

جامعه‌شناسی منبع باز

در آمدی انتقادی به پدیده منبع باز (open source)

تهیه کننده:

محسن صبوریان

استاد راهنما:

دکتر رامتین خسروی

دانشکده برق و کامپیوتر

پردیس دانشکده‌های فنی، دانشگاه تهران

تابستان ۱۳۸۶

۲	مقدمه	۱
۳	نرم افزار آزاد	۲
۳	مفاهیم	۲،۱
۵	تاریخچه	۲،۲
۷	پروژه گنو	۲،۲،۱
۸	گام های تاریخی منبع باز	۲،۲،۲
۹	مدل توسعه	۲،۳
۹	کلیسای جامع و بازار	۲،۳،۱
۱۲	الگوهای مدل بازار	۲،۳،۲
۱۳	مدیریت پروژه های منبع باز	۲،۳،۳
۱۴	قانون بروکس	۲،۳،۴
۱۴	بررسی پروژه هایی در مدل بازار و کلیسای جامع	۵،۳،۲
۱۹	جمع بندی مدل های ارائه شده	۶،۳،۲
۲۱	مدل کسب و کار	۲،۴
۲۲	جامعه شناسی نرم افزار آزاد	۳
۲۳	جنبه های هویتی و روانی منبع باز	۳،۱
۲۴	کسب و کار	۳،۲
۲۴	نخبه گرایی	۳،۳
۲۵	آیا منبع باز یک انقلاب است؟	۳،۴
۲۷	نتیجه گیری	۴
۲۸	منابع	۵

فراگیر شدن سیستم عامل لینوکس^۱ در میان کاربران عادی کامپیوتر، و مدل توسعه منبع باز^۲ در میان برنامه نویسان، توجه بسیاری را به مفهوم منبع باز و نرم افزار آزاد جلب کرده است. اقبال شرکت های بزرگ تجاری به نرم افزارهای منبع باز و معاونت در تولید آنها باعث شده تا از چندی پیش فعالیت های پژوهشی در غالب مقالات، کنفرانس ها و پیمایش های اینترنتی در این باره صورت بگیرد که نمونه بارز آن انجمن پژوهش نرم افزار منبع باز/آزاد دانشگاه MIT است که در بخش مقالات ارسالی و پژوهش ها بالغ بر ۲۸۰ مقاله، رساله و یا گزارش تحقیقات (از سال ۱۹۹۹ تا هم اکنون) دارد.^۳ شرکت رسانه آرایلی^۴ از سال ۲۰۰۱ تا کنون همساله یکی از مهم ترین کنفرانس های پیرامون منبع باز را تحت عنوان OSCON و با حمایت مالی شرکت های بزرگی چون اینتل، مایکروسافت، دل، سان، آی بی ام و اچ پی برگزار می کند. بسیاری از دانشگاه های مطرح آمریکا و اروپا کنفرانس های منظمی را برای پیگیری تحولات در جنبش منبع باز و نرم افزار آزاد و مدل های توسعه نرم افزار پیرامون آنها برگزار می کنند.

علاقه جامعه مجازی به نرم افزار آزاد و منبع باز نه فقط به خاطر مجانی بودن آن بلکه به دلیل فضای مجازی توسعه آن و همچنین باز بودن محیط تولید آن و در نتیجه امکان مشارکت حداکثری کاربران است. تجربه های موفق چون لینوکس و BSD، فعالان فناوری اطلاعات را برآن داشته تا از منظر خود قسمتی از قضایا و ابهامات فنی، اجتماعی و روانی حول محورهای مختلف توسعه فنی نرم افزار، مدل اقتصادی، جامعه کاربران و ... را روشن کنند.

تحقیقات پیرامون نرم افزار آزاد یک امر کاملاً میان رشته ای و متنوع است. مؤلف، رساله های دانشجویان رشته های مختلفی از مهندسی نرم افزار گرفته تا جامعه شناسی و حقوق و از مدیریت تا روانشناسی و اقتصاد را بررسی کرده است. روانشناسان علاقمند به جنبه های انگیزشی و روانی حول توسعه نرم افزارند؛ جامعه شناسان به جنبه جنبش گونه و یا تغییرات اجتماعی ناشی از استفاده از نرم افزار آزاد نظر دارند، اقتصاددانان به ارائه الگوهای ریاضی - اقتصادی برای اثبات سود دهی و غیرخیرخواهانه بودن توسعه نرم افزارهای آزاد علاقه مندند، مدیران به هنر مدیریت پروژه های منبع باز و حقوقدانان به مسائل حقوقی و مجوزهای استفاده از نرم افزارهای آزاد.

پژوهش موردی^۵ حاضر، با پرداختن به مفهوم نرم افزار آزاد و تاریخچه مختصری از آن، با نگاهی نقادانه به بررسی جنبه ها و مدل های مختلف تولید، مدیریت و کسب و کار حول نرم افزارهای آزاد و منبع باز می پردازد. نمونه هایی از نرم افزارهای مدل آزاد و منبع باز بررسی می شود و فراتر از آن تغییرات بنیادین در تفکر تولید نرم افزار مورد توجه قرار می گیرد. در واپسین فصل این تحقیق، مفاهیمی از حوزه جامعه شناسی برای توضیح وضعیت اجتماعی و روانی منبع باز و علاقه مندان به آن مطرح می شود. بدیهی است که به دلیل قلت تحقیقات انجام شده در این حوزه، مبحث «جامعه شناسی نرم افزار آزاد» صرفاً فتح بابی برای گفتگوی بیشتر در این حوزه بین صاحب نظران آن است.

¹ Linux

² Open Source development model

³ Free / Open source community (online papers), http://opensource.mit.edu/online_papers.php.

⁴ O'Reilly Media Inc.

⁵ case study

۲ نرم‌افزار آزاد

۲،۱ مفاهیم

«نرم‌افزار آزاد» به نرم‌افزاری اطلاق می‌شود که حدودی از آزادی‌ها را به کاربر آن اعطا کند. این آزادی‌ها بسته به تعریف ما از این مفهوم مختلف‌اند، لذا به منظور جلوگیری از ابهامی کلام، مقصودی که معمولاً از استعمال لفظ Free Software مدنظر است را در اینجا می‌آوریم.

واژه‌ی Free Software در مفهوم صحیح، به معنی نرم‌افزار آزاد است نه نرم‌افزار مجانی. ریچارد استالمن^۶ - موسس پروژه‌ی گنو^۷ و بنیاد نرم‌افزار آزاد^۸ - در آوریل ۲۰۰۷ در سخنرانی خود پیرامون GPLv3^۹ در بروکسل گفت:

این مهم است وقتی ما می‌گوییم Free Software، در باره‌ی آزادی صحبت می‌کنیم، نه قیمت. منظور ما نرم‌افزار رایگان نیست. بحث به هیچ وجه درباره‌ی قیمت نیست؛ قیمت صرفاً یک امر جزئی است.

البته این موضوع را او به کرات تکرار کرده و حتی شعار بنیاد نرم‌افزار آزاد نیز همین است^{۱۰}. در واقع دلیل این که گاهی برای نرم‌افزار آزاد و منبع‌باز به جای FOSS، از مخفف FLOSS استفاده می‌شود نیز چنین است. FLOSS مخفف Free/Libre Open Source Software است و واژه‌ی libre در زبان‌های اسپانیایی و فرانسوی به معنی آزاد بودن و آزادی داشتن است و از واژه‌ی لاتین liber گرفته شده^{۱۱}.

استالمن، نرم‌افزار آزاد را اینگونه تعریف می‌کند: «نرم‌افزار آزاد نرم‌افزاری است که آزادی کاربر را محترم بشمارد» و در ادامه می‌آورد: «نرم‌افزاری که در دسترس شماست ولی به آزادی شما احترام نمی‌گذارد، اختصاصی یا غیر آزاد نامیده می‌شود». از نظر استالمن چهار مورد آزادی اساسی که یک نرم‌افزار آزاد باید در اختیار کاربر قرار دهد به قرار زیر است:

۱. کاربر برنامه را هر طور که دوست دارد و برای هر منظوری اجرا کند (آزادی ۰).
۲. کاربر بتواند کد منبع (متن) برنامه را مطالعه کند و آنرا تغییر دهد تا کاری را که می‌خواهد انجام دهد (آزادی ۱). دسترسی به کد منبع، پیش‌نیاز این آزادی است.
۳. کاربر کپی برنامه را آزادانه به هر کس خواست بدهد (آزادی ۲).
۴. کاربر بتواند برنامه را بهبود دهد و نسخه‌ی بهبود یافته‌ی برنامه را آزادانه مجدداً منتشر کند تا جامعه از مزایای آن بهره‌مند شوند. (آزادی ۳). دسترسی به کد منبع، پیش‌نیاز این آزادی است^{۱۲}.

⁶ Richard Stallman

⁷ GNU Project

⁸ Free Software Foundation

⁹ نگارش سوم مجوز جی‌پی‌ال

¹⁰ Free Software is a matter of liberty not price.

¹¹ Wikipedia, the free encyclopedia, Alternative terms for free software, <http://en.wikipedia.org/wiki/FLOSS> (accessed 16 Sep 2007).

¹² Stallman, R., Transcript of Richard Stallman on GPLv3 in Brussels, Belgium; 1st of April 2007. The free software definition: <http://www.gnu.org/philosophy/free-sw.html>.

