

Opensource aus der Sicht eines Entrepreneurs	2
Mein Umfeld	2
Annahmen:	3
Wie wird mit OS Software Geld verdient	3
Oekonomie von Open Source	3
Merkmale:	3
Randbedingungen	4
Lebenszyklen von Software / wer nutzt OSS	4
Der Preis von Software	5
Das Pareto Prinzip oder die 20/80 % Regel	6
Lebenszyklen von Software revisted	6
Merkmale:	7
OSS als Werkzeug	7
Merkmale:	8
Wie gut ist OSS geeignet, den gezeigten Randbedingungen zu entsprechen	9
Bedienen der Early Adaptors	9
Bedienen der Early/Late Majorities	9
Nochmals die 20/80% Regel	9
Aussicht / Marktchancen	10
Zusammenfassung	10
Kontakt:	10

OpenSource aus der Sicht eines Entrepreneurs

Ziel des Vortrages:

- Zeigen was verschenken von Software attraktiv macht.
- Positive Elemente zeigen die OSS in den Business Plan einbringt
- Schwächen und deren Konsequenzen von OSS aufzeigen
- Markt-Bereiche andeuten, die noch brach liegen

Was der Vortrag nicht will:

Eine Lanze für Open Source Software brechen!

Ich betrachte OSS als Fact of Life, dessen immenser Erfolg sicher ist.

Mein Umfeld

Ich bin als Sohn eines veritablem Daniel Düsentricks aufgewachsen. Als Knabe stopfte ich meine nassen Schuhe mit alten Konstruktionsplänen aus. Diese Affinität zu technischen Zeichnungen schlug sich im meinem frühen Erwachsenenalter in Begeisterung für CAD-Programme nieder. Während meines Studiums arbeitete ich eng mit einem der Gründer von Autodesk zusammen und wurde einer der ersten AutoCAD-Händler Europas.

Damit begann eine lange und fruchtbare Zusammenarbeit mit einem Unternehmen, dass es wie kein zweites verstand, die eigenen Schwächen durch Offenheit zu kompensieren. Niemand von den Gründervätern der erfolgreichsten CAD-Firma der Welt wussten was CAD ist. Ihr Rezept war, Schnittstellen offenzulegen und Werkzeuge zu schaffen, damit diejenigen, die mehr von CAD verstanden, die fehlenden Teile zum Ganzen fügen konnten. Die Folge war, dass in kurzer Zeit eine riesige „virtual“ Company, entstand, die AutoCAD mit jeder nur erdenklichen Zusatzfunktionalität versah und es damit in die hinterste und letzte Marktnische drückte. Autodesk hatte damit, 10 Jahre bevor Open Source „erfunden“ wurde, ein Geschäftsmodell geschaffen, das diesem in vielem ähnlich ist.

Zu dieser virtual Company trug meine Firma reale 20 Personen bei, die während knapp 15 Jahren technische Software entwickelte und vertrieb. Dabei wandelten wir von einem Systemhaus, das die nach heutigen Masstäben unglaublich hohen Hardwarepreise mit einer Dreingabe eigener Software versüsste, zu einem reinen Softwarehaus, das seine Produkte hauptsächlich über den Wiederverkauf absetzte. 1997 verkaufte ich die Lizenzrechte an meinem damaligen Hauptprodukt. Dieses wird nach wie vor erfolgreich verkauft und bis vor kurzem wurde dessen Entwicklung wesentlich von meinen Mitarbeitern mitgetragen.

Im Jahre 2000 gründete ich mit einigen Mitstreitern die redCOR AG. Deren Ziel ist es, die Effizienz von Prozessen, die wir früher in einer Desktop Umgebung unterstützten, per Internet zu verbessern

Wir lernten dabei den Open Source Application Server Zope kennen. Ein Grossteil der Projekte die wir heute betreuen, basieren darauf, bzw auf dem CMS System Plone, das Zope als Grundlage nutzt. RedCOR beschäftigt 6 Personen.

