

# Freie kommerzielle Software als Geschäftsmodell

---

Der folgende Artikel beleuchtet ein neuartiges Geschäftsmodell. Es ist die konsequente Entwicklung von der agrarischen Gesellschaft über die Industrialisierung (Taylorismus, Fordismus) hin zu einer noch erwachsenden Informationsgesellschaft. Die Zukunft der Wirtschaft, vor allem im deutschsprachigen Raum liegt in diesem Geschäftsmodell, das die Vorteile der Globalisierung nutzt ohne den Nachteilen hilflos ausgeliefert zu sein. Diese Art von Geschäftsmodell findet weniger Beachtung, als es die wirtschaftliche Bedeutung rechtfertigen würde. Der größere Teil des Umsatzes wird in Deutschland von mittelständischen Betrieben erwirtschaftet. Das Open-Source-Geschäftsmodell ist auch noch etwas – sagen wir mal "unspektakulär" – und obendrein auch noch ganz und gar transparent.

---

Die folgenden Ausführungen stellen das Open-Source-Geschäftsmodell vor und zeigen am Portfolio des Unternehmens WhereGroup, wie es in Geschäftsbereiche umgesetzt werden kann. Die WhereGroup ist ein typischer mittelständiger Dienstleister mit Hauptsitz in Bonn und 25 Beschäftigten. Der Umsatz wird zum größten Teil in Deutschland und den angrenzenden Staaten erwirtschaftet. Die von der WhereGroup eingesetzten GIS-Produkte, Datenbanken, Server und Anwendungen werden international von über 180 Entwicklern aus 30 Unternehmen entwickelt und weltweit eingesetzt. Da sämtliche Software als Open Source lizenziert ist, gibt es viel gesunden Wettbewerb. Die WhereGroup tritt in diesem Software-Weltmarkt ebenfalls als Entwickler einer Software auf. Das Projekt Mapbender (<http://www.mapbender.org/>) wird maßgeblich von dem Bonner Unternehmen entwickelt. Die Open Source

Geospatial Foundation (<http://www.osgeo.org/>) ist eine Non-Profit-Organisation, die über Qualität und rechtliche Sicherheit wacht und die technische Infrastruktur für die professionelle Entwicklung von Enterprise-Software bereitstellt.

## Freie und Open-Source-Software – Herkunft und Ursprung

Der Begriff Software ist noch recht jung. Der Begriff Hardware ist zwar auch erst im Zusammenhang mit Software geprägt worden, die damit verbundenen Konzepte haben aber eine wesentlich längere Tradition. Der Schmied nutzt mit einem Hammer im materiellen Sinn ebenfalls eine Hardware. Die entsprechende Software ist lediglich die Anleitung wie der Hammer geschwungen werden muss, um das gewünschte Ergebnis zu erzielen. Ob das Ergebnis gelungen ist, hängt größtenteils davon ab, wie geübt der Schmied in der

Anwendung des Hammers ist. Hier kann man eine Parallele zum Begriff der Softwareschmiede ziehen. Ein Teil der Firma WhereGroup ist eine Softwareschmiede im ureigensten Sinne, allerdings kommt dabei nicht ein Hufeisen, Pflug oder Kolben heraus, sondern etwas ganz Neues, was

sich die Welt vor hundert Jahren noch nicht einmal vorstellen konnte: Software.

Zunächst erscheint Software nicht so speziell, sie hat aber Eigenschaften, die sich ganz und gar von denen der Hardware unterscheiden. Die folgende Tabelle gibt eine kleine Übersicht:

Hardware	Software
Wenn der Hersteller einer Hardware (jeder materiellen Ware) diese Ware weggibt, erleidet er logischerweise den Verlust dieser Ware. Üblicherweise wird dieser Verlust ausgeglichen, z.B. durch Zahlung von Geld.	Wenn die Kopie eines Software-"Produktes" weggegeben wird, entsteht kein Verlust, da die Kopiervorlage nicht verloren geht. Es sind nur exakte Kopien.
Wenn Hardware kaputtgeht, wird sie unbrauchbar und verliert ihren Wert.	Software kann nicht kaputtgehen. Ein Datenträger kann zerstört werden, aber die Kopiervorlage ist davon nicht betroffen.
Hardware kann nicht dupliziert werden. Jede "Kopie" einer Hardware benötigt genauso viele Rohstoffe wie die Vorlage. Kopien von komplexer Hardware können auf atomarer Ebene niemals eine exakte Kopie sein, sondern immer nur eine Annäherung.	Software kann dupliziert werden. Jede (technisch korrekte) Kopie eines Software-Produktes ist eine identische Kopie der Vorlage. Das "Rohmaterial", der Quelltext ist unbegrenzt haltbar.
Hardware nutzt sich ab, rostet, zerbricht und wird aufhören zu funktionieren.	Software nutzt sich nicht ab, rostet nicht, zerbricht nicht und hört nie auf zu funktionieren. Software wird vielleicht nicht mehr eingesetzt, aber sie kann nicht im materiellen Sinn kaputtgehen.

