

Der Nutzen des Konzeptes Open Source bei der
Softwareentwicklung am Beispiel des
Linux-Kernels

Autor: Josua Brandhofer

In: Informatik (GK1)

Betreuungs-/Kurslehrer: Herr Berger

Name der Schule: Siegtal-Gymnasium Eitorf

Jahreszahl der Entstehung: 2015

Inhaltsverzeichnis

1	Einleitung	3
2	Definition Open Source Software und Abgrenzung	4
3	Was ist der Linux-Kernel?	5
4	Geschichte von Open Source und die des Linux-Kernels	6
4.1	Vorgeschichte	6
4.2	Freie Software und Open Source Software	7
4.3	Linux	8
5	Nutzen für die Beteiligten	9
5.1	Nutzen für das Projekt	9
5.2	Nutzen für den Verbraucher	9
5.2.1	Ein Fallbeispiel	9
5.3	Nutzen für die Programmierer	9
5.4	Nutzen für Firmen	10
6	Kritik an Open Source	10
7	Schlussenteil	10
8	Literaturverzeichnis	11

1 Einleitung

Computer sind aus dem heutigen Leben kaum noch wegzudenken. Sie werden zur Freizeitgestaltung oder bei der Arbeit genutzt. In den Industrieländern hat beinahe jeder Haushalt einen Computer. Hinter dem Computer steckt nicht nur die bloße Hardware. Noch viel wichtiger dabei ist die Software. Hinter der Softwareentwicklung oder den IT-Dienstleistungen steckt ein großer Markt, welcher immer wichtiger wird. Solche Software kann ein Spiel, ein Programm zur Steuerung komplexer Anlagen oder ein ganzes Betriebssystem sein. Als bekanntestes Betriebssystem gilt dabei das proprietäre Betriebssystem „Windows“ von Microsoft. Dieses wird kommerziell vertrieben und wird auf den meisten Privat Computern vorinstalliert. Daneben gibt es aber auch die sogenannte Open Source Software. Prominente Beispiele dafür sind z. B. der Internet Browser „Mozilla Firefox“ oder das Betriebssystem „Linux“ mit all seinen verschiedenen Distributionen. Diese können von jedem, anders als bei „Windows“, unentgeltlich genutzt und weiterentwickelt werden. Meistens wird „Linux“ aber eher versteckt genutzt. So nutzt man eigentlich täglich Linux wenn man „Google.com“ benutzt, da die Server von Google mit einem Linux-Betriebssystem laufen. Auch bei Supercomputern wird Linux sehr häufig genutzt. Aktuell werden 94,6% der Supercomputer zumindest teilweise mit Linux betrieben¹. Bei diesem häufigen Vorkommen kam auch Herr Prof. Dr. Schwalbe zu dem Schluss: „Somit wird deutlich, dass auch Open Source Software für viele Bereiche des Lebens bedeutend ist und aus der heutigen, durch Informationstechnologien geprägten Welt, kaum noch heraus zu denken ist.“ (Schwalbe, 1. Einleitung). Bei so einem großen Erfolg von Open Source Software stellt sich die Frage: Inwiefern hilft das Konzept von Open Source Software ein so gutes Produkt² zu erstellen, welches häufig kostenlos ist? Inwiefern ist dieses Konzept vom Ergebnis her besser als proprietäre Software? In dieser Facharbeit möchte ich mich mit dieser Frage auseinandersetzen und dabei mehrere Texte untersuchen. Dabei werde ich mich am Beispiel des Linux-Kernels orientieren. Zu Anfang werde ich erstmal eine kurze Definition von Open Source Software geben und sie dabei von anderen prominenten Softwaretypen abgrenzen. Zusätzlich gebe ich eine kurze Darstellung des Linux-Kernels. Darauf werde ich im 4. Kapitel die Geschichte von Open Source Software erörtern, wobei ich mich auf die des Linux-Kernels konzentriere. Im nachfolgenden Kapitel werde ich den Nutzen des Open Source Konzepts für verschiedene Beteiligte untersuchen. Dies wären das Projekt an sich, die Verbraucher, für die die Software gemacht wurde, die teilnehmenden Programmierer und schlussendlich die Firmen, die die Open Source Software nutzen oder mitentwickeln. Im letzten Kapitel werde ich mich mit Kritik

¹vgl.: Wikipedia; Supercomputer; 2015; Abschnitt: Betriebssysteme

²So gut, dass Microsoft es als große Bedrohung ansieht(siehe auch Halloween-Dokumente)

an Open Source Software auseinandersetzen.