Annahmen:

Ich gehe in meinen Erläuterungen von den folgenden Annahmen aus:

- Das oberste Ziel jeder Softwareentwicklung und jedes Softwareeinsatzes ist wirtschaftlicher Gewinn. Andere, genauso vitale Gründe, interessieren uns hier nicht.
- Meine Überlegungen gehen davon aus, dass sich ein Unternehmen im Umfeld OSS als Dienstleister etablieren, nicht ein Produkt als Open-Source freigeben oder finanzieren möchte.
- Open Source schafft keinen Markt. Es kann ihn nur bedienen. Einen Markt für ein Unternehmen zu finden ist völlig unabhängig davon, ob dieser mit Closed - oder Open-Source Produkten bearbeitet werden soll.

Wie wird mit OS Software Geld verdient

Oekonomie von Open Source

Die Ökonomie von Opensource basiert auf dem Umstand, dass jedes Produkt in einem Markt andere Produkte hat, mit denen es im Tandem gebraucht wird. Ein Auto braucht Benzin, ein Portable Batterien. Solche Produkte werden komplementär zueinander genannt. Senkt man die Preise der Komplemente, so hat das eine Zunahme der Nachfrage für das Produkt zur Folge. Ein Autohändler hat z.B. ein vitales Interesse, den Preis des Benzins zu kontrollieren.

Der selbe Mechanismus gilt für das Tandem Software-Produkt/Dienstleistung. Je günstiger der Gesamtpreis wird, je höher ist die Nachfrage danach. Woher immer ein Produkt stammt, aus der eigenen Softwareschmiede, aus Closed- oder Opensource, wir schnüren für unsere Kunden ein Angebot bestehend aus Produkt und Dienstleistungen. Dabei haben wir für das Produkt bestenfalls eine relativ kleine Verkaufsmarge. Der Löwenanteil der Einnahmen kommt von den erbrachten Dienstleistungen.

Ein Beispiel:

In einem Interview schätzte Paul Everitt, einer der Gründervater des CMS-Systems Zope, die Anteile Lizenzen/Dienstleistungen/Hardware eines Projektes im Umfeld von CMS auf je ein Drittel. Zudem nimmt er an, dass der Dienstleistungsaufwand bei Closed-Source in einem ähnlichen Rahmen liegt wie bei Open Source. Das bedeutet, dass ein Zope-Projekt zu einem Drittel günstiger angeboten werden kann.

Es liegt auf der Hand, dass mit einem derart reduzierten Gesamtpreis Projekte realisiert werden können, die am Kostenrahmen gescheitert wären.

Merkmale:

- OSS erhöht die Nachfrage nach unseren Dienstleistungen, da das Tandem Produkt/Dienstleistung billiger wird.

Randbedingungen

Wenn etwas 33% billiger ist als ein gleichwertiges Produkt, so sollte man annehmen, dass sich der Anbieter kaum der Kunden erwehren kann.

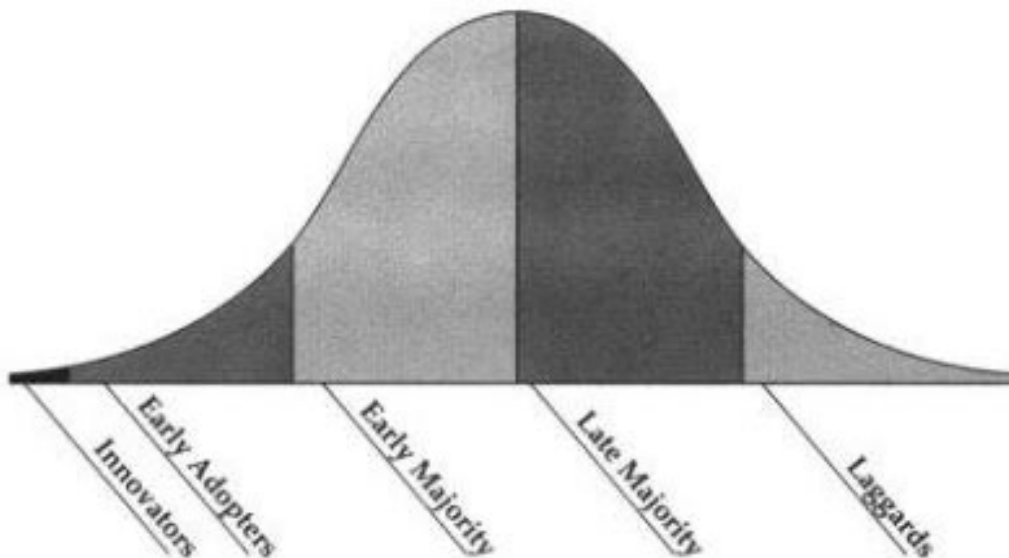
Dem ist kaum je so. Den Grund dazu wollen wir zu verstehen versuchen, in dem wir einige Randbedingungen näher betrachten, die den Markt kontrollieren.