در کنار مفهوم نرم‌افزار آزاد، «نرم افزار منبع‌باز»^{۱۳} قرار دارد و به نرم‌افزاری گفته می‌شود که تحت مجوزی منتشر می‌شود که به دیگران اجازه می‌دهد تا کد منبع^{۱۴} آن را تغییر دهند، مجدداً منتشر کنند و با دیگران به اشتراک بگذارند^{۱۵}. حوزه‌ی نرم‌افزار منبع‌باز که ذیل عنوان عمومی‌تر «منبع‌باز» قرار می‌گیرد خود شامل مواردی چون «سخت افزار منبع‌باز»، «مستندات و محتوای منبع‌باز»، «آموزش منبع‌باز»، «استاندارد باز»، «سیستم باز» و ... می‌باشد^{۱۶}.

اشتباه عامی که معمولاً صورت می‌گیرد، یکسان انگاشتن دو مفهوم نرم‌افزار «منبع‌باز» و «مجانی» است. هر چند که هر دوی این نرم‌افزارها «آزاد» محسوب می‌شوند، لیکن هیچ رابطه‌ی جزء به کل یا هم‌ارزی بین این دو مفهوم وجود ندارد. بسیاری از نرم‌افزارهای مجانی، منبع‌باز نیستند و تحت یک مجوز اختصاصی منتشر می‌شوند. همچنین نرم‌افزارهای منبع‌بازی هستند که مجانی نیستند و یا نوع استفاده را محدود می‌کنند و یا اساساً برای خرید آن‌ها باید پول پرداخت کرد.

تعریف ارائه‌شده از «نرم‌افزار آزاد» متعلق به «بنیاد نرم‌افزار آزاد» و در رأس آن استالمن است. این بنیاد حق تعریف نرم‌افزار آزاد را برای خود محفوظ می‌داند و علاوه بر آن تأیید می‌کند که آیا یک مجوز، مجوز نرم‌افزار آزاد هست یا خیر.

در مورد استعمال لفظ «نرم‌افزار منبع‌باز» یا «نرم‌افزار آزاد»، در میان پیشگامان نرم‌افزار آزاد اختلاف وجود دارد. واژه‌ی منبع‌باز اساساً به منظور جلوگیری از بدفهمی معنای واژه‌ی free (فهم غلط gratis یا مجانی به جای freedom یا آزادی) در زبان انگلیسی ارائه شد. ریموند دو دلیل برای استفاده از منبع‌باز به جای آزاد بر می‌شمرد: اول اینکه free یک واژه‌ی دویپهلوی و بسیار مبهم است (چیزی که تبلیغات بنیاد نرم‌افزار آزاد باید دائماً برای بیان آن تقلا کند). آیا free به این معناست که «هیچ پولی پرداخت نکنید»، یا اینکه به معنای «آزاد بودن هر کس برای تغییر آن» است^{۱۷}؟ دلیل دوم این است که بسیاری از شرکت‌های صنفی از به کار بردن واژه‌ی free عصبی می‌شوند^{۱۸}. ریموند استدلال می‌کند که ما در حال حاضر علاقه‌ عملگرایانه‌ای به نوکیش^{۱۹} کردن این افراد به جای روی گرداندن از آنها داریم. هم‌اکنون ما شانس بدست آوردن بهره‌ی بالایی از جریان اصلی کسب و کار را داریم بدون اینکه در مورد ایده‌آل‌های خود مصالحه کنیم. پس اکنون زمان عوض کردن موقعیت است. ما یک برچسب جدید و بهتر نیاز داریم.

استالمن با بیان اینکه اکثر نرم‌افزارهای منبع‌باز، نرم‌افزار آزاد هم هستند، معتقد است معانی مختلف عقاید مختلفی را به همراه دارند. وی جنبش منبع‌باز و جنبش نرم‌افزار آزاد را مانند دو اردوگاه سیاسی، درون جامعه‌ی نرم‌افزار آزاد می‌داند:

¹³ OSS: Open Source Software; OS: Open Source; FOSS: Free and Open Source Software; FLOSS: Free/Libre Open Source Software

¹⁴ source code

¹⁵ Mulgen G., et. al., Wide Open: Open source methods and their future potential. 2005, p. 10.

¹⁶ Wikipedia, the free encyclopedia, Open-source software: http://en.wikipedia.org/wiki/Open-source_software, (accessed: 16 Sep 2007).

¹⁷ Raymond, E, Goodbye, "free software"; hello, "open source", 2004

¹⁸ ریموند توضیح نداده به چه علتی عصبی می‌شوند. احتمالاً منظور او این بودن که یک واحد صنفی که به دنبال درآمد است با تولید نرم‌افزار مجانی موافق نیست.

¹⁹ convert

ما در اصول اولیه با هم اختلاف داریم ولی کم و بیش در توصیه‌های عملی^{۲۰} موافق هستیم. ما منبع‌باز را دشمن نمی‌دانیم؛ دشمن ما نرم‌افزار اختصاصی^{۲۱} است.

استالمن تعریف رسمی منبع‌باز توسط پیشگامان منبع‌باز را بسیار نزدیک به تعریف نرم‌افزار آزاد می‌داند؛ با این حال نرم‌افزار منبع‌باز را در برخی جنبه‌ها ضعیف‌تر از نرم‌افزار آزاد می‌داند و بیان می‌کند که آنها برخی گواهی‌نامه‌هایی که به نظر ما بسیار محدود کننده بود را پذیرفته‌اند. استالمن در ادامه مثالی را بیان می‌کند که برای درک بینش و ایدئولوژی او بسیار آموزنده است:

توسعه‌دهندگان نرم‌افزارهای اختصاصی لزوماً افراد بی‌مهارتی نیستند. برخی اوقات آنها نرم‌افزارهایی را تولید می‌کنند که قدرتمند و قابل اطمینان است، هرچند به آزادی کاربر احترام نمی‌گذارد. یک فعال مدافع نرم‌افزار آزاد و یک هوادار نرم‌افزار منبع‌باز چگونه به آن واکنش نشان می‌دهند؟ یک مشتاق مخلص منبع‌باز - که هرگز تحت تأثیر ایده‌ال‌های نرم‌افزار آزاد قرار ندارد - به او می‌گوید: «من غافلگیر شدم؛ شما این برنامه را به خوبی ساخته‌اید بدون اینکه مدل توسعه ما را استفاده کنید؛ در هر حال شما موفق شده‌اید چگونه می‌توانم یک نسخه از نرم‌افزار شما تهیه کنم؟» فعال نرم‌افزار آزاد می‌گوید: «برنامه شما بسیار جذاب است، ولی نه به قیمت آزادی من. به همین دلیل من مجبورم بدون آن ادامه دهم. در عوض من از یک پروژه جایگزین پشتیبانی خواهم کرد.»^{۲۲}

در ادامه ما آزادانه دو مفهوم نرم‌افزار منبع‌باز و آزاد را به جای هم به کار می‌بریم.

۲،۲ تاریخچه

«باز متن» اساساً واژه‌ی بسیار جدیدی است که قدمت آن به اواخر ۱۹۹۷ و اوایل ۱۹۹۸ باز می‌گردد. این واژه اولین بار در جلسه‌ای که اریک ریموند^{۲۳} افراد آن را گرد هم آورده بود، توسط کریستین پیترسن^{۲۴} مطرح شد و مورد قبول قرار دیگران. این جلسه به طور منظم برای تبادل اطلاعات در مورد نرم‌افزارهای پیشرفته‌ای که در مدل غیر اختصاصی^{۲۵} توسعه یافته بودند تشکیل می‌شد. این گروه دامنه‌ی opensource.org را ثبت کردند، واژه منبع‌باز را تعریف کردند، گواهی‌نامه پیشگامان منبع‌باز (OSI)^{۲۶} را تدوین کردند و لیستی از مجوزهایی را که با معیارهای منبع‌باز مطابقت داشتند ارائه دادند.^{۲۷}

هر چند در مورد مبدع واژه Open Source، اختلافاتی وجود دارد^{۲۸}، لکن قدر مسلم این است که همین گروه و پیشگامان منبع‌باز این لفظ و اصطلاح را مرسوم کردند، به طوری که اگر در این زمان از یک فرد آشنا به کامپیوتر درباره مشخصه لینوکس سؤال شود، او پاسخ می‌دهد که لینوکس منبع‌باز است، و بعید است چیزی در مورد نرم‌افزار آزاد شنیده

²⁰ practical recommendations

²¹ proprietary software

²² Why "Open Source" misses the point of "Free Software", 2007

²³ Eric Raymond

²⁴ Christine Peterson

²⁵ non-proprietary

²⁶ Open Source Initiative

²⁷ David Bretthauer, Open Source software: A history.

²⁸ پیشتر واژه open source code در کتاب زیر در سال ۱۹۹۶ استفاده شده بود:

Rosen, K., Rosinski, R., Farber, J., Host, D., UNIX System V Release 4: An Introduction, 2nd Edition, Osborne, 1996.

باشد. همین حقیقت که اکثر افراد تفاوتی بین نرم‌افزار منبع‌باز و آزاد قائل نیستند، کفایت می‌کند که نفوذ مفهوم منبع‌باز مشخص شود.

در اوایل دهه‌ی ۵۰ (و حتی در طول دهه‌های ۶۰ و ۷۰) بیشتر نرم‌افزارها را شرکت‌هایی مانند SHARE^{۲۹} و DECUS^{۳۰} تولید می‌کردند و شرکت‌های تولید سخت‌افزار این نرم‌افزارها را به همراه سخت‌افزارهای تولیدی خود عرضه می‌کردند. در آن زمان فعالیت عمده‌ی شرکت‌های کامپیوتری در حوزه سخت‌افزار بود و هر چیزی که قیمت نرم‌افزارهای موجود برای آن سخت‌افزار را کاهش و تعداد آن‌ها را افزایش می‌داد، شرکت‌های سخت‌افزاری را رقابتی‌تر می‌کرد.

