

FOSS Geschäftsmodelle

Inhalt

- Hardware vs. Software
- Restriktive Vertriebsbedingungen
- Open Source Software
- Lebens- und Entwicklungszyklus von Software
- Freie Software Lizenzen; Copyleft
- Open Source Projekte
- FOSS Geschäftsmodelle
- Entwickler, Anwender, Dienstleister



WhereGroup

Copyright: WhereGroup GmbH & Co. KG.

GNU FDL Lizenz <http://www.gnu.org/licenses/fdl.txt>

Hardware vs. Software

Der Unterschied zwischen Hardware und Software ist wesentlich tiefgreifender, als die Ähnlichkeit der Begriffe vermuten lässt.

- Wenn der Hersteller einer Hardware diese Ware weggibt, erleidet er den Verlust dieser Ware. Üblicherweise wird dieser Verlust ausgeglichen, z.B. durch Zahlung von Geld.
- Kaputte Hardware ist unbrauchbar und wertlos.
- Hardware kann nicht dupliziert werden. Jede "Kopie" einer Hardware benötigt genauso viel Rohstoffe wie die Vorlage und ist auf atomarer dennoch Ebene nie exakt.
- Hardware nutzt sich ab, rostet, zerbricht und wird aufhören zu funktionieren.
- Wenn die Kopie eines Software "Produktes" weggegeben, entsteht kein Verlust, da die Kopiervorlage nicht verloren geht. Es sind nur exakte Kopien.
- Software kann nicht endgültig kaputtgehen, weil es immer noch eine Kopie gibt.
- Software kann beliebig oft verlustfrei dupliziert werden. Jede Kopie einer Software ist immer identisch. Das "Rohmaterial", der Quelltext ist unbegrenzt haltbar.
- Software nutzt sich nicht ab, rostet nicht, zerbricht nicht und hört nie auf zu funktionieren.

Restriktive Vertriebsbedingungen

Software unterliegt nicht den Bedingungen materieller Gegenstände ist deshalb inkompatibel zu den üblichen Praktiken der Marktwirtschaft.