Lebenszyklen von Software / wer nutzt OSS

Während den verschiedenen Lebens und Reifephasen einer Software wird diese von unterschiedlichen Gruppen genutzt, die unterschiedliche Ziele verfolgen, und unterschiedliche Ansprüche an Funktion und Qualität des Produktes stellen

Nur wenn es uns gelingt unser Angebot diesen verschiedenen Gruppen anzupassen, werden wir in der Lage zu sein, sie zu unseren Kunden zu gewinnen.

Die folgende Grafik stammt aus dem sehr lesenswerten Buch „Crossing the Chasm“ von G.A.Moore.



Wir sehen hier das Marktvolumen, das mit einer Software erzielt wird, über deren Lebensdauer aufgezeichnet. Das Volumen bildet eine klassische Glockenkurve, welche wir hier nach den Käufertypen, mit denen es erzielt wird, segmentiert sehen.

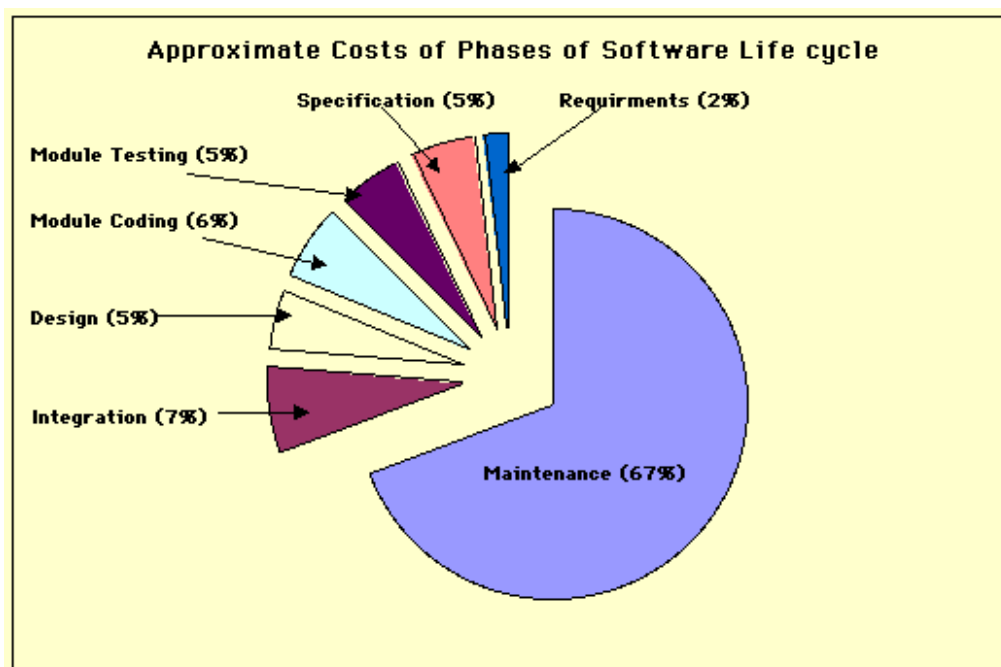
Die ersten drei Gruppen schauen wir etwas näher an.