این مدل در دهه‌ی ۶۰ تا اندازه‌ای تغییر کرد. در ۱۹۶۵، ADR اولین نرم‌افزار دارای مجوز را مستقل از سخت‌افزار توسعه داد. ADR در واقع با بسته نرم‌افزاری مجانی که توسط مشتری‌های IBM توسعه داده شده بود رقابت می‌کرد. ADR در ۱۹۶۸ حق انحصاری^{۳۱} نرم‌افزار خود را ثبت کرد. به منظور جلوگیری از به‌اشتراک گذاشتن نرم‌افزار، این شرکت پرداخت قیمت نرم‌افزار را در طول مدت حیات استفاده از نرم‌افزار تقسیم کرد و به این ترتیب ADR مالکیت نرم‌افزار را برای خود حفظ کرد و می‌توانست فروش مجدد و استفاده‌ی مجدد آن را کنترل کند.

در ۱۹۶۹ وزارت دادگستری ایالات متحده، IBM را به تخریب کسب‌وکارها به وسیله‌ی گنجاندن نرم‌افزارهای مجانی به همراه سخت‌افزارهای متهم کرد. به تبع آن IBM دیگر نرم‌افزارها را به همراه سخت‌افزارهایش عرضه نکرد و بدین ترتیب نرم‌افزار از سخت‌افزار جدا شد.^{۳۲}

در طول دهه‌های ۶۰ و ۷۰ عموماً افرادی که نرم‌افزاری را دریافت می‌کردند، این آزادی را داشتند تا آن را برای هر منظوری اجرا کنند: کد منبع را مطالعه کنند، تغییر دهند و یا حتی مجدداً آن را پخش کنند. در این مدت نرم‌افزار به عنوان یک کالا در نظر گرفته نمی‌شد و نرم‌افزارها توسط انجمن‌های^{۳۳} برنامه‌نویسی و محافل دانشگاهی تولید و در اختیار عموم قرار می‌گرفت. به عنوان نمونه نسخه‌های اولیه سیستم‌عامل یونیکس^{۳۴} توسط انجمنی از کاربران نگهداری^{۳۵} می‌شد. به این ترتیب در یک دوره‌ی زمانی، نرم‌افزار نه به دلیل هماهنگی کاربران و توسعه‌دهندگان بلکه بیشتر به دلیل ماهیت توسعه‌ی نرم‌افزار در محیط‌های به‌اشتراک‌گذاری و دانشگاهی، نرم‌افزار آزاد محسوب می‌شد.

بخش‌هایی از نسخه‌های اولیه‌ی یونیکس به دلیل دعوای حقوقی قانون ضد انحصار - که AT&T (صاحب مجوز انتشار یونیکس) را از فروش هر چیزی به جز سرویس‌های تلفنی باز می‌داشت - آزادانه توزیع می‌شد. وکلای AT&T با برداشت بسیار محافظه‌کارانه از این حکم، شرکت را از تجاری کردن نرم‌افزاری که توسعه می‌داد بازداشتند.^{۳۶} در انتهای دهه‌های ۷۰ و ابتدای ۸۰، شرکت‌ها یکی پس از دیگری محدودیت‌هایی را برای استفاده از نرم‌افزارهایشان با اعمال مجوزها وضع کردند. بعضاً به این دلیل که شرکت‌ها از فروش نرم‌افزارهای اختصاصی خود سود می‌بردند و یا تلاش

²⁹ Name of a user group for the IBM 701

³⁰ Digital Equipment Corporation (DEC) Users Group

³¹ Patent

³² Bruce Montague, Very Brief Open Source History, http://www.freebsd.org/doc/en_US.ISO8859-1/articles/bsdl-gpl/history.html

ویکیپدیا، ذیل عنوان History of free software.

³³ Community

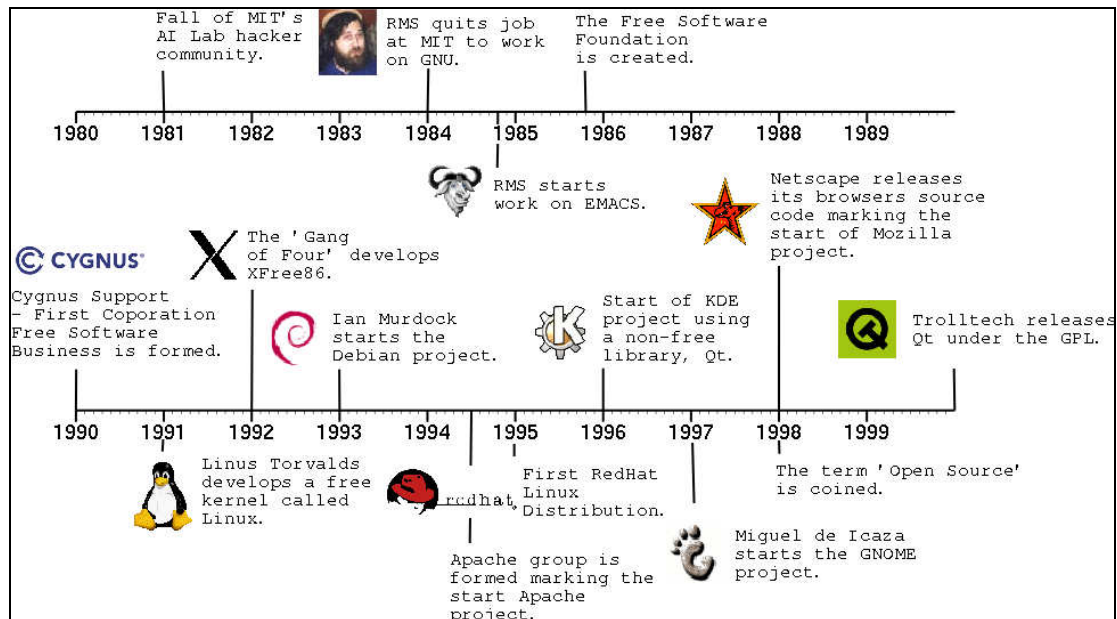
³⁴ UNIX

³⁵ Maintain

³⁶ Weber, S.. The Success of Open Source. 2004.

می‌کردند تا مشخصات سخت‌افزاری را با پنهان کردن کد منبع مخفی نگاه دارند. این وضعیت به مرور به یک عادت تجاری تبدیل شد و حتی شرکت‌هایی که نرم‌افزار خود را مجانی عرضه می‌کردند و سودی از فروش آن عایدشان نمی‌شد، کد منبع آن را مخفی نگاه می‌داشتند. بیل گیتس^{۳۷} این تغییر دوره را با انتشار «نامه سرگشاده به عاشقان سرگرمی»^{۳۸} هشدار داد و بیان کرد آن چیزی که هرکها^{۳۹} آن را به اشتراک گذاری می‌خوانند در کلام او دزدی خوانده می‌شود.^{۴۰}

۲،۲،۱ پروژه گنو



شکل ۱. خط زمانی جنبش نرم‌افزار آزاد (۱۹۸۰ تا ۲۰۰۰)

در ۱۹۸۳ ریچارد استالمن پروژه‌ی گنو^{۴۱} را با هدف ایجاد یک سیستم عامل - بدور از هرگونه محدودیت در استفاده از کد منبع آن - آغاز کرد. او پس از آن اصطلاح «نرم‌افزار آزاد» را ابداع کرد و «بنیاد نرم‌افزار آزاد» را ترویج این مفهوم تأسیس کرد، هر چند تعریف رسمی نرم‌افزار آزاد تا سال ۱۹۸۹ به تعویق افتاد. در همین سال بود که اولین نسخه‌ی مجوز جامع عمومی گنو^{۴۲} منتشر شد. سیستم استالمن سیستم عامل خود را «هرد»^{۴۳} نامید، سیستم عاملی که هرگز تحقق پیدا نکرد، ولی به هر حال نرم‌افزارهای پروژه‌ی گنو - به خصوص مجموعه کامپایلرهای گنو^{۴۴} - جایگزین مناسبی برای نمونه‌ی قبل خود بر روی سیستم‌های یونیکس بودند.^{۴۵}

³⁷ Bill Gates

³⁸ Open Letter to Hobbyists

³⁹ Hacker

⁴⁰ ویکی‌پدیا، ذیل عنوان History of free software، با کمی تصرف.

⁴¹ GNU Project

⁴² GNU General Public License (GPL)

⁴³ The Hurd

⁴⁴ GNU Compiler Collection (GCC)

⁴⁵ همان.

