

Open Source Software

Möglichkeiten und Grenzen

Frank Klapper, CIO, Universität Bielefeld
Hamburg, 23.10.2008



1

Open Source Software

Begriffsdefinition **Open Source Software** (OSS):

- Freier Zugang zum Quellcode
- Beliebige Modifikation der Software
- Freie Weitergabe von Software (unverändert oder modifiziert)
- Keine Einschränkung der Nutzung

(Definition der Open Source Initiative (OSI))

zentraler Gedanke hinter Open Source Software:

- (Weiter-)Entwicklung von qualitativ hochwertiger Software sicherstellen
- Mehrfache Entwicklung derselben Komponenten verhindern
- aber nicht: eine kostenlose Alternative zu kommerzieller Software zu schaffen



2

„Vorteile“ von Open Source Software

- Anpassbarkeit
 - Anstelle einer Neuentwicklung
- Wiederverwendbarkeit von Code
 - Einsparen von Entwicklungszeit; KnowHow-Transfer
- Höhere Produktqualität
 - Anderer Entwicklungsprozess als bei kommerzieller Software; keine Marktzwänge, keine festen Veröffentlichungstermine
- Anbieterunabhängigkeit
 - Kein Abhängigkeitsverhältnis von bestimmtem Hersteller
- Höhere Sicherheit
 - Sicherheitslücken werden schneller erkannt, da Quellcode öffentlich
- Offene Standards
 - Dateiformate und Schnittstellen über die Quelltexte offen gelegt
- Keine Lizenzkosten



3

„Nachteile“ von Open Source Software

- Keine Gewährleistungsansprüche
- (In der Regel) kein Support durch Entwickler
 - Dienstleistungen durch Dritte; wachsender Markt
- Höherer Schulungsaufwand
 - Mehr KnowHow über kommerzielle Produkte aufgrund größerer Verbreitung
- Ungewisse Weiterentwicklung
 - Keine Verpflichtung der Entwickler zu Pflege und Wartung
- Benötigte Applikationen sind nicht verfügbar
- Mangelhafte Interoperabilität mit kommerzieller Software
 - Dateiformate und Schnittstellen kommerzieller Software sind oft nicht bekannt



4

Das Geschäftsmodell hinter Open Source Software

- Beratungs- und Support-Dienstleistungen im Umfeld von OSS
 - Information / Beratung
 - Zu Funktionen, Ablauf, und Einsatzszenarien
 - Zur Kopplung mit anderen Systemen
 - Unterstützung
 - Bei der Einführung (neue Software und/oder neue Version)
 - Beim Betrieb
 - Bei der Verbindung mit anderen Systemen
 - Fehlerdiagnose und –behebung
 - Bei der Einführung (neue Software und/oder neue Version)
 - Im laufenden Betrieb
 - Garantieleistung zur Fehlerbehebung innerhalb einer Frist
- Verkauf von Linux-Versionen eigener proprietärer Software



5

Konfektionierung von Open Source Software

- Es gibt eine kaum überschaubare Menge freier Software
 - Systeme von Spitzenqualität
 - Wenig stabile Nischenprodukte
- Softwareauswahl und Komposition sind für den einzelnen Anwender (zu) komplex
- (Linux-)Distributionen als Ordnungsmechanismus
 - Eigene Dienstleistungsbranche (→ Geschäftsmodell)
 - Kombination von hunderten an Systemkomponenten, Tools und Anwendungen zu einem konsistenten Paket
 - Herstellen von Übersichtlichkeit
 - Gewährleistung von Qualität und Zusammenspiel
 - Für technisch „weniger versierte“ Anwender einfach zu installieren und zu konfigurieren
 - Beispiele: Red Hat, SuSE, ...



6

Einsatzbereiche von Open Source Software

Breiter Einsatz

- Server-Betriebssysteme
- Datenbanken
- Entwicklungssysteme
- Web Browser und –Server
- Content Management Systeme
- eLearning Systeme
- Cluster und Grid Software

Geringer Einsatz

- Client-Betriebssysteme
- Office Produkte
- Grafik Software
- Wissenschaftlich-Technische-Software

Keine nennenswerte Durchdringung
in anderen Einsatzbereichen



7

Open Source Software ist nicht nur Linux

- Open Source Software wird oft mit dem Open Source Betriebssystem Linux gleichgesetzt
- Die Umstellung auf Linux ist aber KEINE zwingende Voraussetzung für den Einsatz von Open Source Software
- Beispiele von Open Source Anwendungen, die auch unter Windows betrieben werden können:
 - OpenOffice.org
 - Apache Web-Server
 - Mozilla Web-Browser
 - MySQL Datenbank
 - PHP Skriptsprache
 - ...