**Tabelle 1: Unterschiede von Hardware und Software**

Freie Software, Lizenzmodell und Open-Source-Entwicklungsmethodologien berücksichtigen die speziellen Eigenschaften von Software. Proprietäre Geschäftsmodelle versuchen hingegen, diese Eigenheiten von Software auszuhebeln, z.B. indem die Software "personalisiert" wird, also an eine bestimmte Hardware gebunden wird. Ändert sich die Hardware, funktioniert auch die Software nicht mehr (siehe auch Windows-Produktaktivierung). Das ist an sich nichts Verwerfliches (auch wenn der Autor manchmal diesen Eindruck erwecken sollte), aber

es ignoriert das unglaubliche Potenzial, das in Software steckt, weil sie eben ganz andere Eigenschaften hat als unsere gewohnte materielle Welt.

### **Die Ausbildung der Open-Source-Praxis im 21. Jahrhundert**

Basis dieses Abschnittes ist der ausgezeichnete Artikel von Gundolf S. Freyer-muth „Offene Geheimnisse. Die Ausbildung der Open-Source-Praxis im 20. Jahrhundert“ (Open Source Jahrbuch 2007). Freyer-muth erläutert die Entstehung und Entwicklung des Open-Source-Gedan-

kens in drei Phasen und beschreibt sechs Innovationen, die zur heutigen Praxis führen.

- Erste Phase: Historischer Ursprung der Open-Source-Praxis
- Zweite Phase: Praktische Begründung der Open-Source-Praxis
- Dritte Phase: Soziale Durchsetzung der Open-Source-Praxis

Die sechs grundlegenden Bestrebungen, die sich in der Open-Source-Praxis Ausdruck verschafften, sind:

Die erste Innovation: Der Wille der Nutzer, Souveränität im Umgang mit digitaler Technologie zu erlangen, wie er sich zuerst in den Hacker- und Time-Sharing-Bewegungen der sechziger Jahre zeigte - der Wille der Nutzer.

Die zweite Innovation: Das Streben der Nutzer nach Kompatibilität und Standards, das sich in der Frühzeit digitaler Vernetzung offenbarte - Evolution der Standards.

Die dritte Innovation: Der Wunsch nach egalitärer Kooperation, wie sie erst entortete dialogische Echtzeit-Kommunikation in Kombination mit dem Prinzip des Peer-Review ermöglicht - Produktionsgemeinschaft der Gleichen.

Die vierte Innovation: Das Verlangen nach geistigem Gemeineigentum, insofern es Informationen über Produkte betrifft, die in Arbeit und Alltag genutzt werden - Geistiges Gemeineigentum.

Die fünfte Innovation: Das Streben nach Selbstorganisation - Vernetztes Wissensarbeiten.

Die sechste Innovation: Die Verschränkung von Konkurrenz und Kollaboration im Interesse der Beschleunigung und Verbesserung kreativer Arbeits- und Entwicklungsprozesse - Konkurrierende Kollaboration.

## Das Open-Source-Entwicklungsmodell

Das Open-Source-Entwicklungsmodell beruht auf der Erkenntnis, dass die Herstellung von Software in enger Abstimmung mit allen Beteiligten (Akteuren) erfolgen sollte. Die künstliche Trennung, die eine Erstellung im Geheimen und Nutzung nur mit eingeschränkten Rechten mit sich bringt, ist für die Entwicklung der Software immer von Nachteil. Je offener und vor allem transparenter der Entwicklungsprozess einer Software ist, um so qualitativ hochwertiger wird sie. Zu den Vorteilen, die ein offenes, kollaboratives Entwicklungsmodell haben kann, gibt es viel Literatur, weshalb in diesem Artikel nicht näher darauf eingegangen wird (Eric S. Raymond, *The Cathedral and the Bazaar*; (<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>)).