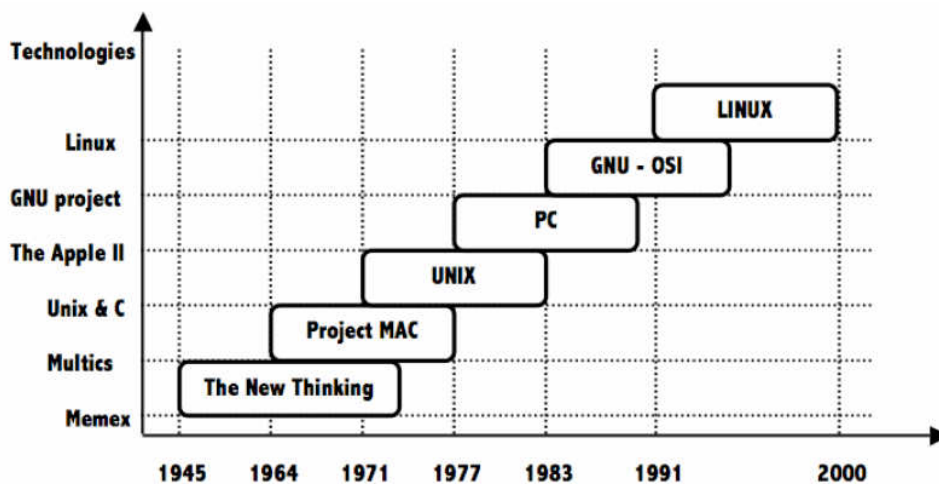
در ۱۹۹۲، اولین نسخه‌ی هسته‌ی سیستم عاملی که لینوس توروالدز^{۴۶} یک سال قبل توسعه‌ آن را شروع کرده بود، به عنوان نرم‌افزار آزاد و تحت عنوان «لینوکس»^{۴۷} منتشر شد. در آن زمان پروژه‌ گنو اکثر نرم‌افزارهای لازم برای یک سیستم عامل به جز هسته‌ آن را تولید کرده بود. ترکیب نرم‌افزارهای پروژه‌ی گنو و هسته‌ لینوکس از ۱۹۸۳ تا کنون سیستم عاملی را می‌سازد که به لینوکس شهرت دارد، نامی که استالمن همواره تأکید دارد باید «گنو لینوکس» خوانده شود^{۴۸}. این تأکید، هرچند هنگامی که به سیستم عامل به عنوان یک گُل که بر روی هسته‌ لینوکس بنا شده است، قابل قبول است، لکن در صورتی که منظور از لینوکس، همان هسته‌ سیستم عامل باشد، محلی از اعراب ندارد.

پروژه‌های بزرگ دیگری نیز در پیدایش و گسترش نرم‌افزار آزاد نقش داشته‌اند که به لزوماً مانند پروژه‌ گنو ایدئولوژیک نبوده‌اند. یک نمونه‌ مشهور، توزیع نرم‌افزار برکلی^{۴۹} (BSD) است. سیستم عامل یونیکسی که به تدریج در دهه‌ ۷۰ در دانشگاه کالیفرنیا در برکلی توسعه یافت^{۵۰}.

۲.۲.۲ گام‌های تاریخی منبع‌باز

بونیسی در رساله‌ در حال تکمیل خود ۶ گام مختلف در روند توسعه‌ تاریخی منبع‌باز ذکر کرده است و جالب اینجاست که او روند آغازین توسعه‌ منبع‌باز را از اواخر دهه‌ ۴۰ بر می‌شمرد. او این ۶ مرحله را هم از دیدگاه اقتصادی و هم فناوری تقسیم بندی کرده است.

شش دوره‌ای که وی بر می‌شمرد عبارتند از: (۱) روش جدید/اندیشیدن (۱۹۴۵ تا ۱۹۶۹)، (۲) پروژه‌ بزرگ MAC (۱۹۶۳ تا ۱۹۷۵)، (۳) زبان (۱۹۷۱ تا ۱۹۸۲)، (۴) تولد PC (۱۹۷۷ تا ۱۹۹۱)، (۵) پروژه‌ گنو و پیشگامان منبع‌باز (۱۹۸۳ تا ۱۹۹۸) و (۶) عصر لینوکس (۱۹۹۱ تا کنون). خلاصه‌ این روند را در شکل ۱ می‌توان دید^{۵۱}.



⁴⁶ Linus Torvalds

⁴⁷ Linux

⁴⁸ همان، همچنین ن.ک. <http://www.gnu.org/gnu/why-gnu-linux.html>

⁴⁹ Berkeley Software Distribution

⁵⁰ Fogel, K., Producing Open Source software, 2005

⁵¹ Benussi, L., Analysing the technological history of the Open Source phenomenon. Stories from the Free Software evolution, 2005. pp. 6 - 8.

به طور خلاصه، گام اول ناظر به دوران پس از جنگ جهانی دوم و آغاز جنگ سرد است. ماشین میمیکس^{۵۲} و پروژه آرپانت^{۵۳} - پدر اینترنت - از محصولات این دوره هستند. گام دوم مربوط به دوره توسعه سیستم‌های اشتراک زمانی و سیستم‌های برخط است. گام سوم به دوره ظهور سیستم عامل یونیکس، زبان برنامه‌سازی سی، و فرهنگ هکرها دلالت می‌کند. گام چهارم عبارتست از فراگیر شدن رایانه‌های شخصی، رابط‌های گرافیکی و پیدایی نرم‌افزارهای اختصاصی و به تبع آن ایجاد انگیزش برای تولد نرم‌افزار آزاد و منبع‌باز. گام پنجم اشاره به دوره‌ای از پیدایش پروژه گنو تا استفاده از لفظ منبع‌باز دارد و نهایتاً گام ششم و آخرین قطعه این دوره یعنی عصر لینوکس^{۵۴}.

۲،۳ مدل توسعه

۲،۳،۱ کلیسای جامع و بازار

منظور از مدل یا الگوی توسعه^{۵۵}، اصول و عرف‌های نانوشته‌ای است که بعضاً در توسعه نرم‌افزارهای آزاد به کار می‌رود. یکی از عمده کارهای الهام‌بخش پیرامون مدل توسعه در جامعه نرم‌افزار آزاد، کتاب «کلیسای جامع و بازار» ریموند است^{۵۶}. این کتاب مجموعه‌ای از مقالات ریموند پیرامون مسائل مربوط به مدل توسعه منبع‌باز است. مهم‌ترین مقاله کتاب که هم‌نام کتاب است، در مورد مدل توسعه پروژه fetchmail نویسنده است. محتوای این مقاله، بررسی روند توسعه برنامه‌ای برای دریافت و ذخیره ایمیل‌ها بر روی کامپیوتر شخصی، با الهام از مدل توسعه آزاد لینوکس است. هرچند پروژه ریموند از نظر اندازه، شهرت و تعداد کاربران و مشارکت‌کنندگان، به هیچ وجه با لینوکس قابل مقایسه نیست، لکن توصیه‌ها و نکاتی که نویسنده در خلال مباحث مطرح می‌کند در درک مدل‌سازی او آموزنده است.

ریموند اشاره می‌کند که برای آزمون نظریه خود درباره کاری که لینوس توروالدز انجام داده، بر روی این پروژه کار می‌کند و بالطبع از توصیه‌های این مدل تبعیت می‌کند:

زود به زود و پیش از موعد برنامه را منتشر می‌کردم (تقریباً هر ۱۰ روز یک نسخه و در دوره توسعه شدید، هر روز یک نسخه). لیست آزمونگرهای بتای^{۵۷} برنامه را با اضافه کردن هر کسی که درباره fetchmail با من تماس گرفته بود، بزرگ می‌کردم. آگهی‌های ویراژانه‌ای را هنگام انتشار هر نسخه به لیست [آزمونگرهای] بتا می‌فرستادم و در آن افراد را تشویق به مشارکت می‌کردم. به حرف‌های آزمونگرهای بتای [برنامه] خود گوش می‌دادم، از آنها در مورد تصمیم‌گیری‌های طراحی نظرخواهی می‌کردم و هر وقت وصله^{۵۸} یا بازخوردی می‌فرستادند، تشکر می‌کردم^{۵۹}.

تأکید ریموند در این مقاله همین جمله است که او بارها به انحاء مختلف تکرار می‌کند: «هر کار خوبی در نرم‌افزار با خاراندن خارش شخصی توسعه‌دهنده آغاز می‌شود». منظور او از این جمله نسبتاً روشن است: برنامه‌نویسان و

⁵² Memex machine

⁵³ ARPANET

⁵⁴ همان منبع

⁵⁵ development model

⁵⁶ Raymond, E., The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary, Revised Edition, 2001.

⁵⁷ beta tester

⁵⁸ patch

⁵⁹ همان منبع، ص. ۳۸.

توسعه‌دهندگان منبع‌باز، آدم‌های نوع‌دوست و فداکاری نیستند که وقت خودشان را برای جلب رضایت دیگران تلف کنند. آنها - لاف‌ل در درجه نخست - به فکر ارضای نیاز خود هستند:

جهان لینوکس در بسیاری از جنبه‌ها همانند بازار آزاد یا بوم‌شناسی - مجموعه‌ای از عامل‌های خودخواه که برای بیشینه کردن منفعت خود تلاش می‌کنند - است و در فرایند، یک نظم خود اصلاح‌گر و خود انگیزه تولید می‌کند، که بسیار ماهرانه‌تر و کارا تر از چیزی است که هر تصمیم‌گیری مرکزی به آن نائل می‌شود.^{۶۰}

«کارکرد منفعت» که هرکدام لینوکس در پی حداکثر کردن آن هستند، از نظر کلاسیک اقتصادی نیست، بلکه [عامل] نامرئی، ارضاء «نفس»^{۶۱} آنها و شهرتشان در میان سایر هرکها است. بعضی‌ها ممکن است انگیزه آنها را نوع‌دوستانه پندارند، ولی این پندار حقیقت این مطلب را که نوع‌دوستی هم نوعی ارضاء نفس نوع‌دوست است، نادیده می‌گیرد.^{۶۲}

... هر دوی fetchmail و هسته لینوکس نشان دادند که با پاداش‌دادن مناسب به نفس بسیاری از هرکهای دیگر، یک راهبر/توسعه‌دهنده می‌تواند از اینترنت استفاده کند تا منفعت کمک توسعه‌دهندگان فراوانی را جلب کند بدون اینکه پروژه در یک شلوغی پر هرج و مرج فروریزد.^{۶۳}

در خلال توضیحات و توصیه‌های خود پیرامون روش توسعه مدل بازار آزاد، ریموند به نکاتی اشاره می‌کند علاوه بر اینکه زیربنای مباحث او را مطرح می‌کند، مورد انتقاد قرار گرفته است. از جمله منتقدان دائمی مباحث ریموند، دکتر نیکولای بزروکوف^{۶۴} است که نه تنها انتقاداتی را به مقالات (و حتی خصوصیات شخصی ریموند) مطرح می‌کند، بلکه یک تنه به جنگ تمامی سران منبع‌باز و نرم‌افزار آزاد رفته است.^{۶۵}

از جمله اشارات جنجالی ریموند این جمله است که «با داشتن تعداد کافی چشم، همه باگ‌ها سطحی هستند». ریموند این گونه به بسط این ادعا پرداخته است:

با داشتن تعداد زیادی آزمونگر بتا و مبنای کمک توسعه‌دهنده، تقریباً تمامی مشکلات خیلی سریع مشخص و راه حل آن برای کسی روشن می‌شود.