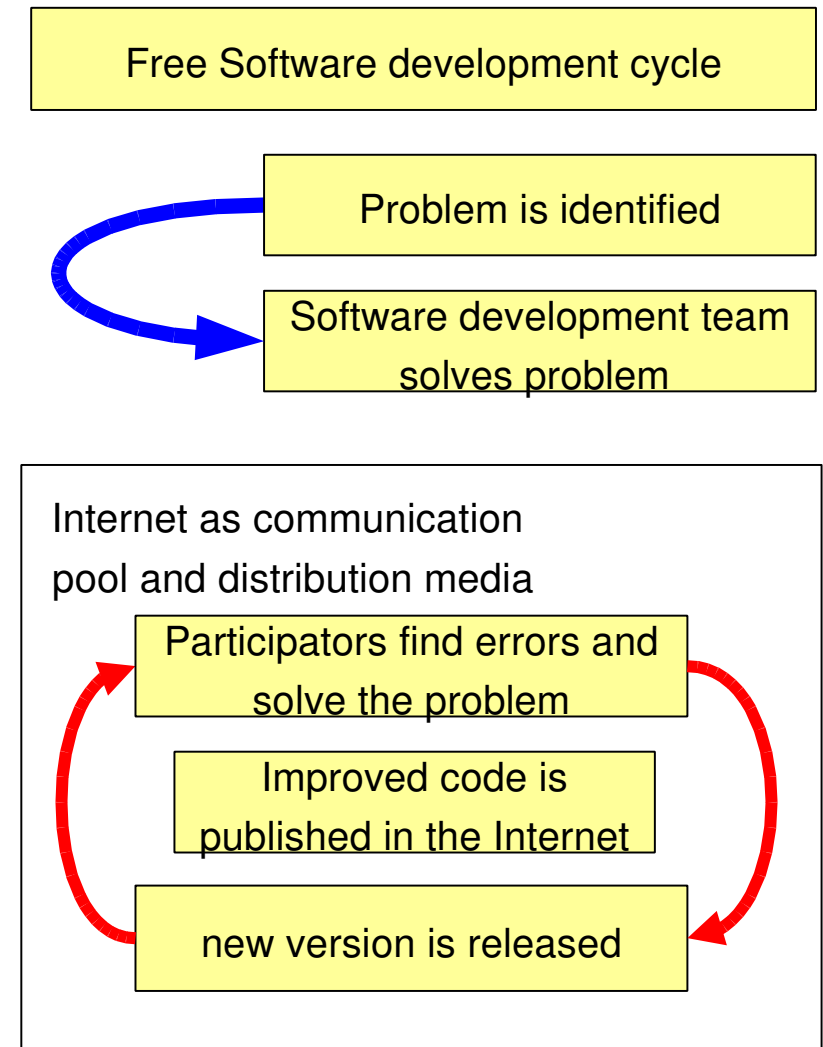
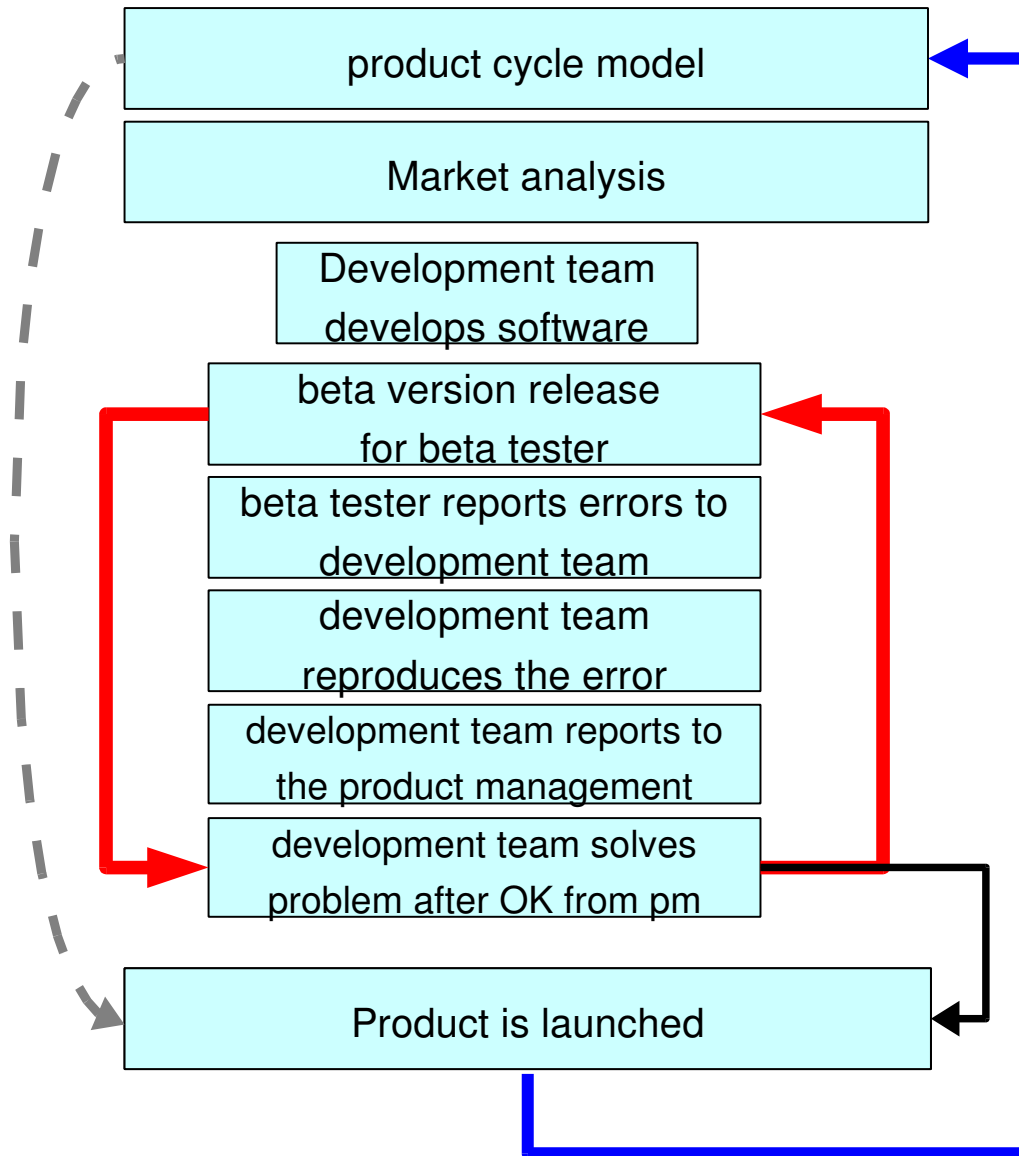
- Marktpreise (Stückpreise) richten sich nach Angebot und Nachfrage. Bei einem Überangebot sinkt der Preis. Bei steigender Nachfrage steigt der Preis.
- Bedingt durch die verlust- und fast kostenfreie Kopierbarkeit von Software tendiert der Stückpreis von Software gegen Null.
- Deshalb werden die Stückpreise durch Verknappung der Resource Software (Anwendung proprietärer Lizenzen) künstlich hoch gehalten.
- Nur ein Monopol gestattet ein Preisdiktat. Das Ziel proprietärer Lizenzpolitik muss deshalb immer ein Monopol sein (z.B. auch durch Patente).