Das gilt auch und im Besonderen für sicherheitsrelevante Aspekte. Sicherheit durch Geheimhaltung ist keine Sicherheit, sondern ein riskantes Spiel, da niemals transparent ist, wo eine Sicherheitslücke auftritt oder wann sie geschlossen wird. Hier ist auch wichtig festzuhalten, dass die Nutzung von Open-Source-Software nicht bedeutet, dass Inhalte und Daten ebenfalls "offen" oder "frei" verfügbar sind. Es ist sogar andersherum, die Mehrheit der großen Sicherheitsarchitekturen weltweit laufen bereits mit Open Source.

## Das Freie-Software-Lizenzmodell

Jedes Werk kann unter ein Copyright gestellt werden. Das Lizenzmodell definiert, welche Vertriebsbedingungen und Nutzungsmöglichkeiten der Lizenznehmer erhält. Im proprietären Gebrauch muss der Lizenznehmer für die Nutzung der Software ein Entgelt entrichten. Das ist die

Einnahmequelle aus der die Softwareentwicklung finanziert wird. Die Vertriebsbedingungen zeichnen sich allgemein durch Beschränkungen und Verbote aus (die Software darf nicht kopiert, verändert oder weitergegeben werden).

Freie-Software-Lizenzen drehen dieses Konzept um und räumen den Lizenznehmern explizite, weitgehende Rechte ein. Der Benutzer darf die Software nicht nur nutzen, sondern auch ändern, anpassen, kopieren und weitergeben, mit und ohne Veränderungen. Manche Lizenzen wie die GNU GPL fixieren zudem die rechtlich gesicherte freie Verfügbarkeit durch das sogenannte Copyleft. Es ist rechtlich nicht möglich, eine durch die GNU-GPL-Lizenz geschützte Software mit einem alleinigen Verwertungsrecht zu versehen. Die Software ist dauerhaft gegen die Übernahme durch ein Unternehmen mit proprietären Wirtschaftsinteressen, also den Verkauf von Lizenznutzungsrechten geschützt. Die Software wird dabei zunächst unter das Copyright gestellt und damit rechtlich schützenswert. Dann werden Vertriebsbedingungen hinzugefügt (z.B. die GNU-GPL-Lizenz), die die öffentliche Verfügbarmachung der Software vorschreiben. Damit sind Software und Lizenz untrennbar miteinander verbunden.

### **Open-Source-Geschäftsmodelle**

Vor allem der starke Copyright-Schutz mutet zunächst geschäftsfeindlich an, ist doch hinlänglich bekannt, dass durch den Verkauf von Software-Lizenzen viel Geld verdient werden kann. Der Anteil von Einnahmen durch Lizenznutzungsrechte ist gesehen am Gesamtvolumen des Marktes allerdings wesentlich geringer als der Dienstleistungsanteil. Die wenigen aber großen proprietären Unternehmen

finden oft weltweite Aufmerksamkeit, da sie durch wirtschaftliche Entscheidungen ganze Märkte verändern können. Deshalb kennt man diese Firmen auch mit Namen. Da sie einen großen Teil ihrer Einnahmen aus dem Verkauf von Nutzungslizenzen generieren, sind sie auch stark davon abhängig, dass der Bekanntheitsgrad steigt und der Verkauf weitergeht.

Wie kann ein Unternehmen aber Geld verdienen, wenn es die Software verschenkt?

Für Dienstleister ist das Freie-Software-Lizenzmodell sehr praxisnah und entspricht eher den Geschäftsfeldern des Mittelstands. Hier bietet die Open-Source-Praxis viele Vorteile, keine Nachteile und gewährleistet gleichzeitig mehr Unabhängigkeit. Das Prinzip ist sehr einfach. Immer wenn ein Unternehmen Dienstleistung für einen Kunden erbringt, fallen Kosten an, denn es gibt keine kostenfreie Dienstleistung. Die Ressource, die hier verbraucht wird, ist Arbeitszeit. Jede Dienstleistung wie Beratung, Installation, Schulung und Support kostet Geld. Die Nutzung der Software ist frei. Wenn es jedoch erforderlich ist, dass ein Programmierer spezielle Anpassungen für den Kunden vornimmt, kostet das Geld. Genauso muss die Durchführung einer Schulung bezahlt werden, so wie das bei proprietärer Software auch der Fall ist.