8

Wirtschaftliche Aspekte

- Ein Kostenvergleich ist nur auf Basis der TCO (total cost of ownership) möglich
- Es gibt unzählige Studien; jedes Ergebnis ist möglich
- Die TCO muss immer situationsbezogen ermittelt werden
- **Es hilft nur selbst rechnen!**

Zwei Beispiele:

- Fraunhofer Studie (2005; ca. 10.000 Client-Rechner):
bessere TCO bei open source;
aber „nur“ 2,4% Einsparpotential
- Linux München:
Kosten für open source und Microsoft nahezu identisch



9

Details der Fraunhoferstudie

	Microsoft basiert	Open source basiert
Clients:	MS Vista Professional MS Office MS Outlook	Linux OpenOffice.org OpenGroupware.org
Mailserver:	MS Vista Server MS Exchange Server	Linux OpenGroupware.org
File-/Print-Server:	MS Vista Server	Linux

- Einsparpotential bei open source software:
 - Lizenzkosten (7,0 % der TCO)
- Mehraufwand bei open source software:
 - Schulungen (2,7% der TCO)
 - Reduzierte Mitarbeiterproduktivität (0,7 % der TCO)
 - Einführungsunterstützung (0,2 % der TCO)
 - Upgrades alter Software (1,0 % der TCO)



10

Linux München: Worum geht es?

- Auslöser: Umstellungsbedarf weg von Windows NT4 (ab 1993)
- Strategie: Unabhängigkeit von Softwarelieferanten
- Zugleich: Umstellung als Change zur Konsolidierung (Infrastruktur, Prozesse, Applikationen)
- Linux steht für
 - Linux-basierter Basisclient
 - ODF als Dokumentenformat
 - Fachanwendungen künftig plattformunabhängig, Präferenz webbasiert
- Linux ist kein primär technisches Projekt; Change Management steht im Vordergrund
 - Intensive Kommunikation, Veränderungsbereitschaft schaffen
 - Schwerpunkt bei Mitarbeiterschulung und Kompetenzaufbau



11

Linux München: Projektstatus

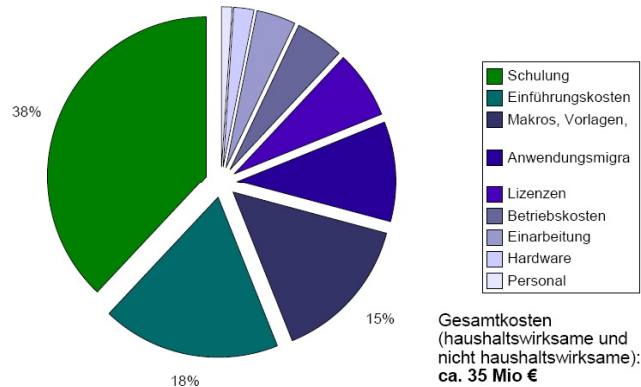
- Ausgangslage
 - Mehr als 14.000 PCs in 21 eigenständigen Organisationseinheiten
 - Vielzahl unterschiedlicher Produkte und Verfahren
 - 13.000 Vorlagen, Makros, Formulare, ...
- Bisher erreicht:
 - Alle Clients arbeiten mit Firefox und Thunderbird
 - 8.000 Clients arbeiten mit OpenOffice
 - 1.200 Clients arbeiten auf der Basis von GNU/Linux
 - Wollmux als einheitliche Vorlagenlösung (Eigenentwicklung)
 - Anzahl der Vorlagen, Makros und Formulare um 30% reduziert
 - Verbesserte Geschäftsprozesse → Einsparung von 80 FTEs



12

Linux München: Zeitplan und Kosten

- Zeitplan
 - Vorstudie 2001-2003, Feinkonzept 2003-2004
 - Migration seit 2005, Projektlaufzeit bis 2012
 - Umstellung von 80% der Clients bis 2012
- Kosten



13

Softwareauswahl: IT-Strategie als Basis

- Die IT-Strategie muss verschiedene Ebenen berücksichtigen
 - Fachliche Ebene
 - Organisatorische Ebene
 - Technische Ebene
- Framework für die Technische Ebene
 - Welche Softwarebestandteile werden für welche Funktionalitäten eingesetzt?
 - Welche Formate und Schnittstellen werden genutzt?
 - Wie arbeiten die Komponenten des Frameworks zusammen?
- Einsatz von (Open Source) Software auf Basis der IT-Strategie



14

Softwareauswahl: weitere Kriterien

- Passender Funktionsumfang
 - Bedienerfreundlichkeit
 - Zuverlässigkeit, Fehlerfreiheit, Sicherheit, Stabilität
 - Integration mit dem bestehenden Umfeld
 - Supportmöglichkeiten
 - Kosten für Lizenzen, Einführung und Support
 - Zukunftsfähigkeit
- Ausgangsbasis ist immer eine Aufgabe, die mit geeigneten IT-Mitteln gelöst werden soll



