

Open Source Software ökonomisch betrachtet

Vortrag zum Treffen der
.NET Developers Group Berlin Brandenburg
am 02.03.2006

Dieter Bremes, dbjobs@snafu.de

Übersicht

- ⇒ Open Source Software (OSS) ökonomisch betrachtet
 - Die OSS-Geschichte in 5 Minuten
 - OSS als logische Entwicklung in Software-Märkten
 - Voraussetzungen für OSS
 - OSS-Geschäftsmodelle
 - Vermeintliche OSS-Probleme

GNU/Linux, Unix: Die OSS-Geschichte in 5 Minuten

- ⇒ BS bis in 60er Beigabe zur Hardware
 - = **Kein Geschäftsmodell** für Software
- ⇒ 1969: Bell Labs (AT&T) beginnen Unix
 - AT&T darf wg. **Monopol** aber nicht vermarkten
- ⇒ Bell Labs + University of California at Berkeley (UCB) entwickeln Unix zus. weiter
- ⇒ UCB sendet Code an Interessierte + arbeitet ihre Verbesserungen ein (= **OSS**)

GNU/Linux, Unix: Die OSS-Geschichte in 5 Minuten

- ⇒ UCB -> div. BSD-Versionen
- ⇒ AT&T -> Unix System V
- ⇒ Ab 1983: Richard M. Stallman -> GNU
freies, vollständiges, Unix-kompat. System
- ⇒ Andere Anbieter -> Unix-Varianten
IRIX, XENIX, ULTIX, HP-UX, ...
- ⇒ = **Inkompatible Systeme**

GNU/Linux, Unix: Die OSS-Geschichte in 5 Minuten

- ⇒ 1988 bis 1993: Unix-Wars
 - AT&T **verklagt** UCB auf Distributionseinstellung
 - Unix auch danach dauerhaft geschwächt
- ⇒ 1991: Linus Thorvalds beginnt Linux
 - BSD in juristischen Schwierigkeiten
 - GNU fehlt der Kernel

GNU/Linux, Unix:

Die OSS-Geschichte in 5 Minuten

➔ Heute

- FreeBSD: 200 - 300 Programmierer
- Linux Kernel: 600 - 800 Programmierer
 - **Offen**ere Community
 - Vermeidet Unix-Fehler: Forks, Anbieter-Zersplitterung
- Stückzahlenwachstum Server-Markt 2005 (IDC)
 - Windows: +15,3 %
 - Linux: **+20,5 %**
 - Unix: -13,7 %

GNU/Linux, Unix: Die OSS-Geschichte in 5 Minuten

⇒ Zukunft (Gartner, IDC)

- Ab 2008: Windows **verliert** 5 % - 10 % vom Server-Markt an Linux
- "Unix wird nicht völlig verschwinden, aber es wird zunehmend zum Nischenprodukt in Datenzentren absteigen und bei KMUs völlig verschwinden"
- Linuxmarkt **wächst** jährlich um 37 % - 45 %

GNU/Linux, Unix: Die OSS-Geschichte in 5 Minuten

⇒ Zusammenfassung

- Am Anfang war OSS
- Entscheidend ist die Community (zumindest bei größeren Projekten)
- (Geschäftsmodelle für) Standardsoftware: CSS unterliegt im Wettbewerb mit OSS