Open Source Software

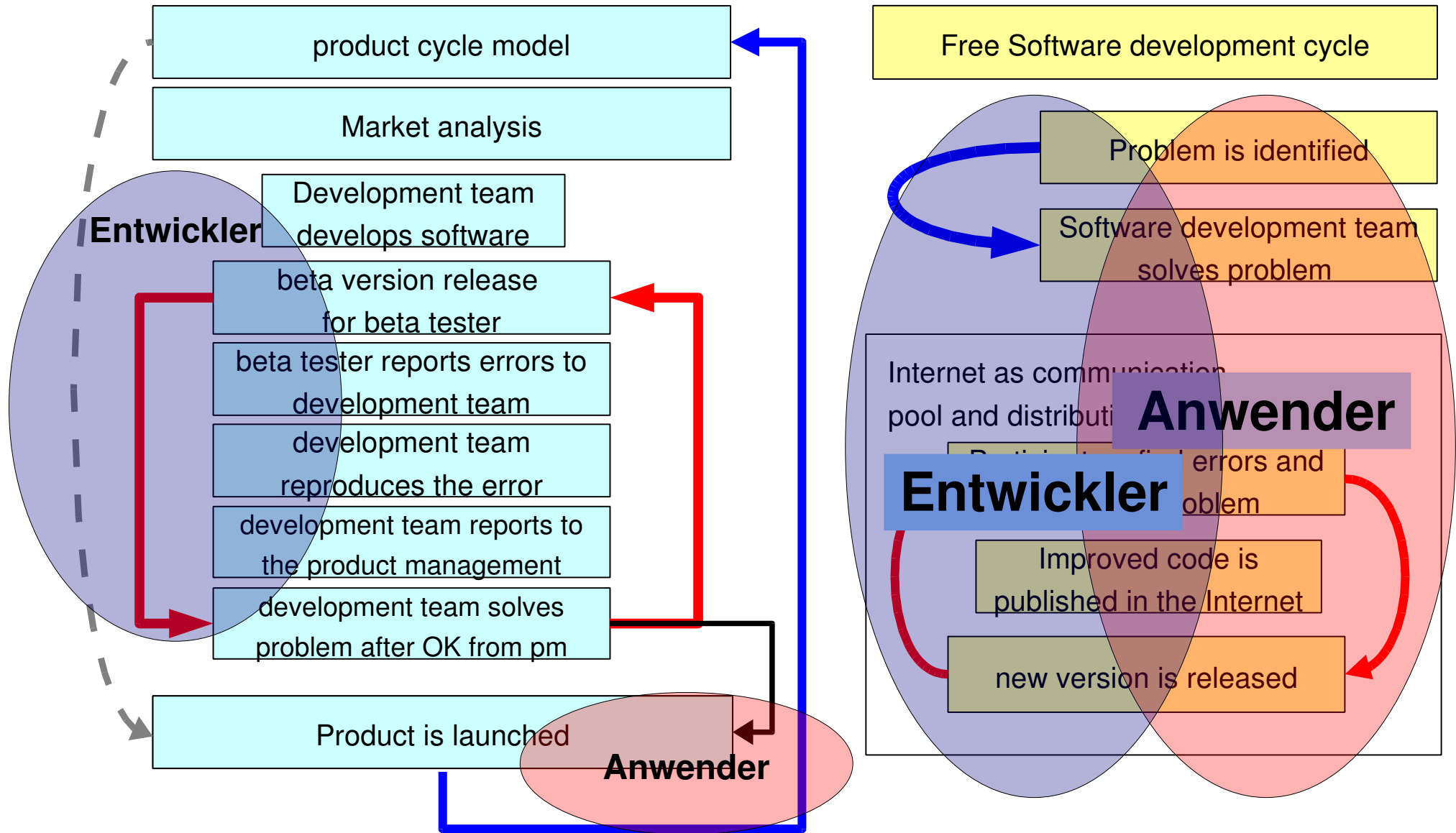
Um den speziellen Eigenschaften von Software als Wissens-Resource gerecht zu werden gibt es neue Geschäftsmodelle, die durch Lizenzmodelle geschützt sind.