- Inovators: Das sind Personen, für die Technologie der Technologie willen wichtig ist. Diese Gruppe generiert zwar nur einen kleinen Markt, ist aber für „proof of Concept“ und Marktöffnung unabdingbar

- **Early Adopters:** Das sind ebenfalls Personen mit Freude an der Technik. Sie sind in der Lage, den Vorteil einer neuen Technologie für ihr Fachgebiet sofort zu sehen. Sie setzen neue Technologien aber nicht deretwillen ein, sondern um einen Wettbewerbsvorteil zu erringen. Sie basieren ihren Investitions-Entscheid auf den Glauben, mit dem neuen Werkzeug ein Ziel erreichen zu können, nicht auf eine lange Referenzliste. Sie wissen, dass gewisse Ungereimtheiten und Schwächen zu jeder neuen Technologie gehören und können damit leben. Sie erwarten aber auch, dass auf ihre speziellen Wünsche eingegangen wird.
Mit dieser Gruppe werden die ersten „grossen“ Projekte durchgeführt und so der Markt geöffnet.
- **Early Majority:** Dieses Marktsegment ist von Akteuren besetzt, die ein gewisses Verständnis für neue Technologien mitbringen. Sie wissen aber, dass viele dieser Erfindungen vorübergehende Launen sind. Sie sind bereit, zu warten und zu schauen, wie sich das Ganze bei andern bewährt. Sie entscheiden sich zum Einsatz, wenn sich auch der Nachbar dazu entschlossen hat.
Über ein Drittel des gesamten Marktes ist hier zu finden. Dementsprechend ist dieses Segment Schlüssel zum Erfolg.

Der Preis von Software

Bevor wir den angefangenen Betrachtung weiterführen, fragen wir uns, wie sich die Kosten über die Entwicklungsphasen einer Software verteilen.



<http://www2.umassd.edu/CISW3/coursepages/pages/cis311/LectureMat/process/lifecycles/lifecycles.html>

Die Grafik zeigt auf, wo die Kosten der Software-Entwicklung anfallen. Wir sehen, dass nur ein kleiner Teil bei der Kodierung anfällt, der grosse Brocken wird vom Unterhalt geschluckt. Damit wird aber nicht die Leistungsfähigkeit, sondern „nur“ die Brauchbarkeit des Produktes verbessert.

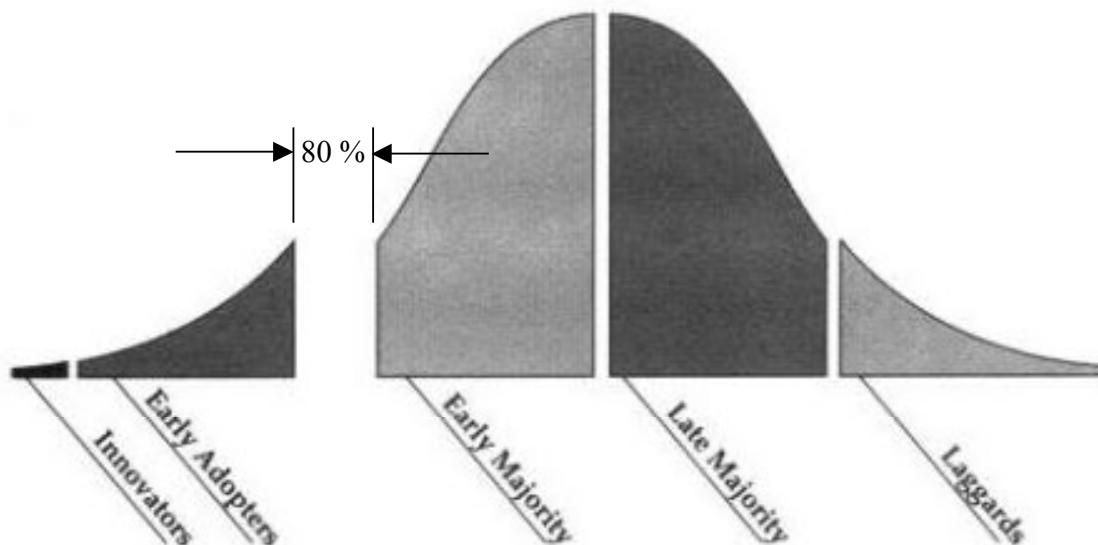
Das Pareto Prinzip oder die 20/80 % Regel