15

Freiheiten von Open Source Software innerhalb der Organisation einschränken

- Freier Zugang zum Quellcode &
- Beliebige Modifikation der Software
 - Unüberschaubare Menge an Varianten
 - Arbeiten werden mehrfach durchgeführt
 - Planung & Koordination durch „Produktverantwortliche“
- Freie Weitergabe von Software (unverändert oder modifiziert)
 - Unterschiedliche Versionsstände
 - Managementsystem zur Softwareverteilung
- Keine Einschränkung der Nutzung
 - Unüberschaubarer Softwareeinsatz
 - Softwareeinsatz auf der Basis eines Frameworks



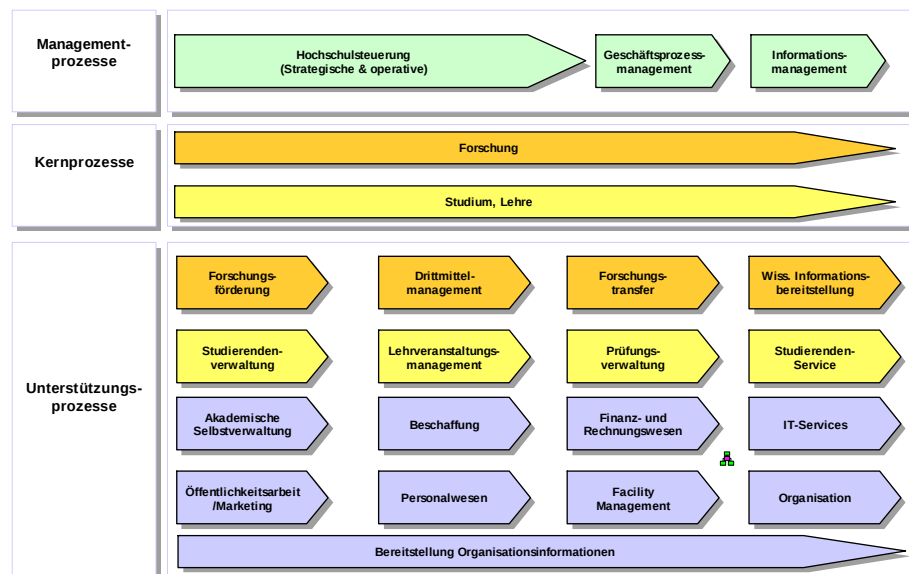
16

Rolle von OSS beim Aufbau eines Integrierten Informationsmanagements

- Integriertes Informationsmanagement → „neue“ Aufgaben
 - Prozessebene
 - Erfassung und Dokumentation der Geschäftsprozesse
 - Abstimmung zwischen den Anforderungen der Geschäftsprozesse und den Möglichkeiten der IT-Verfahren
 - Datenebene
 - Konzeption von logischen Datenmodellen
 - Konzeption von Schnittstellen zwischen IT-Verfahren
 - Datenkonsolidierung / Vermeidung redundanter Datenhaltung
 - Infrastrukturebene
 - Harmonisierung von IT-Infrastruktur und IT-Verfahren
 - Portfolio-Management
- Software spielt eine nachgeordnete Rolle
- Schwerpunkte liegen in den Bereichen Strategie und Konzeption

17

Kaum Open Source Software für organisationsbezogene Prozesse



18

Open Source Software und HIS

„Eine Aktivität von HIS besteht in der Identifizierung und Erprobung vorteilhafter Open Source-Komponenten für die Integration in die Hochschul-IT“

- HIS nutzt verstärkt Open Source Komponenten
 - QIS und LSF sind auf reiner Open Source Basis einsetzbar
 - HIS nutzt Erkenntnisse aus Open Source-Entwicklungsprozessen
 - HIS wird das Gesamtpaket der HIS-Software NICHT für jedermann veröffentlichen
 - aber: Entwicklungspartnerschaften → Zugang zum Quellcode
 - an Open Source-Projekte eng angelehnte Vorgehensweise
- „Open Source ist ein kalkulierbarer strategischer Faktor in IT-Lösungen für das Hochschul-Management geworden“



19

Fazit

- Open Source Software
 - Ist eine ernst zu nehmende Alternative zu proprietärer Software
 - Lebt von einer aktiven Community
 - Kann heute nicht alle Anwendungsbereiche abdecken
- Die Lizenzkosten sind nur ein kleiner Teil der TCO
- Mängel in der IT-Strategie, unzureichende Konzepte und organisatorische Mängel können nicht durch (Open Source) Software kompensiert werden



21