Open-Source-Geschäftsmodelle sind also nicht geheimnisvoll. Sie sind nur etwas ungewohnt, da wir üblicherweise die Schlagzeilen der wenigen großen Softwareunternehmen vor Augen haben, die Milliardengewinne einfahren. Zusammen mit der Annahme, Software sei nichts anderes als eine Ware im herkömmlichen Sinn, fällt es schwer zu ermessen, welche anderen Einnahmequellen es noch

gibt. Im Gegensatz zu anderen Dingen wird Software durch Gebrauch nicht abgenutzt. Es entsteht auch kein Verlust, wenn die Software als Kopie zu einem neuen Besitzer gelangt, weil die Kopiervorlage dadurch nicht beeinträchtigt wird. Software vermehrt sich (so wie jedes Wissen) erst durch seine Weitergabe. Diese einfache Erkenntnis ist die Basis für eine ganz neue Wirtschaftsform, die der materiellen Mangelwirtschaft gegenübersteht. Die Ressource Wissen ist immer im Überfluss, unbegrenzt und wird stetig mehr.

### Vorteile der neuen Wirtschaftsform für den Nutzer

Es gibt eine Vielzahl von Vorteilen der Open-Source-Praxis. Die Vorteile für den Anwender und Nutzer sind hinlänglich dargelegt worden. Dazu zählen:

**Kostenvorteil:** Der Einstiegspreis ist niedriger, da keine initialen Lizenzkosten anfallen.

**Wahlfreiheit:** Der Nutzer kann beliebig viele verschiedene Lösungen in vollem Umfang testen, hat keine Beschränkungen, was die Dauer des Tests angeht oder den Umfang der Funktionalität.

**Investitionssicherheit:** Der Nutzer kann sicher sein, dass die Investition in das Know-How dieser Software auch ohne langfristige finanzielle Verpflichtungen durch Wartungsverträge genutzt werden kann.

**Unabhängigkeit:** Es gibt keine Abhängigkeit von einem einzelnen Hersteller. Dienstleister sind ebenfalls unabhängig von einzelnen Herstellern und können neutral beraten.

**Selbstständigkeit:** Investitionen können zielgerichtet die Software an den gewünschten Stellen verbessern.

### Vorteile der neuen Wirtschaftsform für den Dienstleister

Die Vorteile für die Wirtschaft werden seltener beschrieben und sind je nach Sektor zudem sehr unterschiedlich. Für die WhereGroup sind folgende Punkte von besonderer Bedeutung:

**Wahlfreiheit:** Die WhereGroup kann aus einer Vielzahl von verschiedenen Programmen die am besten geeigneten Produkte zusammenstellen. Es können beliebig viele verschiedene Lösungen in vollem Umfang getestet werden, es gibt keine Beschränkungen, was die Dauer des Tests angeht oder den Umfang der Funktionalität. Dadurch kann die WhereGroup eine sehr breite Kompetenz entwickeln und die Mitarbeiter bleiben auf dem aktuellen Stand.

**Glaubwürdigkeit:** Schon während der Beratung kann der Kunde zwischen sehr verschiedenen Softwarepaketen auswählen. Es liegt keine Abhängigkeit zu einem bestimmten Produkt vor.

**Vielfalt:** Die WhereGroup hat sich auf Geo-Portale, Web-Anwendungsentwicklung sowie den Aufbau und die Betreuung von Geodateninfrastrukturen spezialisiert. Hierbei kann aus einem großen Pool unterschiedlicher Software-Produkte, die allesamt als Open Source lizenziert sind, eine individuelle Lösung zusammengestellt werden. Bei einigen dieser Produkte ist das Entwicklungsteam der WhereGroup auch direkt an der Entwicklung der Software beteiligt. Welche das sind, richtet sich auch nach dem Kundenprofil und der Nachfrage auf dem Markt. Open Source steht also keineswegs dem Wettbewerb im Weg. Im Gegenteil, die große Vielfalt spornt zu immer besserer Leistung an.