2 Definition Open Source Software und Abgrenzung

Open Source Software ist, wenn man es wörtlich übersetzt, Software, deren Quelltext offengelegt ist. Man sollte dies aber nicht mit gemeinfreier Software, welche nicht lizenziert wird und allen kostenlos zugänglich ist, verwechseln. Denn Open Source Software wird sehr wohl lizenziert. Dabei gelten laut der Open Source Initiative (im folgenden OSI) für Lizenzierung im Sinne von Open Source folgende 3 Grundmerkmale (laut Wikipedia, Open Source, Abschnitt: Definition der OSI): 1. „Die Software (d. h. der Quelltext) liegt in einer für den Menschen lesbaren und verständlichen Form vor.“, 2. „Die Software darf beliebig kopiert, verbreitet und genutzt werden.“ und 3. „Die Software darf verändert und in der veränderten Form weitergegeben werden.“³. Also gibt Open Source Software seinen Nutzern weitaus mehr Freiheiten bei der Softwarenutzung als proprietäre Software, welche ihren Quelltext meistens geheimhält. Diese weitergehenden Rechte bedeuten aber nicht, dass die Software unbedingt kostenlos angeboten wird. Die mögliche Missinterpretation des Wortes frei in der zur Open Source Software weitgehend identischen freien Software hat primär zur Entstehung der OSI beigetragen. Daher stammt auch die Aussage der Free Software Foundation (FSF) über die Bedeutung des Wortes „frei“: „free speech, not free beer“ (freie Meinungsäußerung, nicht Freibier⁴). Nichtsdestotrotz wird sehr viele Open Source Software kostenlos angeboten⁴. Die meisten Open Source Projekte werden unter der von der FSF erstellten GNU General Public License veröffentlicht. Open Source muss man zwangsläufig zu der sehr ähnlichen freien Software abgrenzen. Diese sind zwar sehr ähnlich, nur garantiert freie Software mehr Freiheiten für den Nutzer als Open Source Software⁵. Der größte Unterschied besteht aber in der Bedeutung der Begriffe und in den Werten, den sie vermitteln: Open Source zeigt die freie Verfügbarkeit des Quelltextes. Auch stellt die OSI primär den praktischen Nutzen dieser Entwicklungsmethode heraus und zeigt eher technische und wirtschaftliche Vorteile. Die freie Software nach der FSF vertritt eher die Rechte der Nutzer an der Software und zeigt auch sozial ethische Aspekte auf. So geht es ihr bei der Ablehnung proprietärer Software nicht bloß darum, wie gut sie trotz nicht öf-

³Bei mancher Software wird ergänzend zum 3. Merkmal noch weiterführend das Copyleft-Prinzip angewendet, um auch weiterhin die Rechte des Nutzers zu wahren. Dabei wird bewusst auf das Urheberrecht verzichtet.

⁴Das Unternehmen ID Software geht dabei einen interessanten Weg. Sie machen den Quelltext ihrer Spiele öffentlich zugänglich, z. B. auf GitHub, zugänglich und verlangen nur Geld für die graphischen und musikalischen Inhalte.

⁵Was nicht bedeutet, dass Open Source Software diese nicht bieten kann.

fentlichen Quelltextes auch sein mag, sondern behauptet, „dass proprietäre Software allein schon aus moralischen Gründen abzulehnen sei⁶“. Insgesamt sind sich Open Source und freie Software sehr ähnlich. Eigentlich ist es nur ein Konflikt aufgrund der Bezeichnung. Deshalb werden auch Begriffe wie „Free/Libre and Open Source Software“ genutzt.