... در دیدگاه معماری کلیسای جامع، باگ‌ها و مشکلات توسعه نیرنگ‌آمیز، خائنه و پدیده‌ای عمیق هستند. ماه‌ها بررسی دقیق اختصاصی برای توسعه اطمینان توسط آن تعداد محدودی که قبلاً آنها را بیرون انداخته‌اید لازم

⁶⁰ همان منبع، ص. ۵۲.

⁶¹ ego

⁶² همان منبع، ص. ۵۳.

⁶³ همان منبع؛ ص. ۵۴.

⁶⁴ Nikolai Bezroukov

⁶⁵ مباحث بزروکوف از وبسایت او که برای ارائه خدمات به برنامه توسعه پایدار شبکه سازمان ملل ساخته شده است، قابل مشاهده است: <http://www.softpanorama.org>

است.^{۶۶} به این ترتیب وقفه‌های طولانی انتشار را به علاوه ناامیدی اجتناب ناپذیر در اثر این وقفه طولانی در صورتی که برنامه منتشر شده متکامل نباشد خواهیم داشت.

در دیدگاه بازار از طرف دیگر، شما فرض می‌کنید که عموماً باگ‌ها پدیده‌هایی سطحی هستند؛ یا دست‌کم با نمایاندن آن به کمک توسعه‌دهندگانی که تک‌تک نسخه‌های منتشر شده را آزمایش می‌کنند، خیلی سریع سطحی می‌شوند. در نتیجه، شما به کرات منتشر می‌کنید تا اصلاحات بیشتری را دریافت کنید و به عنوان یک اثر جانبی سودمند، اگر سر و کله یک مشکل اتفاقی پیدا شود، شما کمتر چیزی را از دست می‌دهید.^{۶۷}

بزروکوف در جواب این ادعا اشاره می‌کند:

... برای لینوکس این (رفع باگ‌های طراحی) مشکلی نبوده چرا که لینوکس، همزاد^{۶۸} یونیکس است و بسیاری از مسائل طراحی آن پیشتر حل شده. ولی در حالت کلی اگر باگ به دلیل مشکلی در طراحی باشد، می‌تواند به شدت بیشتر برای پیدا کردن و اصلاح - در فاز توسعه - زمان ببرد. محول کردن اشکالیابی برای مدیر پروژه مناسب است، ولی شرایطی که در آن تعدادی دوجین توسعه‌دهنده با استعداد وقتشان را در تلاش برای پیدا کردن یک باگ مشترک بدون هماهنگی تلف می‌کنند، چندان کارا نیست.^{۶۹}

باگ‌های مرتبط با معماری فارغ از تعداد کمک‌توسعه‌دهندگان مشتاق، به هیچ عنوان سطحی نیستند. در اینجا لینوکس به این دلیل که همزاد یونیکس بوده تا اندازه‌ای نجات پیدا کرده، و بنابر این مشکلات معماری کمی در آن موجود است (امنیت یکی از آن مشکلات است. این مورد در طراحی یونیکس از اولویت بالایی برخوردار نبوده).

بزروکوف با مقایسه مدل گسترده آزمون بتای شرکت مایکروسافت که در آن کد منبع در اختیار آزمونگرها قرار نمی‌گیرد، با آنچه که ریموند آن را بازار می‌خواند این سؤال را مطرح می‌کند که آیا مایکروسافت یک سازمان مبتنی بر بازار آزاد غایی است؟

وی در برابر این ادعای ریموند که مدل باز و آزاد توسعه لینوکس را کاملاً مخالف معماری کلیسای جامع می‌داند، مدل توسعه لینوکس را یک مدل هوشمندانه مخفی کلیسا می‌داند که لینوس توروالدز دیکتاتور آن است.^{۷۰}

بزروکوف در مقاله انتقادی مشهوری که در مجله اینترنتی دوشنبه اول^{۷۱} - مدتی پس از انتشار مقاله کلیسای جامع و بازار ریموند منتشر کرد - انتقادات خود را از منظر یک استاد دانشگاه، در چند جهت متوجه مقاله ریموند کرده

^{۶۶} ظاهراً اشاره ریموند به این مطلب است که توسعه‌دهندگان در مدل کلیسای جامع، پس از پایان یا در خلال پروژه ممکن است آن را ترک کنند.

^{۶۷} Raymond, E., The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary, Revised Edition, 2001, pp. 8-9.

^{۶۸} clone

^{۶۹} Bezroukov, N., (Slightly skeptical) annotated collection of quotes from Eric Raymond's CatB book, 2000

^{۷۰} مدل توسعه لینوکس از بسیاری از پروژه‌های آزاد و منبع‌باز نظیر GCC، PostgreSQL یا سرور وب آپاچی بسته‌تر است، چرا که تمامی تغییرات با نظارت کامل لینوس توروالدز اعمال می‌شود. اتفاقاً برخی معتقدند همین اعمال قدرت یک‌جانبه و مدل شبه-کلیسا باعث موفقیت و یکپارچگی لینوکس است، نه آنچه ریموند آن را بازار آزاد می‌نامد.

^{۷۱} First Monday, peer reviewed journal on the Internet

است. وی ادعای ریموند مبنی بر این که قانون بروکس به پروژه‌های اینترنتی مربوط نمی‌شود را رد می‌کند.^{۷۲} در ادامه مطالب، در فصل «قانون بروکس» پس از توضیح قانون و رد ریموند بر این قانون، اشکال بزروکوف را مطرح می‌کنیم.

وی در همان مقاله انتقاداتی را متوجه این شعار ریموند می‌سازد که «با داشتن تعداد کافی چشم، همه باگ‌ها سطحی هستند»:

اشکال‌زدایی یک برنامه پیچیده بسیار مشکل‌تر از آن است که بتوان به سادگی با داشتن تعداد زیادی کمک-توسعه دهنده مشتاق برای تجزیه و تحلیل کد، از عهده آن بر آمد. برای اکثر پروژه‌های پیچیده، به ازای هر باگ دوم و سومی که اصلاح می‌شود، باگ جدیدی شناسایی می‌شود.

[کتاب] کلیسای جامع و بازار فرض می‌کند که تعدادی توسعه‌دهنده‌ی با استعداد می‌توانند روی قطعه کد واحدی به موازات کار کنند بدون هیچ‌گونه هماهنگی به جز ایمیل. نهایتاً یکی از آنها کد را اصلاح می‌کند سریع‌تر از محیط تجاری با آزمون‌گرهای کارآموده.

۲،۳،۲ الگوهای مدل بازار

روبلز اهم الگوهای مدل بازار را به این صورت ذکر کرده است:^{۷۳}

۵. با کاربران مثل همکار توسعه‌دهنده رفتار کنید: روند توسعه را در بیشترین حد باز نگهدارید تا توسعه‌دهندگانی که به پروژه علاقمندند به آسانی با آن هماهنگ شوند.
۶. زود منتشر کنید: نسخه اول نرم‌افزار - هرچند از نظر امکانات محدود باشد - باید به محض فراهم بودن چیزی برای ارائه، منتشر شود.
۷. یکپارچه‌سازی مکرر (دائماً منتشر کنید): برداشتن قدم‌های کوتاه به پروژه این امکان را می‌دهد که در یک مسیر افزایشی رشد کند، و در همان زمان اجازه اشکال‌زدایی و بازخوردهای مکرر از طرف کمک توسعه‌دهندگان را فراهم می‌کند.
۸. نسخه‌های متعددی را نگهداری کنید: این هدف برای این است که نرم‌افزار کمتر آزمایش شده و دارای مشکل را به کاربران محافظه‌کار تحمیل نکنیم. بنابراین دو نسخه هم‌زیست در کنار هم وجود دارند؛ یکی برای توسعه (ناپایدار) با آخرین امکانات نرم‌افزار و دیگری نسخه محصول (پایدار) برای آنهایی که تأکید بر پایداری نرم‌افزار به جای امکانات بیشتر آن دارند.
۹. تا حد ممکن برنامه را بسته‌بندی^{۷۴} کنید: بسته‌بندی زیاد امکان توسعه موازی و استفاده مجدد از کد را می‌دهد. متأسفانه تولید کد به اندازه استقرار و رفع اشکال آن قابلیت موازی شدن ندارد.
۱۰. ساختار پویای تصمیم‌گیری: علی‌رغم اینکه عموماً تصور می‌شود که تصمیم‌نهایی را توسعه‌دهنده‌ای که برنامه را کد می‌کند می‌گیرد، معمولاً یک ساختار سازمانی وجود دارد که گاهی رسمی و گاهی غیر رسمی است. صورت سازمانی وابسته همچنان که به وجوه تکنیکی پروژه (اندازه پروژه، تعداد توسعه‌دهندگان و ...)