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

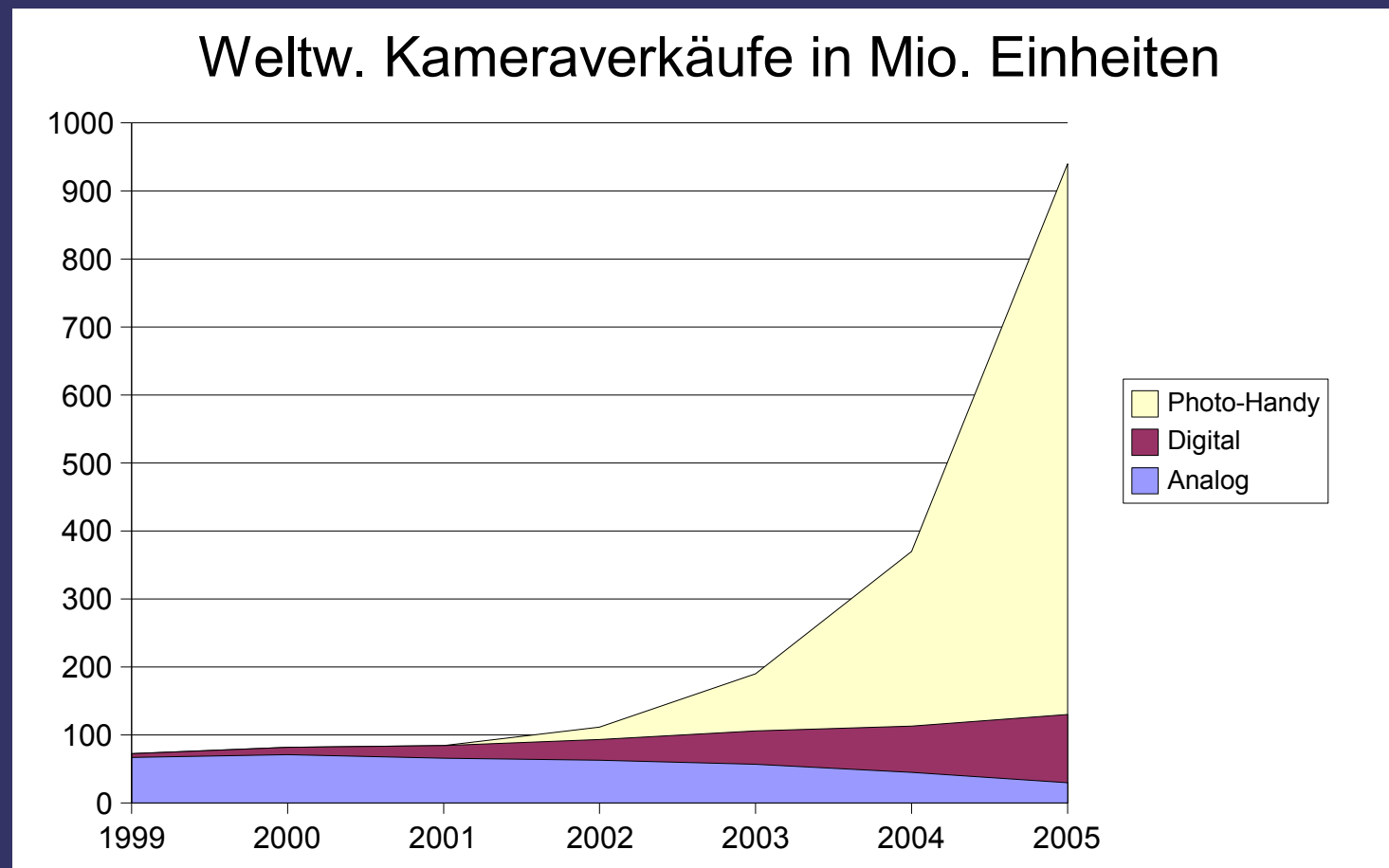
⇒ Marktphasen

- 1. Experimentierphase
- 2. Expansionsphase
- **3. Ausreifungsphase / Sättigungsphase**
- 4. Rückbildungsphase

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

- ⇒ Ausreifungsphase / Sättigungsphase
 - Nachfrage stabilisiert sich auf hohem Niveau
 - Unternehmenskonzentration steigt
 - Survival of the fattest – not the fittest
 - Sehr profitabel
- ⇒ Und dann kommt der Rivale ...
 - Röhren- -> Plasma- / LCD-TV
 - Analoge -> digitale Fotografie
 - Aber: Vergleich OS- + PC-Markt in D / China

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten



Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

⇒ Disruptive Technologien / Innovationen

- Neue Technologie völlig anders als bisherige, stört deren Verwertung
- 2 Szenarien
 - **low-end disruption**: Anfänglich schlechter, aber gut genug + billiger. Dann kontinuierliche Verbesserung bei bleibendem Kostenvorteil. Beispiel: Digitalkamera
 - new-market disruption: Bietet Funktionalität, die bisheriger Technologie fehlt + dort nicht nachrüstbar ist. Beispiel: Internet