Das Pareto Prinzip besagt, dass 1/5 des Aufwandes für 4/5 der Gesamtleistung verantwortlich sind. Das Pareto Prinzip ist kein Zahlenverhältnis, nach dem sich die Natur in jedem Fall exakt richtet. Dennoch zeigt sich überraschend oft, dass der Grossteil der Gesamtleistung eines Systems von nur wenigen seiner Elemente erbracht wird, während der Rest der Leistung vom Rest der Elemente resultiert.

Anders formuliert, für 80% der Leistung muss nur 20% aufgewendet werden. Verfügen wir über ein „beinahe“ fertiges Produkt, so muss nochmals der 4-Fache Betrag eingesetzt werden, um es fertig zu stellen.

Lebenszyklen von Software revisited

Gewappnet mit diesen Einsichten werfen wir einen zweiten Blick auf die Grafik mit den Marktvolumina eines Softwareproduktes.



Wir sehen hier dieselbe Kurve, aber die einzelnen Anwendergruppen getrennt durch Lücken. Diese weisen darauf hin, wie sehr sich das Verhalten der einzelnen Gruppen unterscheidet, wenn es darum geht, neue Werkzeuge einzusetzen. Welcher Aufwand erbracht werden muss, um ein Produkt dass den Ansprüchen der einen Gruppe genügt, fit für die nächste zu machen.

Hier die wesentlichen Unterschiede zwischen den Gruppen:

- Early Adopters wollen mit einer neuen Technologie zu neuen Ufern aufbrechen um sich z.B. vom Wettbewerb abzusetzen. Sie sind bereit, dafür den eigenen Business-Prozess umzukrempeln. Was sie kaufen ist ein Werkzeug, um den Wechsel zu vollziehen.
Beispiel: Eine Entwicklertruppe stellt auf Linux um, da sie damit effektiver und kostengünstiger arbeiten können.
- Early Majority: Diese Gruppe kauft ein Werkzeug, um den bestehenden Business-Prozess zu verbessern. Sie scheuen Änderungen wie der Teufel das Weihwasser. Die Umstellung muss völlig transparent und ohne Unterbruch des gewohnten Arbeitsflusses erfolgen. Probleme oder fehlende Elemente oder Dokumentation werden nicht toleriert.
Beispiel: Eine Behörde will von Microsoft Office auf Open-Office umstellen.

Die Early Adopters können wir mit dem „beinahe“ fertigen Produkt bedienen, die Early Majority verlangt, die wir ja als Schlüssel zum Erfolg erkannt haben, dass es **wirklich** fertig ist. Dazwischen liegen 80% des Entwicklungsaufwandes.

Merkpunkte:

- Eine erste Gruppe von Anwendern kann erreicht werden, wenn erst ein geringer Teil der Entwicklungsleistung erbracht wurde. Diese verzichtet auf viele Details verlangt aber erhebliche Anpassungen an ihre individuelle Situation.
- Der grosse und lukrative Teil der Anwender investieren erst in das Produkt, wenn es fertig ist. Diese können erst mit ungleich höherem Entwicklungsaufwand erreicht werden.

OSS als Werkzeug

Eine der sichtbarsten und von den Medien am häufigsten kolportierten Eigenschaften von Opensource ist der Umstand, dass deren Produkte lizenzkostenfrei sind.