⁷² Bezroukov, N., A second look at the cathedral and bazaar, First Monday, 1999.

⁷³ نقل با تلخیص از:

Robles, G., A Software Engineering approach to Libre Software, Open-Source-Jahrbuch 2004, Berlin (Lehmanns Media) 2004.

⁷⁴ Modularize

وابسته است، به وجوه دیگر (علل تاریخی، تعادل بین علائق و ...) نیز وابسته است. به عنوان مثال لینوکس مبتنی است بر یک مدل خودکامگی خیرخواهانه: لینوس توروالدز دیکتاتوری است که برخی وظایف نظارتی را به نوآینس محول می‌کند؛ حال آنکه پروژه‌های دیگر مبتنی بر یک شایسته‌سالاری دموکراتیک (آپاچی، گنوم و ...) و یا کمیته‌ای راهبردی است (GCC).

۲،۳،۳ مدیریت پروژه‌های منبع‌باز

کارل فاگل^{۷۵} - توسعه‌دهنده اصلی نرم‌افزار SVN^{۷۶} - کتاب خود را با این جمله آغاز می‌کند: «بیشتر پروژه‌های نرم‌افزار آزاد شکست می‌خورند.»^{۷۷} وی در ادامه استدلال می‌کند که فقط پروژه‌های موفق جلب توجه می‌کنند و به‌علاوه شکست (بر خلاف موفقیت) یک واقعه به حساب نمی‌آید. فاگل با بررسی شواهد از تجربه‌ی بیش از یک دهه خود در عرصه نرم‌افزار منبع‌باز و جستجو بر روی سایت سورس‌فورج^{۷۸} این احتمال را مطرح می‌کند که ۹۰ تا ۹۵ درصد از پروژه‌های منبع‌باز شکست می‌خورند. او اضافه می‌کند که اگر پروژه‌هایی که به درستی و با قابلیت اطمینان بالا کار نمی‌کنند را هم اضافه کنیم، این آمار باز هم افزایش می‌یابد.^{۷۹}

در توسعه پروژه‌های منبع‌باز، مدیریت پروژه - اگر در درجه اول اهمیت نباشد - از اهمیت بالایی برخوردار است:

... پس از آن به این استدلال غلط می‌رسیم که برای منبع‌باز، مدیریت پروژه لازم نیست، یا خیلی کم لازم است؛ یا به عکس، شیوه‌هایی مدیریتی که برای توسعه داخلی سازمانی استفاده می‌شود، به همان اندازه به خوبی برای یک پروژه منبع‌باز کار می‌کند. مدیریت در یک پروژه منبع‌باز همواره هویدا نیست، ولی در پروژه‌های موفق، این [مدیریت] معمولاً در پشت صحنه به صورتی اتفاق می‌افتد.

فاگل برای اثبات این ادعا استدلال می‌کند که یک پروژه منبع‌باز از مجموعه‌ای تصادفی از برنامه‌نویسان تشکیل شده که پیش از آن به طور انگشت‌نمایی در رده‌های ذهنی مستقل از هم بودند. به احتمال زیاد هیچ‌گاه همدیگر را ندیده‌اند و ممکن است هر کدام از آنها هدف شخصی متفاوتی را از کار بر روی پروژه داشته باشد. با همین ارزیابی فاگل معتقد است بدون مدیریت، هیچ چیز جز معجزه نمی‌تواند یک پروژه منبع‌باز را سرپا نگه دارد. فاگل اضافه می‌کند: «تنها چیزی که یک گروه توسعه دهنده را کنار هم نگه می‌دارد، این اعتقاد مشترک است که آنها می‌توانند با هم کار بیشتری انجام دهند تا به صورت منفرد.» لذا هدف عمده مدیریت این است که با وضع استانداردهای ارتباطی، این اعتقاد مشترک را تعمیق بخشد، مراقب باشد که توسعه دهندگان به خاطر خصوصیات فردی حاشیه نشین نشوند و به طور کلی پروژه را به مکانی تبدیل کند که توسعه دهندگان دوست دارند در آن بمانند.

پیرامون نظر کلی فاگل درباره شکست بیش از ۹۰ درصد از پروژه‌های منبع‌باز، توضیح یک نکته لازم است. غالب پروژه‌های منبع‌باز به صورت فردی توسعه داده می‌شوند و معمولاً پیش از اینکه استفاده از آنها شایع شود، ذی‌نفعی ندارند. منظور فاگل از شکست یک پروژه بدون ذینفع که به صورت فردی توسعه داده می‌شود، قطعاً نمی‌تواند

⁷⁵ توسعه‌دهنده اصلی نرم‌افزار کنترل نسخه ساب‌ورژن (Subversion)

⁷⁶ Subversion

⁷⁷ Fagel, K., Producing Open Source software, 2005

⁷⁸ <http://sourceforge.net>

⁷⁹ همان.

ورشکستگی یا خسارت مالی صاحب پروژه باشد، چرا که پروژه‌های شخصی غالباً با دلایل شخصی توسعه داده می‌شوند. منظور فاگل از شکست پروژه به احتمال زیاد در درجه نخست توقف پروژه پیش از دستیابی به چشم‌انداز ترسیم شده توسعه آن و یا عدم توانایی در جذب کاربران و نهایتاً مغفول ماندن از پروژه است. این موضوع با نگاهی به لیست پروژه‌های سایتی مانند سورس‌فورج، به سادگی قابل مشاهده است. پروژه‌هایی که مدت زیادی از ثبت آنها می‌گذرد، حتی یک فایل آماده برای دانلود ندارند. پروژه‌های دیگری پیش از رسیدن به یک محصول استوار و در همان نسخه‌های اولیه ۰/۱ متوقف شده‌اند و موارد دیگری که اساساً فایل‌های آنها قابل ساختن و یا اجرا نیست. با نگاهی به آمار میزان دانلود هر پروژه می‌توان به مرگ آن پروژه پی برد. پروژه‌هایی که از ۱ سال (یا پیش از آن) تا کنون کمتر از ۱۰۰ بارگذاری^{۸۰} داشته‌اند؛ پروژه‌هایی که میلینگ لیست^{۸۱} ندارند؛ پروژه‌هایی میلینگ لیست دارند ولی پاسخی به سؤال شما در لیست آن داده نمی‌شود؛ تمامی این موارد علائم مرگ یک پروژه است.

۲،۳،۴ قانون بروکس

یکی از آثار کلاسیک و ماندنی در حوزه مهندسی نرم‌افزار و مدیریت پروژه‌های نرم‌افزاری، کتاب «نفر-ماه افسانه‌ای»^{۸۲} فردریک بروکس است. بروکس در این کتاب ادعایی را در مورد تعداد افراد درگیر در یک پروژه نرم‌افزاری مطرح می‌کند که به قانون بروکس^{۸۳} مشهور شده است. بروکس مدعی است که با افزایش خطی افراد یک پروژه، به دلیل برقراری ارتباطات متقابل بین افراد پروژه، پیچیدگی (ارتباطی) پروژه به صورت نمایی افزایش می‌یابد. در واقع ادعای بروکس مبتنی بر صورت‌بندی ساده زیر است که تعداد ارتباطات دوسویه بین افراد یک پروژه (n) را بدست می‌دهد:

$$complexity = \frac{n \times (n-1)}{2}$$

صورت‌بندی فوق با این شرط صحیح است که بپذیریم در یک پروژه، همه افراد دخیل با هم اندرکنش دارند^{۸۴} و در واقع ما یک گراف کامل از ارتباطات داریم.

ریموند معتقد است در یک پروژه منبع‌باز، توسعه‌دهندگان هاله‌ای^{۸۵} به صورت نسبتاً موازی با سایر توسعه‌دهندگان کار را پیش می‌برند و نیاز بسیار کمی به ارتباطات متقابل دارند. منظور ریموند از توسعه‌دهندگان هاله‌ای، افراد غیر دخیل در توسعه مرکزی نرم‌افزار است. افرادی که کد منبع را مرور می‌کنند و باگ‌های آن و در صورت لزوم وصله‌های اصلاحی آن را ارائه می‌کنند^{۸۶}. به نظر ریموند ارائه راه حل‌هایی در سطح کد منبع به توسعه‌دهندگان نرم‌افزار ارزش بسیار زیادی در کاهش هزینه تولید نرم‌افزار دارد.

۲،۳،۵ بررسی پروژه‌هایی در مدل بازار و کلیسای جامع

⁸⁰ download

⁸¹ mailing list

⁸² Brooks, Frederik P., The mythical man-month: essays on software engineering, 20th anniversary ed., 1995.

⁸³ Brooks law

⁸⁴ به نظر می‌رسد منظور اصلی بروکس تیم توسعه و تولید نرم‌افزار باشد.

⁸⁵ halo developer

⁸⁶ Raymond, E., The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary, Revised Edition, 2001. pp. 34 - 35.

بر طبق کتاب ریموند^{۸۷} می‌توان مدل توسعه پروژه‌های نرم‌افزاری حوزه منبع‌باز را به دو دسته بازار و کلیسای جامع تقسیم‌بندی کرد. هر چند کمتر پروژه منبع‌بازی را می‌توان یافت که بتوان آن را به دقت در یکی از این دو مدل گنجانند و اکثر پروژه‌های بزرگ منبع‌باز رگه‌هایی از هر کدام از مدل‌های توسعه دارند، لکن از روی شباهت هر پروژه به هر یک از این انواع ایده‌ال می‌توان آن پروژه را تقسیم‌بندی کرد.