- Die von Entwicklern geschriebene ursprüngliche Form der Software heißt "Quellcode" oder englisch "Source Code". Darin sind alle Leistungen des Programms vollständig und nicht verschlüsselt enthalten.
- Der Begriff "Open Source Software" (OSS) definiert, dass der Quellcode offengelegt wird. Hier wird der Quellcode als die begrenzte Resource erkannt, nicht die Stückzahl kompilierter, proprietär lizensierter Software.
- Die OSI (Open Source Initiative <http://www.opensource.org/>) definiert erweiternd zur Offenlegung des Quellcodes, dass Anwender Software individuell weiterentwickeln oder verändern, Schwachstellen oder Fehler veröffentlichen und beheben können.

Lebenszyklus vs. Entwicklungszyklus



Kommunikation und Schnittstellen



Freie Software Lizenzen

Freie Software ist ein Begriff, der einen Typ von Lizenz kennzeichnet. Der Begriff an sich ist nicht geschützt und unterliegt keinem Copyright, ist aber sehr klar definiert. Um als Freie Software zu gelten müssen vier Freiheitsgrade erfüllt sein:

- Die Freiheit das Programm für jeden Zweck zu benutzen.
 - Die Freiheit zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann. Der Zugang zum Quellcode ist dafür Voraussetzung.
 - Kopien zu verbreiten, so dass man anderen Anwendern unkompliziert weiterhelfen kann.
 - das Programm zu verbessern, und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen. Der Zugang zum Quellcode ist dafür Voraussetzung.
- (Ein Programm ist dann Freie Software, wenn die Benutzer alle diese Freiheiten haben.

Der Copyleft Effekt

Es gibt verschiedene Freie und Open Source Software Lizenzen. Die General Public License (GNU GPL) implementiert den sogenannten Copyleft – Effekt.

Die Software wird dabei zunächst unter das Copyright gestellt und damit rechtlich schützbar. Dann werden Vertriebsbedingungen hinzugefügt, die vorschreiben, dass jeder Mensch den Quellcode und daraus erstellte Software nutzen und auch öffentlich weitergeben darf, allerdings nur unter den gleichen Vertriebsbedingungen. Damit werden Software und Lizenz untrennbar miteinander verbunden.

Weitere Informationen:

Freie Software: <http://www.fsf.org>

Open Source: <http://www.opensource.org>

Open Source Projekte

- Open Source Projekte sollten bestimmten Kriterien folgen, um den Bedingungen einer offenen Entwicklung gerecht werden zu können.
- Durch die Lizenz werden lediglich ein Teil Bedingungen vorgeschrieben, vornehmlich die der Distribution. Die Lizenz sagt noch nichts über die Form der Projektsteuerung, die "Politik" aus.
- Erst eine Organisation um das Projekt, die sich an bestimmte Regeln hält macht aus einer Software, die als Open Source lizenziert ist ein echtes Open Source Projekt.
- Die [Open Source Geospatial Foundation](#) (OSGeo) hat deshalb einen Kriterienkatalog angelegt, der erfüllt werden muss, um den offiziellen Status eines Open Source Projektes zu erhalten.
- Erst wenn alle Kriterien erfüllt sind kann ein Projekt als offizielles OSGeo Projekt firmieren.