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

⇒ Kreative Zerstörung

- Alte Struktur / Wert etablierter Unternehmen zerstört, um Produktionsmittel und Nachfragepotential für Neues zu erhalten
- Kurzfristig oft schmerzhaft (Arbeitslosigkeit), langfristiger Nutzen durch höheren Wohlstand. Beispiele siehe Disruptive Technologien

⇒ Kondratjew-Zyklus

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

- ⇒ OSS ist Basis disruptiver Geschäftsmodelle
- ⇒ Konkurrieren mit Lizenzgeschäft
 - Lizenzgeschäft nur 30 % des Software-Geschäfts
 - Hersteller muss Entwicklung, Vertrieb, etc. vorfinanzieren -> Risiko + Kosten
 - Nur ca. 10 % der Lizenzkosten fließen in Produktentwicklung (Perens, MS) -> ineffizient

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

- ⇒ Märkte für Office-Pakete, BS, DBs, ... in Sättigungs- / Rückbildungsphase
 - Kunde hat keinen Wettbewerbsvorteil durch bestimmtes BS, etc. (= **Commodity-Software**)
 - Kunde will seine Datenhoheit zurück
 - **Offene Standards**
 - Kunde will sich keinem (Monopolisten) ausliefern (Software Assurance, ...)
 - OOo, Linux / BSD, PostgreSQL, ... gut genug

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

- ⇒ OSS ist Basis neuer Geschäftsmodelle
 - Unternehmen, die vom Lizenzgeschäft leben, wollen Störung unterdrücken
 - Software-Patente
 - FUD (Get the facts ...)
 - Ähnliche Probleme mit Medien-Industrie
- ⇒ OSS ist keine Ideologie
 - Konkurriert nicht mit Kapitalismus, ...
 - Ist nicht auf Missionare + Gläubige angewiesen

Das musste ja so kommen: OSS als logische Entwicklung in Software-Märkten

⇒ Zusammenfassung

- Zyklische Marktentwicklung, Altes macht Neuem Platz
- Markt wächst rasant bei fallendem Preis
 - Mehr Kunden können sich Produkt leisten
 - Mehr Arbeitsplätze bei Produktion, Vertrieb, etc. des Produkts
- OSS-Geschäftsmodelle setzen sich bei Commodity-Software gegen Lizenzgeschäft durch

Aber wenn das jeder machen würde: Voraussetzungen für OSS

- ⇒ OS funktioniert nur für bestimmte Produkte
 - Kosten müssen im Design stecken
 - Reproduktions- + Transportkosten minimal
 - Produkt muss von "Designer"-Anzahl profitieren
 - Also nicht für Ferraris!
 - Economies of scale
 - Minimale Beiträge Einzelner -> riesige Programme

Aber wenn das jeder machen würde: Voraussetzungen für OSS

- ⇒ Programmierer benötigt zur Teilnahme
 - Physische + soziale Sicherheit (Rechtsstaat, Einkommen, Patente?)
 - Mindestbildung (Lesen + Schreiben können, Englisch können)
 - Erschwingliche Hardware (PC)
 - Erschwingliche Kommunikation (Internet)
- ⇒ -> Entwickler haupts. in EU + USA + Japan (www.debian.org/devel/developers.loc)

Aber wenn das jeder machen würde: Voraussetzungen für OSS

- ⇒ Warum arbeitet Programmierer kostenlos?
 - Lösen des eigenen Problems ("scratch an itch")
 - Handwerkerstolz
 - Neugier, Lernen
 - Anerkennung von anderen Programmierern
 - Türöffner für Job
 - Ist Altruismus langfristig kalkulierter Eigennutz?

Aber wenn das jeder machen würde: Voraussetzungen für OSS

- ⇒ Warum arbeitet Unternehmen kostenlos?
 - Arbeitet garantiert **nicht** kostenlos!
 - **OK**, business-enabler (enabling technology = Infrastruktur) in OSS zu wandeln (z. B. SAP DB)
 - **Nicht OK**, differentiating technology in OSS zu wandeln (z. B. Googles Wertungsalgorithmus)
 - Bugfixes, neue Features zurückgeben effektiver als eigenes Patch-Management

Aber wenn das jeder machen würde: Voraussetzungen für OSS

- ➔ Warum arbeitet Unternehmen „kostenlos“?
 - Strategie, z. B. MS von seinen Office- und BS-Einkünften abschneiden = Quersubventionierung der Konkurrenz eigener Produkte unterbinden
 - (Strategie MS: Office -> Office Live = reklamefinanziert
Verk. von Infrastruktur-Lizenzen -> Integration)
 - Selbstdarstellung - wir sind die Guten!