در مدل بازار، توسعه پروژه مبتنی بر یک آزادی کلی (ملهم از تئوری بازار آزاد) است که هر کسی می‌تواند آزادانه در روند توسعه پروژه مشارکت داشته باشد. افراد مختلف امکانات مورد علاقه خود را به نرم‌افزار اضافه می‌کنند، باگ‌های نرم‌افزار را گزارش می‌دهند، در بحث‌های توسعه در میلینگ لیست‌ها مشارکت می‌کنند و به این ترتیب هسته مرکزی توسعه نرم‌افزار را تشکیل می‌دهند. روند و برنامه توسعه پروژه با توافق افراد دخیل در پروژه (بسته به میزان مشارکت آنها) تعیین می‌شود. نمونه این روش توسعه را می‌توان در پروژه‌های موفق چون سرور وب آپاچی، پایگاه داده PostgreSQL، و هسته سیستم عامل لینوکس مشاهده کرد. از دیگر موارد موفق این روش توسعه - در حوزه‌های غیر نرم‌افزاری - می‌توان به دانشنامه آزاد ویکی‌پدیا اشاره کرد.

الف) ویکی‌پدیا (مدل بازار) - دانشنامه آزاد ویکی‌پدیا یک نمونه بسیار موفق از پروژه‌های مدل بازار است که از دو رکن ویکی و دایرةالمعارف تشکیل شده. ویکی که واژه‌ای مربوط به اهالی هاوایی است، به معنی سریع و چابک است و در اصطلاح (از سال ۱۹۹۵ به بعد) به وب‌سایت یا برنامه تحت وبی اطلاق می‌شود که علاوه بر اینکه به کاربر امکان مشاهده آن را می‌دهد، امکان ویرایش و حتی اضافه کردن صفحات جدید را هم به سادگی برای او مهیا می‌کند.^{۸۸} رکن دیگر ویکی‌پدیا ماهیت دایرةالمعارف گونه آن است. جیمی ویلز^{۸۹} (یکی از مؤسسين ویکی‌پدیا) چشم‌انداز ویکی‌پدیا را چنین ترسیم می‌کند:

جهانی که در آن به هر فرد دسترسی آزاد به مجموعه‌ای از تمام دانش‌های انسانی داده شود. این همان چیزی است که ما انجام می‌دهیم.^{۹۰}

ویکی‌پدیا^{۹۱} یک پروژه چند زبانه مبتنی بر وب است که توسط بنیاد ویکی‌مدیا^{۹۲} راهبری می‌شود. در سپتامبر ۲۰۰۷، ویکی‌پدیا بیش از ۸ میلیون مقاله در ۲۵۳ زبان مختلف داشته است.^{۹۳} تعداد مقالات انگلیسی این دانشنامه در تاریخ ۹ دسامبر ۲۰۰۷ از مرز دو میلیون مقاله گذشت.^{۹۴} مقالات ویکی‌پدیا به صورت مشارکتی توسط داوطلبانی از سراسر جهان نوشته می‌شود و در حال حاضر این سایت در بین ۱۰ سایت اول از نظر تعداد مراجعه کننده قرار دارد.

ویکی‌پدیا دانشنامه‌ای است که به طور کامل توسط داوطلبان و مشارکت‌کنندگان نوشته و نگهداری می‌شود. این دانشنامه که در سال ۲۰۰۱ تأسیس شد، خیلی سریع در میان ۱۵ سایت رایج و محبوب دنیا قرار گرفت. ویکی‌پدیا یک

⁸⁷ همان منبع

⁸⁸ Schroer, J., & Hertel, G., , Voluntary engagement in an open web-based encyclopedia: wikipedians, and why they do it, 2007, pp. 4.

⁸⁹ Jimmy Wales

⁹⁰ همان منبع.

⁹¹ Wikipedia

⁹² Wikimedia

⁹³ Wikipedia: <http://en.wikipedia.org/wiki/Wikipedia>

⁹⁴ همان منبع

دانشنامه چند زبانه است و بیش از ۱۵۴ میلیون بازدید کننده در ماه دارد^{۹۵}. در سال ۲۰۰۵ این سایت در ۲۰۳ زبان مختلف قابل دسترسی بوده است^{۹۶}.

در یکی از پیمایش‌های اخیر، از ۳۷۰ نفر از کاربران فعال ویکی‌پدیای انگلیسی خواسته شده تا نظر خود را در مورد انگیزه‌ای که برای مشارکت دارند بیان کنند. ۱۵۱ پاسخ معتبر (نرخ پاسخ ۴۰/۸٪) از این افراد دریافت شد که ۱۴۰ مورد آن متعلق به مردان بود (۹۲/۷٪). ساختار پرسشنامه مبتنی بر طیف لیکرت^{۹۷} ولی صرفاً با پاسخ‌های مثبت است. در جدول زیر پرسش‌های این پیمایش آورده شده است^{۹۸}.

انگیزه	نمونه سؤال
حمایتی	با نوشتن/ویرایش کردن در ویکی‌پدیا من کمتر احساس تنهایی می‌کنم.
ارزش‌ها	احساس می‌کنم این مهم است که به دیگران کمک کنم.
شغل	من می‌توانم ارتباطات جدیدی برقرار کنم که می‌تواند به موقعیت تجاری یا شغلی من کمک کند.
اجتماعی	افرادی که من به آنها نزدیک هستم، از من می‌خواهند تا در ویکی‌پدیا بنویسم/ویرایش کنم.
درک	نوشتن/ویرایش کردن در ویکی‌پدیا به من امکان می‌دهد تا چشم‌انداز تازه‌ای در مسائل بدست آورم.
بهبود	نوشتن/ویرایش کردن در ویکی‌پدیا به من کمک می‌کند تا احساس کنم به من نیاز دارند.
تفریح	نوشتن/ویرایش کردن در ویکی‌پدیا یک تفریح است.
ایدئولوژی	من فکر می‌کنم اطلاعات باید آزاد باشد.

جدول ۱-۲. انگیزه‌های نویسندگان/ویرایش کنندگان ویکی‌پدیا

متوسط زمانی که کاربران بر روی ویرایش و یا نوشتن در ویکی‌پدیا صرف می‌کردند، ۸/۲۷ ساعت در هفته بود. روی هم رفته عمده‌ترین انگیزه، دو مورد آخر یعنی تفریح و ایدئولوژی بوده است. لکن همبستگی قابل توجهی میان میزان مشارکت و موارد تفریح و ایدئولوژی دیده نمی‌شود^{۹۹}. در جدول زیر خلاصه نتایج نهایی این پیمایش آورده شده.

انگیزه	میان
تفریح	۶/۱۰
ایدئولوژی	۵/۵۹
ارزش‌ها	۳/۹۶
درک	۳/۹۲
بهبود	۲/۹۷
حمایتی	۱/۹۷
شغل	۱/۶۷

^{۹۵} آمار مربوط به سال ۲۰۰۶ است.

^{۹۶} Lawler, C., Wikipedia as a learning community: content, conflict and the 'common good', 2006.

^{۹۷} Likert scale

^{۹۸} Nov, O., What motivates Wikipedians, or how to increase user-generated content contribution, 2007

^{۹۹} همان منبع. برای توضیح بیشتر در این موارد رجوع شود به منبع مذکور.

جدول ۲-۲. میانه میزان انگیزه

در حال حاضر هزینه نگه‌داری ویکی‌پدیا توسط شرکت‌های تجاری بزرگ و همچنین هدیه کاربران تأمین می‌شود. در سال ۲۰۰۶، بیش از یک و نیم میلیون دلار به ویکی‌پدیا هدیه شد یعنی روزانه در حدود ۴۳۰۰ دلار^{۱۰۰}.



آمار رشد ترافیک سالیانه ویکی‌پدیا

«ویکی‌پدیا» در آغاز به عنوان یک پروژه مکمل برای دانش‌نامه «نیوپدیا»^{۱۰۱} در ۱۵ ژانویه ۲۰۰۱ توسط جیمی ولز برپا شد. نیوپدیا در سال ۲۰۰۰ به عنوان یک دانش‌نامه الکترونیکی راه‌اندازی شد. این پروژه یک ویکی نبود و بالطبع ویرایش آن برای همه کاربران آزاد نبود و افزودن یک مقاله به آن، روندی نسبتاً طولانی - به منظور مرور مقاله توسط گروهی خبره - داشت.

«بامس»^{۱۰۲} یک شرکت اینترنتی بود که در سال ۱۹۹۶ تأسیس شد و کسب و کار اصلی آن از راه فروش تبلیغات در درگاه جستجوی سایت بامس دات کام بود. این شرکت توسط جیمی ولز^{۱۰۳} و تیم شل^{۱۰۴} برپا شد و بعدها پشتیبانی مالی و تکنیکی را جهت پروژه‌های «نیوپدیا» و «ویکی‌پدیا» فراهم کرد.

تعداد مقالات ویکی‌پدیا در پایان سال ۲۰۰۱ به ۲۰,۰۰۰ رسید که در ۱۸ زبان مختلف قابل دسترس بود. در پایان ۲۰۰۲ این تعداد به ۲۶ زبان افزایش یافت و در پایان سال‌های ۲۰۰۳ و ۲۰۰۴ به ترتیب به ۴۶ و ۱۶۱ زبان رسید.