3 Was ist der Linux-Kernel?

„Linux ist ein Betriebssystem-Kernel. [...] Der Kernel wird in einer Vielzahl von Betriebssystemen genutzt, die oft selbst als Linux bezeichnet werden.“ (Wikipedia, Linux(Kernel)). Einer der Hauptentwickler des Linux-Kernels ist Linus Torvalds. Als Kernel ist der Linux-Kernel eines der wichtigsten Teile des Betriebssystem. Er sorgt dafür, dass eine einheitliche Schnittstelle für die Software zur Verfügung steht, welche Hardware unabhängig ist. Der in weitgehend in C geschriebene Kernel ist ein strikt monolithischer Kernel⁷. Des weiteren bietet der Kernel 4 verschiedene Schnittstellen: die in jedem Fall stabile externe Programmierschnittstelle, die nicht stabilitätsgarantierte interne Programmierschnittstelle, die „Linux Standard Base[, welche eine Binärschnittstelle ist,] soll es ermöglichen kommerzielle Programme unverändert zwischen Linux Betriebssystemen zu portieren“ (Wikipedia, Linux(Kernel)) und schlussendlich die nicht stabile interne Binärschnittstelle. Obwohl es eigentlich nicht so geplant war, hat sich der Linux-Kernel weitestgehend in Richtung eines portierbaren Kernels entwickelt und gehört zu einem der am häufigsten portierten Systemen. Der Kernel ist seit 1992 unter der freien GPL lizenziert. Als Linux bezeichnet man Betriebssysteme, die auf dem Linux-Kernel und im Wesentlichen auf GNU-Software basieren. Auf Computern kommt Linux in Form einer Distribution vor. „Eine Distribution fasst den Linux-Kernel mit verschiedener Software zu einem Betriebssystem zusammen, das für die Endnutzung geeignet ist.“ (Wikipedia, Linux). Die hohe Anzahl an Linux-Distributionen bietet dem Endnutzer die Möglichkeit, sein auf ihn persönlich zugeschnittenes Linux zu erhalten. Das auffälligste Merkmal an Linux ist seine hohe Sicherheit vor Viren und dergleichen. Dies wird durch eine strenge Unterteilung der Zugriffsrechte erreicht, so dass die Verunreinigung des gesamten Systems schwer ist. Auch die schnelle und häufige Aktualisierung hilft, Sicherheitslücken zu schließen.

⁶Wikipedia, Freie Software, Abschnitt: Open Source

⁷D. h. werden der Quellcode und alle Treiber in das Kernelimage kompiliert. Das Kernelimage ist der ausführbare Kernel.

4 Geschichte von Open Source und die des Linux-Kernels

4.1 Vorgeschichte

Am Anfang der Computerindustrie wurde Software noch nicht als eigenständiges Wirtschaftsgut behandelt. Die Software wurde als Teil des Rechners behandelt und individuell auf den Kunden angepasst. Der Quelltext wurde oft mit der Software mitgeliefert. Es hatte also Eigenschaften von heutiger Open Source Software. Ca. zwischen 1960 und 1970 entwickelte sich an Universitäten und Forschungseinrichtungen eine so genannte „Hacker-Kultur“. Diesen leidenschaftlichen Programmierern ermöglichte die freie Verfügbarkeit des Quelltextes das Austauschen ihrer Programme und die gemeinsame Weiterentwicklung. Dies hatte für die Computerhersteller den Vorteil, dass sie „viele Vorschläge für Verbesserungen und Fehlerkorrekturen“⁸ zurück bekamen.

In den 1970er Jahren begann die Kommerzialisierung der Software. Wegbereitend dabei war das Unternehmen IBM, welches bis 1970 seine Software kostenlos veröffentlichte. „Am 23. Juni 1969 kündigte IBM neue Regeln für die Nutzung und Wartung seiner Software, getrennt von den Hardware-Nutzungsbedingungen an. Für Software wurde urheberrechtlicher Schutz in Verbindung mit Lizenzverträgen eingeführt.“⁸ Andere Firmen zogen nach und so entstand der Wirtschaftsmarkt für die Software und damit auch Softwareunternehmen, die zwar nun auch hardwareunabhängige Software herstellten, aber sie „verfolgten mit der Entwicklung von Software kommerzielle Ziele, die dazu führten, dass der Großteil der Software proprietär war und nur in kompilierter Form veröffentlicht wurde“⁹. Dies machte die Anpassung an eigenen Bedürfnisse und die private Weiterentwicklung unmöglich. Desweiteren schränkten Softwarelizenzen auch noch Nutzungszweck und Weitergabe ein.