FOSS Geschäftsmodelle

- Der erste Schritt zur professionellen Einführung von Software beinhaltet umfangreiche, herstellerunabhängige **Beratung**
- Die reine **Installation** der Software ist heute weitgehend automatisiert (Click & Go). Die Herausforderung liegt in der **Systemintegration**.
- **Schulungen** vermitteln das Wissen zur Selbsthilfe und können je nach Vorwissen individuell dosiert werden.
- Die **Anpassung** an individuelle Fragestellungen helfen das volle Potential der Software auszunutzen.
- **Support** und **Wartung** sichern die Investition in die Anwendungen
- Die **Entwicklung neuer Funktionen** (sowie Dokumentation, Debugging, Erstellung von Tutorials, etc.) ist die begrenzte Resource, nicht Kopien der fertigen Software.

FOSS und die Entwickler

Für Entwickler ist Open Source das natürliche Entwicklungsmodell, da es die Arbeit wesentlich erleichtert.

- Das Rad muss nicht jedes Mal neu erfunden werden.
- Probleme können offen und öffentlich diskutiert werden.
- Entwicklergemeinschaften können sich unkompliziert und ohne marktwirtschaftliche Erwägungen bilden und auflösen
- Quelltext kann wiederverwendet werden.
- Es macht einfach furchtbar viel Sinn.
- und so weiter...

FOSS aus Anwenderperspektive

Anwender können eine große Palette unterschiedlicher Software nicht nur ausprobieren, sondern auch sofort uneingeschränkt produktiv nutzen.

- Kommunikation mit Nutzern und Dienstleistern hilft bei der Auswahl der "richtigen" oder am besten geeigneten Software.
- Leistungen zur Unterstützung bei der Installation und Systemintegration erfolgen wahlweise über die Gemeinschaft oder durch professionelle, kommerziell Unterstützung.
- Schulungen dienen nicht dem Erlernen einer spezifischen Software, sondern dem softwareunabhängigen Aufbau des eigenen Wissens.
- Anpassungsarbeiten können selbst vorgenommen oder eingekauft werden.
- Vertragliche Vereinbarungen sichern die Investition und ermöglichen eine gezielte Gewährleistung.
- Anwender können direkten Einfluss auf die Entwicklung der Software nehmen.

FOSS Dienstleistungen

Dienstleister können bis auf den Verkauf proprietärer Lizenznutzungsrechte die gesamte Palette an Geschäftsmodellen rund um Software anbieten. Geschätzte 90% der Umsätze im IT-Sektor werden durch Dienstleistung erwirtschaftet, nicht durch den Verkauf proprietärer Nutzungslizenzen.

- Es gilt das Handwerkerprinzip. Eine Dienstleistung wird einmalig erbracht. Das Ergebnis kann mehrfach verwendet werden, auch durch den Wettbewerb (Stichwort Coopetition aus "Competition" und "Cooperation").
- "Schnelle Hacks" helfen, unkompliziert Lösungen für spezifische Probleme zu schaffen.
- Die Entwicklung generischer Software ist für den Dienstleister dagegen am schwierigsten zu finanzieren, da diese Arbeiten über das reine "Scratch-your-Itch" hinausgeht.
- Langfristige Wartungsverträge für Software und Lösungen können die Basis für eine dauerhafte Finanzierung generischer Entwicklung bilden.

Für weitere Informationen...

...nutzen Sie die geballte Kompetenz der hier anwesenden Fachmenschen, sowohl Anwender, als auch Entwickler und Dienstleister. Fragen Sie alles was Sie wissen möchten. Heute kostet Fragen ausnahmsweise nichts.

:-)

Autor: Arnulf Christl
arnulf.christl@wherogroup.com
WhereGroup GmbH & Co. KG
Siemensstr. 8
53121 Bonn
Homepage:
<http://www.wherogroup.com>

Copyright: WhereGroup GmbH & Co. KG.

Diese Präsentation unterliegt dem Schutz der GNU FDL Lizenz und darf unter diesen Bedingungen verändert und weitergegeben werden, auch zu kommerziellen Zwecken. Das Deckblatt, der Master und diese Seite unterliegen dem Veränderungsschutz (Invariant Sections) der GNU FDL.

<http://www.gnu.org/licenses/fdl.txt>



WhereGroup