**Kooperationsfähigkeit:** Aus den zwei

englischen Begriffen "Cooperation" (Kooperation) und "Competition" (Wettbewerb) wurde das neue Wort "Coopetition" geschaffen. Es beschreibt die in der Open-Source-Praxis übliche Situation, dass Kooperationspartner gleichzeitig miteinander im Wettbewerb stehen. Für die WhereGroup bedeutet dies, dass sie im Bedarfsfall auf ein großes Netz an Entwicklungspartnern zurückgreifen kann, die in ihrem Spezialgebiet Koryphäen sind. Für den Kunden bedeutet es, dass kein Dienstleister ein Monopol und Marktherrschaft aufbauen kann.

**Auflösung des Kunde-Verkäufer-Antagonismus:** Der Käufer eines Softwareproduktes fühlt sich schnell übervorteilt, wenn eine zugesicherte Qualität des Produkts nicht eingehalten wird und hat weniger Motivation, das Problem selbst zu beheben, denn für die (Nutzung der) Software hat er ja bereits gezahlt. Der Verkäufer hingegen muss mit einem Produkt arbeiten, das aus dem eigenen Haus kommt, auch wenn es vielleicht Alternativen gibt, die einfach besser sind. Manche Bereiche, die eine sehr spezielle Lösung erfordern, können so nicht bedient werden. Oft empfindet der Kunde zuviel gezahlt zu haben, während der Verkäufer versuchen muss, die Einnahmen zu maximieren, weil die Ausgaben ja bereits getätigt aber noch nicht refinanziert wurden. Das direkt angewendete Dienstleistungsprinzip bei dem Geld für die Implementierung oder Anpassung eines Systems gezahlt wird, finanziert die Arbeit in dem Moment, in dem sie entsteht. Das setzt das Ergebnis der Arbeit viel besser in Wert, da es in anderen Zusammenhängen auch erzielt werden kann. Die Ausgaben sind vollkommen transparent und kontrollierbar. Außerdem besteht jederzeit die Möglichkeit, den

Anbieter zu wechseln, da der Code der Software nicht geheim ist.

## **Geschäftsbereiche der Where-Group**

Die WhereGroup hat das proprietäre Lizenzmodell nicht weiter verfolgt und auch seine eigene ehemals proprietäre Software unter eine Freie-Software-Lizenz gestellt. Damit hat das Unternehmen bewusst auf eine Einnahmequelle verzichtet, die allerdings auch in den Jahren zuvor nicht besonders ergiebig war, da Geographische Informationssysteme (GIS) nur von einem relativ überschaubaren, hoch spezialisierten Sektor benötigt werden. Anders verhält es sich mit Softwarepaketen, deren Stückzahlen hoch skalierbar sind wie z.B. Betriebssysteme, Web-Browser oder eine Textverarbeitung. Diese Komponenten finden sich auf fast jedem Endbenutzer-PC, hier geht es also um mehrere hundert Millionen Stück. Für den Verkauf proprietärer Software ist das der mit Abstand lukrativste Bereich, selbst bei sehr geringen Stückpreisen kommen in der Summe große Beträge zustande. Trotzdem finden sich auch hier für alle drei Aufgabengebiete vergleichbare Open-Source-Lösungen wie zum Beispiel das GNU/Linux-Betriebssystem Kubuntu, der Browser FireFox oder die Textverarbeitung [www.OpenOffice.org/](http://www.OpenOffice.org/). Die Stückzahlen von GIS oder gar Geodateninfrastrukturen sind dagegen sehr viel geringer. Wen außerhalb Deutschlands interessiert z.B. die ALKIS®-Datenhaltungskomponente?

Auf der anderen Seite kann inzwischen mit fast jedem PC auf Geodaten und räumliche Funktionen zugegriffen werden – unter anderem auch auf besagte ALKIS®-Daten (wenn sie denn in ein paar Jahren wirklich alle verfügbar sind). Das