Aber wenn das jeder machen würde: Voraussetzungen für OSS

⇒ Zusammenfassung

- Open Source funktioniert für digitale Güter
 - Software, Wikipedia, ...
- Technische Voraussetzungen
 - Internet, ...
- Für Einzelkämpfer: Wohlstand Voraussetzung
 - Weniger Einzelkämpfer als man denkt!
- Zunehmend durch Unternehmen finanziert

Code verschenkt und trotzdem reich + glücklich: OSS-Geschäftsmodelle

⇒ Duale Lizenzen

- MySQL, QT / Trolltech
- Software nur an "uninteressante" Kunden verschenkt, der Rest zahlt

⇒ Endanwenderfinanziert

- eBay, Google
- Software keine differentiating technology (BS, Webserver)
- Mehr Programmierer mit Kenntnissen für OSS-Versionen

Code verschenkt und trotzdem reich + glücklich: OSS-Geschäftsmodelle

⇒ Hardware-Anbieter

- IBM, HP, Sun, LinkSys
- Anbieternutzen
 - Hardware ist differentiating technology, nicht Software
 - BS-Implementierung + Pflege sehr kostenintensiv
- Kundennutzen
 - Mehr Software für Linux als für proprietäres BS verfügbar
 - Linux dank Quelltext anpassbar
 - Umstieg auf andere HW / Anbieter einfacher

Code verschenkt und trotzdem reich + glücklich: OSS-Geschäftsmodelle

⇒ Öffentliche Verwaltung, Forschung >

- Beispiele

- U.S. Department of Defense: ursprünglicher Förderer von BSD-Unix
- National Center for Supercomputing Applications (NCSA): Mosaic (1. Webbrowser)
- Stadt München: LiMux, z. B. Basis-Client, Software-Verteilung + -Konfiguration
- BSI: Kroupware (Groupware, ähnlich Exchange-Server)

Code verschenkt und trotzdem reich + glücklich: OSS-Geschäftsmodelle

- ⇒ Öffentliche Verwaltung, Forschung >
 - Innovation, (Grundlagen-) Forschung
 - Unabhängigkeit von Software-Unternehmen, insbesondere in anderen Ländern
 - Erhöhte Sicherheit (keine Backdoors)
 - Für manche Länder sind Lizenzkosten zu hoch (Mexiko, China, ...)
 - Kann Software ohnehin nicht verkaufen

Code verschenkt und trotzdem reich + glücklich: OSS-Geschäftsmodelle

- ⇒ Öffentliche Verwaltung, Forschung
 - Benötigt oft ähnliche Funktionalität, kann sich aber im Projekt nicht einigen
 - Evtl. verpflichtet, mit Bürgern durch offene Standards zu kommunizieren
 - Vorsicht, Patente trotz öffentlicher Förderung: Eolas - mit BSD-Lizenz wär' das nicht passiert!
- ⇒ Service
 - MySQL, JBoss
 - Auch Red Hat + Novell / SUSE nicht mehr im Lizenzgeschäft -> Support, Zertifizierung, ...

Aber das geht doch nicht: Vermeintliche OSS-Probleme

⇒ Trittbrettfahrer

- Verursachen keine zusätzlichen Kosten (verwendete Ressourcen nicht knapp - keine "Tragedy of the Commons")
- Jeder Mitwirkende fängt als Trittbrettfahrer an

⇒ Unsicherheit durch Software-Patente

- Software-Patente sind das Problem, nicht die (OS-) Software (Eolas-Patent)
- Gleiche Unsicherheit bei Inhouse-Entwicklung
- Kein Endkunde wurde verklagt, "nur" Hersteller

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ⇒ Vernichtet Firmen + Arbeitsplätze
 - Endkunde finanziert Software, nicht der Hersteller
 - Endkunde investiert gesparte Lizenzkosten anderweitig, Geld löst sich nicht in Luft auf
 - Bedarf an Software sinkt nicht, Programmierer findet anderen Arbeitsplatz
- ⇒ Schlechtere Treiberversorgung
 - Auf dem Weg der Besserung
 - Bessere Peer-Unterstützung bei Problemen