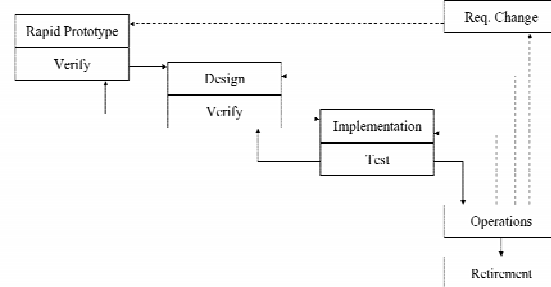
Die fehlenden Lizenzkosten sind zwar für die Medienpräsenz und damit für die Sichtbarkeit von OSS von zentraler Bedeutung, nicht aber für deren Erfolg. Dazu trägt z.B. die erdrückende Dominanz von Microsoft und die dadurch provozierten Investitionen in Opensource der IT-Riesen wie IBM und HP genauso bei.

Kein erfolgreiches Unternehmen investiert Milliarden, wenn es nicht damit rechnet, die Investitionen mit Zinsen zurück zu erhalten. Dem Konkurrenten eins auszuwischen ist dazu ungeeignet. IBM und Co sehen OSS offensichtlich als Vehikel Geld zu verdienen.

Welche Eigenschaften sind es, die versprechen, Investitionen in Opensource zu rechtfertigen ?

Ich glaube, dass Open Source Software ein Umfeld für ausserordentlich schnelles Entwickeln von Applikationen darstellt. Dabei wird ein aus dem Extreme Programming bekanntes Verfahren so erweitert, dass ein weltweit verteiltes Team involviert wird.

Rapid Prototyping



13

Das wichtigste Merkmal dieses Prozesses, ist die permanente Rückkoppelung, die bewirkt, dass Änderungswünsche sofort in den Prozess einfließen und entdeckte Probleme unmittelbar korrigiert werden können.

Dabei wird eine unglaubliche Vielfalt von unterschiedlichen Fähigkeiten und Sichtweisen eingebracht und zu einem Ganzen verschmolzen. Für jeden einzelnen Iterationsschritt kann (und wird in der Regel) die Gruppe der Beteiligten neu definiert.

Jeder Unternehmer, der sich in den OSS-Prozess einbringt, ist so in der Lage, die verfügbaren Produkte den Bedürfnissen seiner Kunden anzupassen. Dabei wird das notwendige eigene Engagement vorausgesetzt, das Produkt verändert und nicht etwa ein Derivat davon.

Jedoch: Die treibende Kraft hinter OSS ist Eigeninteresse. Mit etwas, das man verschenkt, kann man kein Geld verdienen. Jeder Beteiligte muss also daran interessiert sein, seinen Betrag weiterhin zu liefern. Es gibt keine anderen darwinschen Kräfte, die die Entwicklung oder die Richtung, in die sie sich bewegt, beeinflussen.

Die meisten Leute schätzen „schwierige“ Aufgaben nicht und fühlen sich von Ihnen gelangweilt. Unglücklicherweise sind die meisten Aufgaben, die notwendig sind um Software ausserhalb ihrer Entwicklerumgebung gebrauchen zu können solche harten Aufgaben. Dazu gehören, UI-Design, Dokumentation, Support, und Verpackung. Entwickler tendieren diese Bereiche als unwichtig links liegen zu lassen.

Merkmale:

- Open Source als Prozess erlaubt rasch Produkte zu entwickeln.
- Einige wichtige Elemente werden von diesem Prozess nicht geliefert.

Wie gut ist OSS geeignet, den gezeigten Randbedingungen zu entsprechen.

Wie meine Erfahrung mit der Firma Autodesk zeigen, ist nicht in erster Linie wichtig, dass man ein mit allen Wassern gewaschener Experte in einem Gebiet sein muss, um darin erfolgreich zu sein. Wichtiger ist, dass man mit Werkzeugen arbeiten muss, die sich sehr flexibel an die sich entwickelnden Bedürfnisse anpassen lassen.

Das OSS als Entwicklungsmodell für Applikationen entspricht dieser Anforderung in fast idealer Weise.