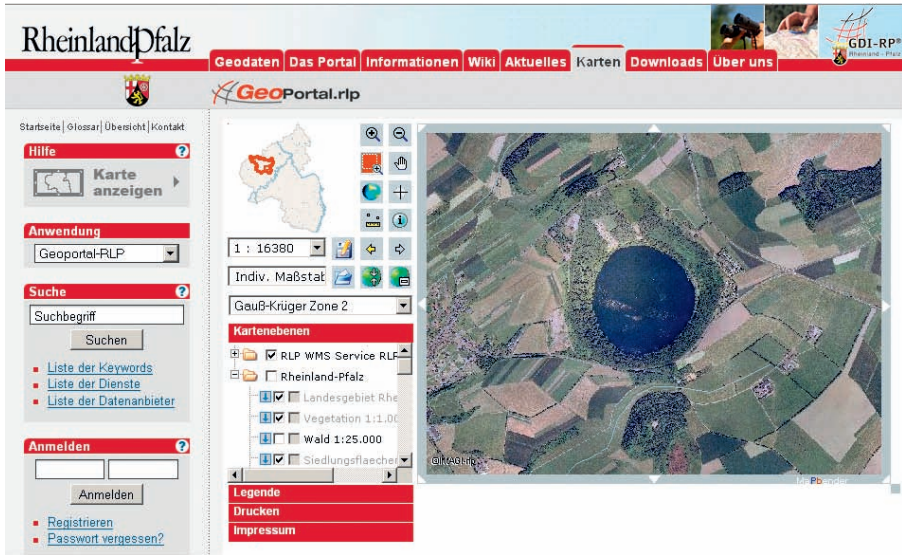


Abb: (Screenshot-Geopor:) Das Geoportal Rheinland-Pfalz (www.geoportal.rlp.de )

hat die massenhafte Verbreitung von Web-Browsern und den hohen Vernetzungsgrad der Endbenutzer-PC ermöglicht. Innerhalb von kurzer Zeit wurde es einer sehr großen Zahl von Menschen möglich gemacht, auf Karten und räumliche Informationen zugreifen zu können.

Der Aufbau von Geodateninfrastrukturen (GDI) wird durch Freie-Software-Lizenzen und Open-Source-Geschäftsmodelle begünstigt. GDI sind komplexe Gebilde, die nicht aus der Software eines Herstellers bestehen, sondern eine Vielzahl von Datenquellen bündeln, verbinden und bereitstellen. Deshalb ist jede GDI unterschiedlich und benötigt viel Dienstleistung. Anders als bei hochskalierter Endbenutzer-Software, wie einer Textverarbeitung, die einfach nur installiert wird und dann hoffentlich so funktioniert wie versprochen.

### Dienstleistung im Mittelstand

Open-Source-Software lässt sich nicht besonders gut als Produkt verkaufen,

da die Lizenzen spezifizieren, dass die Nutzungsrechte immer kostenfrei sein müssen. Lediglich die Kosten der technischen Reproduktion und Verteilung der Software können in Rechnung gestellt werden. Da sich diese aber auf wenige Cent pro Kopie beschränken, ist damit kein rentables Geschäft zu generieren. Der kommerzielle Ansatz rund um Open-Source-Produkte muss also ein anderer sein. Welches Portfolio umfasst nun ein Unternehmen, das Freie Software entwickelt und Projekte mit ebendieser Software umsetzt? Bei der WhereGroup sind dies unter anderem:

- Beratung zur Architektur- und Produktauswahl
- Installation von Basiskomponenten und speziellen Lösungen
- Anpassung an bestehende Software-Architekturen
- Wissensvermittlung und Training
- Implementierung von Geodateninfrastrukturen

- Generische Software-Entwicklung
- Bereitstellung der OSGeo-Distribution
- Wartungsverträge zu ausgewählten Produkten

Beratung und Installation sind traditionelle Geschäftsbereiche. Sie werden durch den Abschluss von Wartungsverträgen investitions gesichert, da jetzt der Haftungsausschluss, der den meisten Open-Source-Software-Lizenzen zugrunde liegt, nicht mehr gilt. Er kann durch den Vertrag mit dem Dienstleister umgangen werden, indem dieser explizit die Haftung für ein Produkt, eine Infrastruktur oder eine Entwicklung übernimmt. Durch die Bereitstellung von Distributionen, z.B. der rechtlich überprüften Software der Open Source Geospatial Foundation (OSGeo) kann sich der Dienstleister selbst wiederum gegenüber Dritten schützen, die möglicherweise über eine Patentklage Ansprüche geltend machen möchten. Zwei Bereiche sollen etwas eingehender beleuchtet werden, zum einen die Wissensvermittlung und zum anderen die generische Softwareentwicklung.