Ein weiteres Beispiel ist die Entwicklung Betriebssystem Unix. Es ist deshalb wichtig, weil die Geschichte von Unix stark mit der Geschichte von Linux zusammenhängt. Unix wurde von der Telefongesellschaft AT&T entwickelt. Es durfte bis 1984 Unix nicht gewerblich vertreiben. Dies war für diverse Universitäten von Vorteil, da sie Unix so benutzen und für ihre Zwecke anpassen konnten. Daher bildeten sich in den 1970er Jahren „Abspaltungen vom ursprünglichen AT&T Unix heraus“⁹. In den 1980er Jahren wurde es AT&T möglich bzw. beschlossen sie, AT&T Unix als proprietäre Software zu vermarkten. „Infolgedessen durfte nun auch der AT&T-Quellcode nicht mehr öffentlich zugänglich gemacht werden.“¹⁰ Dies führte dazu, dass sämtliche Abspaltungen vom ursprünglichen Unix (allen voran BSD Unix) ihren Quelltext von

⁸Wikipedia, Freie Software, Abschnitt: Entwicklungen im Vorfeld

⁹Schwalbe, 2.3.2

¹⁰Wikipedia, Geschichte von Linux, Abschnitt: Entwicklungen im Vorfeld

AT&T Unix Quelltext bereinigen mussten um Lizenzgebühren zu entgehen, was die Softwareentwicklung sehr behindert.

4.2 Freie Software und Open Source Software

Als „Vater der Freien Software“ gilt Richard Stallman. An seinem Arbeitsplatz im Labor für künstliche Intelligenz des Massachusetts Institute of Technologie wurde in den 1970er Jahren immer mehr proprietäre Software eingeführt. Dabei störte ihn unter anderem die fehlende Möglichkeit auf den Quelltext zuzugreifen und so Fehler zu beheben. Auch die so entstehende „Monopolstellung proprietärer Anbieter“ stoß ihn ab. So versuchte er zuerst, „durch das Programmieren alternativer Software“ dem Entgegenzuwirken. Dabei wollte er einen „freien und ungehinderten Austausch von Software“ erreichen (Wikipedia, Freie Software, Abschnitt: Entwicklungen im Vorfeld). Da die Kommerzialisierung der Software weiter voranschritt, wollte er als Basis für „ein freies Software Universum“¹¹ ein freies Betriebssystem entwickeln. So kündigte er im September 1983 das GNU-Projekt an und startete es schließlich im Januar des darauf folgenden Jahres. Während dieser Zeit entwickelte er auch das Copyleft-Prinzip. Um eine Basis zu haben gründete er 1985 die Free Software Foundation und 1989 veröffentlichte er die GNU General Public License, welche „die heute am stärksten verbreitete Lizenz für Freie Software“ ist (Wikipedia, Freie Software, Abschnitt: Die Entstehung Freier Software). Bis zum Jahr 1991 stellte das GNU-Projekt beinahe alle Teile eines Betriebssystems bereit, aber ein Kernel fehlte noch. Es wurde zwar immer weiter weiterentwickelt, aber man benötigte noch diverse, meistens proprietäre, Unix-Varianten, damit es lauffähig wurde. Diese Lücke eines freien Kernels wurde später durch den Linux-Kernel geschlossen.

Nachdem er 1997 seinen Aufsatz „Die Kathedrale und der Basar“, in dem er die dezentrale Entwicklungsmethode als „Basar Methode“¹² bezeichnet und sie mit der herkömmlichen „Kathedralen Methode“¹³ verglich, publizierte, gründete Eric S. Raymond 1998 die Open Source Initiative. Zuvor hatte er sich dazu entschlossen, „dass die Freie-Software-Gemeinschaft ein besseres Marketing benötige“. Er wollte die freie Software als „geschäftsfreundlicher“ darstellen, da das „frei“ in „freie Software“ missverstanden werden konnte. Durch die Einführung des Begriffs „Open Source“ wollte er außerdem die freie Software als „weniger ideologisch belastet“ darstellen (Wikipedia, Open Source, Abschnitt: Geschichte). „Ende der 1990er Jahre gewann Open Source Software [schließlich] immer mehr an Aufmerksamkeit und wurde von zahlreichen Software- und Hardwareanbietern (IBM, HP) unterstützt sowie

¹¹Schwalbe, 2.3.3

¹²Basar, da hier alle interessierten Entwickler mitarbeiten können.