Ein weiterer Grund für den schnellen und langhaltigen Erfolg von Autodesk war, dass sie es verstand, eine „virtual Company“ zu schaffen, die Experten in all jenen Bereichen stellen konnte, für die im „Core-Team“ nicht genügend Wissen vorhanden war. Auch das ist eine der hervorragenden Eigenschaften eines jeden OSS Teams. Jedes aktive Team wird situationsgerecht mit Fachkräften ergänzt. Der eigene „Apparat“ bleibt schlank und kostengünstig.

Kommunikation mit Experten „draussen im Feld“ ist nicht nur Wollen sondern auch Können. Dazu muss der geeignete „Morse-Apparat“ installiert sein. Die OSS-Community bietet eine solche Apparatur.

Bedienen der Early Adaptors

Vorausgesetzt, wir haben dank unserer Marketing-Massnahmen einen Markt gefunden, ist das OSS-Entwicklungsmodell perfekt geeignet, den Ansprüchen nach Individualität der Early Adopters gerecht zu werden.

Bedienen der Early/Late Majorities

Hier sind die Vorteile von OSS gemischt:

Nochmals die 20/80% Regel

Ein grosser Teil des Aufwandes ein Programm „fertig“ zu stellen, ist, alle Fehler zu eliminieren, sodass es innerhalb aller Randbedingungen störungsfrei funktioniert. Ebenfalls dazu gehört, dass es an Randbedingungen angepasst wird, die etwas abseits vom Mainstream liegen. Die duale Funktion vieler Mitglieder des OSS-Prozesses als Entwickler/Anwender leistet dies. OSS ist also gut geeignet diesen Anspruch dieser Benutzergruppe zu erfüllen.

Hingegen ist der OSS-Prozess schlecht in der Lage diejenigen Bedürfnisse abzudecken, die diejenigen Nutzer haben, die nicht am Entwicklungsprozess teilnehmen. Dazu gehört neben der Dokumentation, die in fast allen OSS-Produkten ein ganz wunder Punkt ist, auch Installationssupport, Garantieleistungen, garantierte Interventionsfristen und so weiter.

Closed Source ist hier leistungsfähiger, denn dieser Prozess wird von den Nutzern, nicht von den Entwicklern kontrolliert. Ein Nutzer braucht (und kauft) nur das, was seinen Erwartungen entspricht. Deshalb ist Closed-Source sehr gut darin, den Kundenwünschen zu folgen.

Aussicht / Marktchancen

Wir haben gesehen, dass sich Opensource und der damit verbundene Entwicklungsprozess sehr gut eignet, die eigenen Dienstleistungen im Markt zu positionieren.

Um eine wirklich breite Anwenderbasis zu erreichen, müssen aber Leistungen geboten werden, die nicht aus dem Opensource Umfeld geschöpft werden können.

Hier bieten sich ausgezeichnete Chancen sich zu profilieren. Dabei können einzelne Unternehmen in einem schnell expandierenden Markt Dienstleistungen erbringen, die sie unverwechselbar und unverzichtbar machen.

Entschliesst sich ein Unternehmen, die Early Adopters zu bedienen, dann ist es gezwungen, in seiner Marktnische diese ergänzenden Dienstleistungen aufzubauen. Tut es das nicht, bricht ihm der Markt weg. Je Erfolgreicher ein Produkt ist, desto weniger spricht es ja die Early Adopters sondern die Majority an. Diese wird aber von anderen Anbietern bedient.

Zusammenfassung

Die Ökonomie des Opensource-Prozesses basiert darauf, dass er sein Umfeld leistungsfähiger und kostengünstiger macht.

Wer sich mit seinem Unternehmen darin engagiert, kann mit einem schlanken Apparat Leistungen erbringen, die die Möglichkeiten des klassischen „Closed-Source“ Verfahrens bei weitem übersteigen.

Der Open Source Prozess hat jedoch einige Schwächen, die verhindern, dass ein Produkt „Mainstream“ werden kann. Ein Unternehmer im OSS-Umfeld kann nur dann mittelfristig erfolgreich sein, wenn er diese Probleme löst.

Kontakt:

Robert Rottermann

RedCOR AG

www.redcor.ch

Moserstrasse 22

3014 Bern

031 333 10 20