### Wissensvermittlung

Ein klar definierter Geschäftsbereich ist die Wissensvermittlung. In diesem Bereich ist es noch weitgehend unerheblich, ob sich die Schulung auf Open Source oder proprietäre Software bezieht. Möglicherweise ist es etwas einfacher, Schulungen für Open-Source-Software anzubieten, da hierfür nicht umständlich spezielle Lizenzen erworben werden müssen. Die unterschiedlichen Kategorien von Wissensvermittlung sind:

- Schulungen und Training für bekannte Softwarepakete
- Inhaltlich ausgerichtete Workshops bei

- denenspezielle Themenschwerpunkte mit Softwarepaketen zum Einsatz kommen
- Hochschulausbildung und Weiterbildungsmaßnahmen (eher softwareunabhängig)

Manchmal werden Kosten für den Erwerb von Schulungsmaterial berechnet. Vor allem wegen der arbeitsintensiven Erstellung der Unterlagen ist das auch sinnvoll. Entsprechend der Softwarelizenzen gibt es auch open-source-artige Lizenzen für Dokumente, Tutorials und Präsentationen. Die Enzyklopädie Wikipedia ist ein Beispiel für die Lizenzierung eines Gesamtwerkes, sie wird durch die Lizenz GNU FDL (Free Documentation License, <http://www.gnu.org/licenses/licenses.html#FDL>) geschützt. Eine andere Alternative ist die Creative Commons-Lizenzfamilie (<http://creativecommons.org/>).

### Kosten in Relation gesetzt

An einem Beispiel sollen die Probleme aufgezeigt werden, die der Vermittlung des Open-Source-Gedankens und -Geschäftsmodells zugrunde liegen. Nehmen wir als Beispiel eine relationale Datenbank, in der Flurstückgrenzen und Vermessungspunkte hinterlegt werden sollen. Als Alternativen bieten sich das proprietäre Produkt Oracle Spatial und die Open-Source-Datenbank PostgreSQL/PostGIS an.

Für einen Test kann die kostenlose Oracle-XE-Lizenz genutzt werden. PostgreSQL/PostGIS kann als Open-Source-Software sowieso kostenfrei genutzt werden. (Das Unternehmen Oracle behält sich rechtliche Schritte vor, wenn Leistungsvergleiche mit anderen Datenbanken veröffentlicht werden, die zuvor nicht durch Oracle genehmigt wurden. Der Autor geht allerdings davon aus, dass die bloße

Nennung einer Alternative noch nicht zu rechtlichen Schritten führen wird).

Da es sich um mehrere Millionen Objekte handelt, muss für den Betrieb unter Oracle eine kostenpflichtige Lizenz erworben werden. PostgreSQL bleibt kostenfrei. Die Daten sollen zudem über das Internet bereitgestellt werden, dafür ist eine Internet-Lizenz erforderlich, die mit weiteren Kosten verbunden ist. PostgreSQL bleibt kostenfrei. Da mit hohen Zugriffszahlen zu rechnen ist, wird eine Mehrprozessormaschine benötigt, die weitere Lizenzkosten nach sich zieht. PostgreSQL bleibt kostenfrei.

Um die Performance der Datenbanken zu optimieren, wird ein Angebot für eine Schulung und für Dienstleistung durch einen Datenbank-Administrator angefordert. Erst jetzt fallen auch beim Betrieb von PostgreSQL Kosten an, denn jetzt soll tatsächlich Arbeitszeit verbraucht werden.

Bei der Lösung mit der Oracle-Datenbank sind inzwischen allerdings bereits sechsstellige Investitionen vorgenommen worden, die Lösung ist im Haushalt eingestellt und budgetiert. Die zusätzlichen Kosten von wenigen zehntausend Euro fallen dabei gar nicht mehr ins Gewicht. Anders sieht es bei der Open-Source-Lösung auf Basis von PostgreSQL und PostGIS aus. Hier erscheinen die Preise für die Schulung und den Dienstleistungstag überhöht, weil doch die ganze Software gar nichts kostet. Und genau dieser Satz stimmt nicht. Natürlich kostet die Software Geld, und zwar richtig viel. Über eine ganz einfache Multiplikation der Zeilen des Quelltextes kommt für PostgreSQL ein Investitionsvolumen von gut US\$ 8 Millionen zustande (<http://www.ohloh.net/projects/27?p=PostgreSQL>). Wohl-

gemerkt sind das nur die Zeiten für das Programmieren, nicht für Fehlerbehebung, Pflege, Wartung des Quellbaumes, Design, Optimierung und alles, was sonst noch zur professionellen Softwareentwicklung gehört.

