/2003, S 106-107)

Brooks, Frederick P. Jr. (1995): The Mythical Man-Month: Essays on Software Engineering - Anniversary Edition. Addison-Wesley: Boston

Bundesministerium für Wirtschaft und Technologie (Deutschland/Hrsg.) (2001): Open-Source-Software: Ein Leitfaden für kleine und mittlere Unternehmen, Berlin

DiBona, Chris/Ockman, Sam/Stone, Mark (Hrsg.) (1999): OpenSources: Voices from the Open Source Revolution. O'Reilly: Sebastopol

Dyer, Jeffrey H. / Kentaro, Nobeoka (2000): Creating and Managing a High-Performance Knowledge-Sharing Network: The Toyota Case. In: Strategic Management Journal, 21, S 345-367

Duschek, Stephan/Ortmann, Günther/Sydow, Jörg (2001): Grenzmanagement in Unternehmungsnetzwerken: Theoretische Zugänge und der Fall eines strategischen Dienstleistungsnetzwerkes. In: Ortmann, G./Sydow, J. (Hrsg.) (2001): Strategie und Strukturierung. Strategisches Management von Unternehmen, Netzwerken und Konzernen. Wiesbaden: Gabler, S. 191-233

Freyermuth, Gundolf S. (2001): Offene Geheimnisse: Aus der Open Source Geschichte Lernen. In: c't Magazin für Computer und Technik, Nr. 20, S 176ff und Nr. 21, S 270ff

Garud, Raghu/Kumaraswamy, Arun (1993): Changing Competitive Dynamics in Network Industries: An Exploration of Sun Microsystems' Open Systems Strategy. In: Strategic Management Journal, 14, S 351-369

Garud, Raghu/Kumaraswamy, Arun/Langlois, Richard N. (Hrsg.) (2003): Managing in the Modular Age: Architectures, Networks, and Organizations. Blackwell Publishers Ltd.: Oxford

Gulati, Ranjay / Nohria, Nitin / Zaheer, Akbar (2000): Strategic Networks. In: Strategic Management Journal, 21, S 203-215

Katz, Daniel/Kahn, Robert L. (1966): The social psychology of organizations. New York: Wiley

Katz, M. L./Shapiro, C. (1985): Network Externalities, Competition, and Compatibility. In: American Economic Review, 75, S 424-440

Lawrence, P.R./Lorsch, J.W. (1967): Organization and environment: Managing differentiation and integration. Boston

Leiteritz, Raphael (2002): Der Kommerzielle Einsatz von Open Source Software und kommerzielle Open Source-Geschäftsmodelle: Zur Bedeutung von Open Source Software in Unternehmen und als Grundlage für Geschäftsmodelle. <http://ig.cs.tu-berlin.de/ap/rl/> (letzter Besuch: 29.11.2002)

- Lerner, Josh / Tirole, Jean (2000): The Simple Economics of Open Source. NBER Working Paper No. 7600
- Lincoln, James R. / Gerlach, Michael L. / Takahashi, Peggy (1992): Keiretsu Networks in the Japanese Economy: A Dyad Analysis of the Intercorporate Ties. In: American Sociological Review, 57, S 561-585
- Markus, M.L./Manville, B./Agres, C.E. (2000): What Makes a Virtual Organization Work? In: Sloan Management Review, Fall 2000, S 13-26
- Miller, Mark S./Drexler, Eric K. (2001): Markets and Computation: Agoric Open Systems. <http://www.agorics.com/Library/agoricpapers.html> (letzter Besuch: 20.10.2002)
- Moldoveanu, Mihnea/Bauer, Robert (2003): On the Structure and Dynamics of Organizational Complexity (Working Paper)
- Moody, Glyn (2001): Die Software-Rebellen: Die Erfolgsstory von Linus Torvalds und Linux. Verlag Moderne Industrie: Landsberg/Lech
- Morner, Michèle (2002): Das Open-Source-Software-Phänomen – organsatorisch betrachtet. In: Zeitschrift Führung + Organisation, 71/4, S 219-225
- Netcraft Web Server Survey (2002): <http://www.netcraft.com/survey/> (letzter Besuch: 1. 12. 2002)
- Nohria, Nitin/Gulati, Ranjay (1994): Firms and Their Environments. In: Smelser, NJ/Swedberg, R. (Hg.): The Handbook of Economic Sociology. Princeton University Press and Russel Sage Foundation: Princeton and New York, S 529-555
- Open Source Initiative (2002): <http://www.opensource.org> (letzter Besuch: 2.12.2002)
- Ortmann, Günther (1995): Formen der Produktion: Organisation und Rekursivität. Opladen: Westdeutscher Verlag
- Ortmann, Günther/Sydow, Jörg (Hrsg.) (2001): Strategie und Strukturation: Strategisches Management von Unternehmen, Netzwerken und Konzernen. Wiesbaden: Gabler
- Osterloh, Margit/Rota, Sandra/Kuster, Bernhard (2002): Die kommerzielle Nutzung von Open-Source-Software: Der Einfluss von sozialem Kapital. In: Zeitschrift Führung + Organisation, 71/4, S 211-217
- Raymond, Eric S. (1999): The Cathedral and the Bazaar. <http://www.tuxedo.org/esr/writings/-cathedral-bazaar> (letzter Besuch: 20. 11. 2002)

- Raymond, Eric S. (2001): How to become a Hacker. <http://www.tuxedo.org/~esr/faqs/hacker-howto.html> (letzter Besuch: 2. 12. 2002)
- Raymond, Eric S. (2001): The cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, Revised Edition. O'Reilly & Associates: Sebastopol
- Schinzel, Britta (1998): Komplexität als Ursache für Fehler in und Risiken mit Software. http://mod.iig.uni-freiburg.de/publikationen/publik_online.html (letzter Besuch: 1.12.2002)
- Schneidewind, Uwe/Landsberger, Matthias/Eggers, Hendrik (2002): Mythos Linux? Zur Übertragbarkeit der Koordinations- und Anreizmechanismen der Linux-Entwicklung auf Unternehmen, In: Zeitschrift Führung + Organisation, 71/4, S 226-232
- Schreyögg, Georg (1999): Organisation – Grundlagen moderner Organisationsgestaltung. 3. Aufl.: Wiesbaden: Gabler
- Shapiro, Carl/Varian, Hal R. (1999): Information Rules: A Strategic Guide to the Network Economy. Harvard Business School Press: Boston
- Stallman, Richard (1998): The GNU-Project. <http://www.gnu.org/gnu/the-gnu-project.html> (letzter Besuch 2.8.2003)
- Sydow, Jörg (1992): Strategische Netzwerke: Evolution und Organisation. Wiesbaden: Gabler
- Tippmann, Daniel (2001): Open Source und Zope – Eine Einführung in Freie Software. <http://userpage.fu-berlin.de/~danitipp/daniel/opensource.html> (letzter Besuch: 1. 12. 2002)
- Uzzi, Brian (1996): The Sources and Consequences of Embeddedness for the Economic Performance of Organizations: The Network Effect. In: American Sociological Review, 61, S 674-698
- Wayner, Peter (2002): Free for all. How Linux and the Free Software Movement Undercut the High-Tech Titans. HarperCollins: New York
- Weick, Karl E. (1985): Der Prozess des Organisierens, Frankfurt/Main: Suhrkamp
- Weinberg, Gerald M. (1998): The psychology of computer programming: Silver anniversary ed. Dorest House Publishing: New York