¹³Sprich die Software wird nach einem zentralen Plan entwickelt.

von öffentlichen Einrichtungen gefördert.“¹⁴

4.3 Linux

Die Geschichte des Linux-Kernels beginnt 1991 in Helsinki mit Linus Torvalds, welcher eigentlich nur an einer Terminalemulation arbeitete. Dieses Programm war hardwarenah und lief ohne Betriebssystem auf Grundlage des Minix-Systems und des GNU-C-Compilers. Während der Entwicklung merkte er, dass er eigentlich ein Betriebssystem geschrieben hatte und teilte am 25. August 1991 in einem berühmten Usenet-Posting sein Programm mit einer Minix-Gruppe. In diesem Posting wird klar, dass er damals noch nicht das Ausmaß erahnen konnte, dass seine Arbeit haben würde. So bezeichnete er sein Projekt nur als „ein Hobby“, welches nicht „groß und professionell sein [wird] wie GNU“ (Wikipedia, Geschichte von Linux, Abschnitt: Entstehung des Linux-Kernels). Auch glaubte er, dass es nie portierbar sein würde, etwas, was heute nicht mehr zu trifft, da Linux der meist portierteste Kernel ist. Am 17. September 1991 wurde die erste Linux-Version, 0.01, auf einem FTP-Server hochgeladen, „[um] anderen Leuten die Möglichkeit zu geben, am System mitzuarbeiten oder Verbesserungsvorschläge zu machen“¹⁵. Der Name „Linux“ kommt daher, dass der für den Server Verantwortliche, Ari Lemmke, den Bereich nicht, wie von Torvalds vorgeschlagen, „Freax“ nennen wollte. Er hat ihn einfach Linux genannt und Torvalds hat dies akzeptiert.

Zu Beginn stand der Linux-Kernel noch unter einer von Torvalds formulierten proprietären Lizenz. „Er merkte jedoch bald, dass das den Fortschritt der Entwicklung behinderte. Er wollte allen Entwicklern deutlich mehr Freiraum geben“¹⁶, darum kündigte er im Januar 1992 an, den Linux-Kernel unter die GNU GPL zu stellen. Dies geschah Mitte Dezember des selben Jahres mit Version 0.99 des Kernels. Nun war es möglich, Linux mit Hilfe von GNU als ein freies Betriebssystem zu veröffentlichen. Dadurch wurde viele Entwickler neugierig, da sie nun das System einfacher modifizieren und verbreiten konnten, so dass 1993 schon über 100 Entwickler am Linux-Kernel arbeiteten, was die Veröffentlichung der ersten Linux-Distribution ermöglichte. „Später sagte Linus Torvalds in einem Interview, dass die Entscheidung, Linux unter die GNU GPL zu stellen, die beste gewesen sei, die er je getroffen habe: 'Making Linux GPL'd was definitely the best thing I ever did.'“ (Wikipedia, Geschichte von Linux, Abschnitt: Linux unter der GNU GPL)

Allerdings dauerte es noch bis noch bis zum März des nächsten Jahres, bis eine, laut Torvalds Meinung, ausgereifte Version 1.0 des Kernels veröffentlicht wurde. Diese war erstmals netzwerkfähig. Von da an ging es schnell weiter. 1996 wurde Version 2.0

¹⁴Schwalbe, 2.3.3

¹⁵Wikipedia, Geschichte von Linux, Abschnitt: Der Name Linux

¹⁶Wikipedia, Linux, Abschnitt: Historische Entwicklung

veröffentlicht: „Der Kernel [konnte] nun mehrere Prozessoren gleichzeitig bedienen und wird damit für viele Unternehmen eine ernstzunehmende Alternative in vielen Arbeitsbereichen“¹⁷. Im selben Jahr wurde auch Tux als Linux-Maskötchen festgelegt. 1998 begann die Arbeit an der GUI KDE und „Viele namhafte Unternehmen wie IBM, Compaq und Oracle kündigen ihre Unterstützung für Linux an“¹⁷. 1999 wurde Linux 2.2 veröffentlicht und die Arbeit an der graphischen Benutzeroberfläche GNOME begann. Im Jahre 2000 kam ein wichtiges Programm für Linux dazu, die heute als Open Office bekannte Office-Suite StarOffice. 2001 wurden mit der 2.4er-Serie „bis zu 64 Gigabyte Arbeitsspeicher, 64-Bit-Dateisysteme, USB und Journaling-Dateisystem“¹⁷ unterstützt. Ende 2003 folgte schließlich der Linux-Kernel 2.6. Ein weiteres wichtiges Ereignis war die Gründung der Linux Foundation 2007, welche Linux bis heute fördert. Dort hat auch Linux Torvalds eine Festanstellung. Die aktuelle Kernel-Version ist 3.19 (Stand 15.02.2015).