## Generische Softwareentwicklung

Generische (also allgemeingültig einsetzbare) Softwareentwicklung zu finanzieren, ist eines der schwierigsten Unterfangen im Bereich Open Source. Das hier vorgestellte Geschäftsmodell beschreibt eine direkte Relation zwischen geleisteter Arbeit und anfallenden Kosten – das Dienstleistungs- oder auch Handwerkerprinzip.

Generische Softwareentwicklung dauert erfahrungsgemäß 10-mal so lang (und ist damit 10-mal so teuer) wie eine schnelle Lösung für ein konkretes Problem. Allerdings kann die generische Lösung dann auch für beliebig viele konkrete Probleme eingesetzt werden. Der Nutzer einer Software sieht jedoch oft zunächst nur den kurzfristigen Vorteil einer produktiven Lösung und hat nicht den technischen Weitblick, den ein Software-Entwicklungsteam haben muss, wenn es eine nachhaltige Lösung entwickelt. Die Entwicklung einer Software erfordert aber zusätzliche Arbeiten, wie die Erstellung eines Installationspaketes, die Bereinigung von Fehlern, die Planung neuer Funktionen und so weiter, die nicht direkt finanziert werden. Des Weiteren muss eine Softwareumgebung gepflegt werden, ein Quelltextspeicher, Ticketsystem, Mailing-Listen und anderes. In der proprietären Entwicklung werden diese Investitionen über den späteren Verkauf von Lizenznutzungsrechten refinanziert. In der Open-Source-Welt war das bisher nicht möglich. Aus diesem Grund wurde die Open Source Geospatial

Foundation (OSGeo) gegründet, die einen Teil dieser Infrastruktur bereitstellt. Auch die Finanzierung der anfallenden Kosten für Betriebsarbeiten könnte in Zukunft hierüber finanziert werden, ein spezielles Projekt-Sponsoring-Programm ([http://www.osgeo.org/sponsor\\_werden](http://www.osgeo.org/sponsor_werden)) der OSGeo ermöglicht die gezielte Finanzierung einzelner Projekte oder der gesamten Organisation.

### **Zusammenfassung und Ausblick**

Freie Software und Open Source beschreiben Lizenzmodelle, die grundlegende Freiheiten in Bezug auf die Nutzung der Software garantieren statt sie zu verbieten. Damit stehen sie im Widerspruch zu proprietären Geschäftsmodellen, die den Zugang zu ihrer Software beschränken, um durch Einräumen von begrenzten Rechten Einnahmen zu generieren.

Software unterscheidet sich von herkömmlichen materiellen Produkten durch die fehlende Körperlichkeit. Diese ermöglicht in Zusammenhang mit praktisch kostenfreier Multiplizierbarkeit ganz neue Ansätze im Wirtschaften und ermöglicht eine Vielzahl von Kooperationsformen.

Open Source ist die Basis für klare, wirtschaftlich erprobte Geschäftsbereiche. Lediglich der Verkauf von begrenzten Lizenznutzungsrechten fällt weg. Für den Mittelstand und besonders für Nischensoftware sind die Einnahmequellen aus diesen Bereichen oft gering, der Verzicht darauf wird durch die Vorteile mehr als ausgeglichen. Aus der Offenheit der Quelltexte dieser Software und der fehlenden Einschränkung der Nutzungsrechte ergeben sich eine Vielzahl von Vorteilen nicht nur für den Benutzer, sondern auch den Hersteller und Dienstleister dieser Software.

Der Umstieg von einem System auf

das andere ist auf jeden Fall mit Kosten verbunden, nichts ist umsonst. Dies wird aber durch höhere Transparenz bei den Kosten, mehr Investitionssicherheit und Wirtschaftlichkeit kompensiert. Speziell benötigte Funktionalität kann in Open Source basierter Software zielgerichtet und meist viel schneller umgesetzt werden.

Die aktuelle Entwicklung lässt vermuten, dass sich Open-Source-Lizenzen und die damit verbundenen Geschäftsmodelle immer weiter durchsetzen werden. Es besteht jedoch keine Eile. Wer ein proprietäres System aufgebaut hat und damit alle Anforderungen erfüllt, muss nicht sofort auf eine Open-Source-Lösung umsteigen. Wenn es allerdings um Erweiterungen geht, können auch bestehende Infrastrukturen von den Vorteilen der Open-Source-Praxis profitieren, egal ob sie auf Basis proprietärer oder Open-Source-Systeme aufgebaut wurden.