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ⇒ Unsicherheit, weil Individuen + Staaten böartigen Code einschleusen können
 - Problem besteht auch beim Outsourcing von CS
 - Bewährte Kontrollen durch Maintainer, "unerwartete" Audits problemlos möglich
- ⇒ Schlechtere Dokumentation, GUIs
 - Ja, Programmierer interessieren sich mehr fürs Programmieren
 - Besserung, weil OSS-Motivation jetzt auch bei Konzernen, Redakteuren, Grafikern
 - Bessere Peer-Unterstützung bei Problemen

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ⇒ Keine vorhersagbare Produktpolitik >
 - Beispiel Textverarbeitung:
 - GNU Emacs existiert seit 1984, Emacs seit 1976
 - vi existiert seit 1976
 - WinWord existiert seit 1989
(1. Version 4 Jahre später als angekündigt!)
 - WordStar, WordPerfect: RIP

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ⇒ Keine vorhersagbare Produktpolitik >
 - Beispiel BS:
 - BSD existiert seit 1977
 - FreeBSD, NetBSD existieren seit 1993
 - GNU/Linux existiert seit 1994
(Linux Kernel 1.0, GNU ex. seit 1984, HURD :-)?)
 - Windows NT existiert seit 1993
(Entwicklungszeit 5 Jahre statt geplanter 18 Monate!)
 - AmigaOS, Atari TOS, ...: RIP

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ➔ Keine vorhersagbare Produktpolitik
 - CS-Marketing-Abteilung kann weissagen?
 - Auch Unternehmen ändern ihre Produktpolitik - ohne Ausweg für Kunden ohne Quelltext
 - OSS-Produktpolitik vom Markt bestimmt (Fork, z. B. XFree / X.org)
 - Technisch sinnvolle Lösung gewinnt, da Entscheidung Programmierer-gesteuert
 - Hierarchien (Firmen) skalieren schlechter als Netze (Community) und sterben schneller

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ⇒ Mehr Bugs
 - CS-Anbieter hat Interesse am Verschweigen von Bugs, OSS-Community nicht
 - Bugs werden sofort gefixt, nicht erst am Patch-Day, oder am nächsten, oder ...
- ⇒ Höhere Einarbeitungskosten
 - Meint oft Migrationskosten - die wären beim umgekehrten Migrationsweg nicht geringer
 - Vergleich oft nur mit Lizenzkosten für Upgrade - Nutzen ist aber strategisch (Datenhoheit, ...)
- ⇒ Keiner haftet für Fehler
 - Sie haben Bill Gates für den letzten Datenverlust erfolgreich verklagt?

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ➔ Dann muss ich alle meine Software auch zu OSS machen >
 - Vermeintliches Problem besteht nur bei GPL-, nicht bei LGPL- oder BSD-Lizenz
 - Nein, Quelltext nur beim **Weitergeben** geänderter OSS mitgeben
 - Weitergabeverpflichtung nur an Empfänger der Binaries, nicht an den Rest der Welt
 - Schon die CS-EULA gelesen?
 - Schränkt Nutzung ein, das macht nicht einmal die GPL

Aber das geht doch nicht: Vermeintliche OSS-Probleme

- ➔ Dann muss ich alle meine Software auch zu OSS machen
 - Nicht bei interner Verwendung, auch wenn externe auf Server zugreifen
 - Deployment geänderter OSS auf 5000 Clients im Großunternehmen = **keine Verpflichtungen**
 - Betrieb eines (angepassten) OSS-Webservers = **keine Verpflichtungen**
 - Gilt auch, wenn externer Dienstleister Änderung macht
 - Aber
 - Auch kostenlose Weitergabe ist Weitergabe
 - Auch indirekte Weitergabe, z. B. in Hardware-Produkt, ist Weitergabe

Literatur

- ➔ www.catb.org/~esr/writings/cathedral-bazaar/
The Cathedral and the Bazaar, auch als Buch von O'Reilly erhältlich
- ➔ www.opensourcejahrbuch.de/2005/
OSS Jahrbuch 2005, auch als Buch von Lehmanns Media erhältlich

Danke

➔ Danke für Ihre Aufmerksamkeit

... und jetzt die Fragen!