5 Nutzen für die Beteiligten

5.1 Nutzen für das Projekt

Mehr Personen, die sich damit beschäftigen können => schnellerer Fortschritt stetige Weiterentwicklung => ständige Aktualität

5.2 Nutzen für den Verbraucher

können Rückmeldung geben (genauso wie bei jedem Software-Konzept), können vorarbeiten und konkretere Hinweise geben; können sich das Betriebssystem so anpassen, wie sie es haben wollen (kernel.org) schnelle Hilfe Mitwirkung (Veröffentliche früh. Veröffentliche häufig. Und höre auf die Benutzer. (7. Punkt von Raymond); Fast so gut wie eigene gute Ideen zu haben, ist es, gute Ideen von den Benutzern zu erkennen. Manchmal ist letzteres besser. (9. Punkt von Raymond))

5.2.1 Ein Fallbeispiel

5.3 Nutzen für die Programmierer

können schon vorhandenen Code benutzen und weiterentwickeln (kernel.org (der Quellcode dagegen in jedem Stadium über das Internet einsehbar.)); Teilen von Innovation; Gute Programmierer wissen, was sie schreiben müssen. Brillante wissen, was sie neuschreiben müssen (und was sie wiederverwenden können). (2. Punkt von Raymond)) Zusammenarbeit mit mehr Menschen können durch Quelltext lernen z.

¹⁷Wikipedia, Geschichte von Linux, Abschnitt: Chronologie

B. auch für die Entwicklung proprietärer Software; Fähigkeitsverbesserung Status Verbesserung sie haben es bei eigenen Projekten einfacher, die Kompatibilität zu gewährleisten Einfach nur Freude

5.4 Nutzen für Firmen

Nutzen Open Source Programme selber => Weiterentwicklung hilft ihnen dadurch, das diese programme häufig kostenlos sind, sparen sie viel ein

6 Kritik an Open Source

Durch das GNU-Projekt; durch die Studie; durch Microsoft(widersprechend!)

7 Schlussteil

8 Literaturverzeichnis

Prof. Dr. Schwalbe, Ulrich: **Open Source Software - Eine wirtschaftstheoretische Analyse**, überarbeitete Version von Fahrig, Thomas -
<http://www.mafabo.de/thomas/da/oss-aktuell.php> (08.08.2008)

Wikipedia: **Open Source** - http://de.wikipedia.org/wiki/Open_Source (07.01.2015)

Wikipedia: **Freie Software** - http://de.wikipedia.org/wiki/Freie_Software (31.01.2015)

Wikipedia: **Linux** - <http://de.wikipedia.org/wiki/Linux> (14.01.2015)

Wikipedia: **Geschichte von Linux** - http://de.wikipedia.org/wiki/Geschichte_von_Linux
(28.01.2015)

Wikipedia: **Linux (Kernel)** - [http://de.wikipedia.org/wiki/Linux_\(Kernel\)](http://de.wikipedia.org/wiki/Linux_(Kernel)) (07.01.2015)

Stallman, Richard: **Warum Open Source das Ziel von Freie Software verfehlt**, deutsche Übersetzung von Gehring, Robert; Kohne, Joerg -
<https://www.gnu.org/philosophy/open-source-misses-the-point.de.html> (30.10.2014)

Wilkens, Andreas: **Studie übt harte Kritik an Open-Source-Software** In: heise online -
<http://heise.de/-97045> (15.04.2004)

Vaughan-Nichols, Steven J.: **Why Microsoft loves Linux** In: ZDNet -
<http://www.zdnet.com/article/why-microsoft-loves-linux/> (29.10.2014)

Wikipedia: **Supercomputer** - <http://de.wikipedia.org/wiki/Supercomputer#Betriebssysteme>
(13.02.2015)

Wikipedia: **Die Kathedrale und der Basar** -
http://de.wikipedia.org/wiki/Die_Kathedrale_und_der_Basar (31.06.14)