¹⁰⁰ Wikimedia fundraising: <http://fundraising.wikimedia.org/en/fundcore/browse>

¹⁰¹ Nupedia

¹⁰² Bomis

¹⁰³ Jimmy Wales

¹⁰⁴ Tim Shell

ب) لینوکس (مدل بازار)

شاید بتوان گفت پایبندترین پروژه موفق به مدل بازار، هسته سیستم عامل لینوکس است که ریموند بارها و بارها در کتاب خود به آن اشاره می‌کند. هرچند در نگاه اول توسعه هسته لینوکس یک امر کاملاً مشارکتی و آزاد محسوب می‌شود، لیکن هیچ تغییری بدون تأیید نهایی توروالدز انجام نمی‌پذیرد.

اگر تمرکز می‌کنیم که در پروژه‌های مدل کلیسا شاهد آن هستیم وجود نمی‌داشت، هیچگاه هسته لینوکس به وضعیت فعلی خود نمی‌رسید. امکانات و فراخوانی‌های مختلف امنیتی، پشتیبانی محلی از ریسمان‌ها^{۱۰۵} و موارد مشابه دیگری، همه و همه در نتیجه مشارکت شرکت‌های تجاری مختلفی چون ردهت و در مدل متمرکز و شبیه به کلیسا محقق شده است. در غیر این صورت همانطور که خواهیم گفت، به هیچ عنوان نمی‌توان با ابزار ساده‌ای چون ایمیل، هماهنگی و مدیریت تغییرات یکپارچه بر روی هسته سیستم عامل را کنترل کرد.

اگرچه شاید بتوان مهم‌ترین پروژه مدل بازار را هسته لینوکس دانست، ولی بزروکوف معتقد است این پروژه چندان پروژه پیچیده و چند جانبه‌ای نیست. ایده‌های مهم برنامه‌نویسی و پوسته‌ها و دستورات خط فرمان لینوکس از سیستم عامل تجاری یونیکس اخذ شده و کد منبع آن هم مبتنی بر سیستم عامل مینیکس^{۱۰۶} است. بزروکوف علت اصلی محبوبیت لینوکس را نه پیچیدگی و جامع بودن آن بلکه منتشر شدن کد منبع آن تحت مجوز گنو می‌داند. گنو که کمتر از ۱۰ سال پیش از لینوکس آغاز شده بود، همه چیز داشت به جز هسته یک سیستم عامل؛ چیزی که توروالدز آن را جامعه نرم‌افزار آزاد هدیه کرد.

آمار دقیقی از تعداد کاربران خانگی لینوکس در دست نیست، لیکن طبق آمار رسمی «شرکت بین‌المللی داده»^{۱۰۷} در سال ۲۰۰۴، ۲۵٪ از سرورها و ۲/۸٪ از کاربران خانگی از لینوکس استفاده می‌کردند. عملاً روش مشخصی برای برآورد میزان کاربران لینوکس وجود ندارد و لذا نمی‌توان به آمار IDC لقب رسمی داد. «پروژه شمارنده لینوکس»^{۱۰۸} آمار کاربران لینوکس را در سال ۲۰۰۵ بر اساس پارامترهای مختلف، ۲۹ میلیون نفر تخمین زده است، با فرض اینکه در سال ۲۰۰۱، ۱۸ میلیون نفر بوده‌اند^{۱۰۹}.

ج) اکلیپس (مدل کلیسا)

اکلیپس^{۱۱۰} یک جامعه منبع‌باز است که به تولید پروژه‌های منبع‌باز بر یک سکوی باز توسعه تمرکز دارد^{۱۱۱}. تمرکز اصلی اکلیپس برای تولید یک محیط توسعه مجتمع^{۱۱۲} بر روی سکوی جاوا است، هرچند به هیچ عنوان محدود به آن نمی‌شود. اکلیپس در سال ۲۰۰۱ و با سرمایه اولیه ۴۰ میلیون دلار توسط IBM پایه‌گذاری شد. در این جامعه ۱۵۰ تولیدکننده نرم‌افزار مشارکت کردند و ۱۲۰۰ نفر از ۶۳ کشور مختلف پیشتر در فرایند تولید اکلیپس شرکت داشتند.

¹⁰⁵ Native thread support

¹⁰⁶ Minix

¹⁰⁷ International Data Corporation (IDC)

¹⁰⁸ Linux counter project, <http://i18n.counter.li.org> (accessed: 17 Dec 2007)

¹⁰⁹ همان، <http://i18n.counter.li.org/estimates.php>

¹¹⁰ Eclipse

¹¹¹ <http://eclipse.org>

¹¹² integrated development environment (IDE)

سرمایه‌گذاری IBM و مشارکت شرکت‌هایی نظیر Red Hat، Together soft، Rational، و بعدها BEA، Sybase، SAP، Oracle، Intel، Nokia، و حتی Borland (یکی از پرسابقه‌ترین شرکت‌ها در تولید IDE برای زبان‌های مختلف) باعث شد که اکلپس هم‌اکنون به عنوان محبوب‌ترین IDE جاوا با فاصله زیاد از رقبایش شناخته شده باشد.

اکلپس در ابتدا به صورت غیر متمرکز و توسط نمایندگانی از شرکت‌های مختلف هدایت می‌شد، لکن بعدها در سال‌های ۲۰۰۳ و ۲۰۰۴ به صورت یک بنیاد غیر انتفاعی مستقل درآمد که اعضای راهبردی^{۱۱۳} آن کنترل و هدایت قسمت‌های مختلف آن را به عهده دارند.

با تعریف ریموند این پروژه قطعاً یک پروژه مدل کلیسا خواهد بود که در مدل توسعه منبع‌باز رشد کرده و موفق‌ترین IDE جاوا محسوب می‌شود. جالب است که با وجود پروژه‌های مختلف مدل بازار^{۱۱۴}، هنوز برای توسعه زبان‌هایی مانند ++C (که تمرکز اکلپس بر روی آن نیست) در لینوکس محیط توسعه‌ای بهتر از اکلپس وجود ندارد.

۲،۳،۶ جمع‌بندی مدل‌های ارائه شده

به جرأت می‌توان گفت کمتر برنامه‌نویسی وجود دارد که حاضر شود مجانی نرم‌افزار تولید کند. یکی از علل رشد و توسعه نرم‌افزار منبع‌باز آزاد و بالطبع مجانی، علاقه توسعه‌دهنده آن در حل یک مشکل که پیشتر حل نشده یا راه حل موجود او را ارضا نمی‌کند است. ریموند این دلیل را «خاراندن خارش شخصی توسعه دهنده» می‌خواند^{۱۱۵}. سؤال اینجاست که این توسعه‌دهنده (یا توسعه دهندگان)، تا کجا می‌توانند بدون نیاز به یک پشتیبانی مالی به کار خود ادامه دهند؟

پشتیبانان مدل توسعه بازار - نظیر ریموند - در اینجا به بیان مثال‌های موفقی از مدل بازار متوسل می‌شوند. مثال‌هایی چون هسته لینوکس، سرور وب آپاچی، پایگاه داده PostgreSQL و به طور کلی مواردی از این دست. این مدل هرچند در نگاه اول - به لحاظ تنوع کیفی و گستردگی محصولاتش - بهترین مدل به نظر می‌رسد، لیکن مشکلات متعددی در آن نهفته است.

برای تولید نرم‌افزارهای تجاری که هنوز جایگزینی در نرم‌افزارهای آزاد/منبع‌باز ندارند، باید منتظر خارش شخصی برنامه‌نویسان شد. به علاوه هیچ تضمینی وجود ندارد که هنگامی که یک برنامه‌نویس برنامه منبع‌بازی را ارائه داد، این نرم‌افزار امکانات بیشتری از نمونه غیر منبع‌باز ولی مجانی خود ارائه دهد. به‌راستی چرا برای یک کاربر کامپیوتر باید منبع‌باز بودن یک نرم‌افزار مهم باشد؟ کاربر نهایی^{۱۱۶} هرگز با کد منبع برنامه سر و کار ندارد. حتی بسیاری از کاربران خبره و به بیان ریموند «هکرها» به ندرت به سراغ کد منبع می‌روند، آن هم نه به عنوان اینکه از مکانیزم و صحت عملکرد آن آگاه شوند، بلکه به این دلیل که احتمالاً می‌خواهند آن کد را مورد استفاده مجدد در نرم‌افزار خود قرار دهند.

¹¹³ strategic member

¹¹⁴ به عنوان نمونه: DevC++، Anjuta، Kdevelop

¹¹⁵ Raymond, E., 'The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary, 2001, p. 23.

¹¹⁶ end user

ریموند در مقاله پاتیل جادویی، در دفاع از مدل‌سازی تجاری خود از نحوه کسب و کار مدل منبع‌باز، سه استراتژی را برای تهیه یک سرور وب مطمئن ارائه می‌دهد و با بررسی دو مورد از آن مورد سوم را بهترین می‌داند. در این مدل‌سازی، ریموند دو نوع سود (ارزش) را برای پروژه‌های نرم‌افزاری برمی‌شمرد: ارزش فروش^{۱۱۷} و ارزش استفاده^{۱۱۸}.

راه اول در تهیه یک سرور وب، خریداری و سفارشی کردن یک سرور وب تجاری و اختصاصی است. ریموند مشکل این مورد را عدم تطبیق کامل سرور وب با نیاز مشتری می‌داند، چرا که مشتری محدود به گزینه‌های سفارشی‌سازی است که سرور وب اختصاصی در اختیار او قرار می‌دهد. راه دوم این است که مشتری خود اقدام به نوشتن یک سرور وب کند. هرچند این راه در اکثر موارد چندان مقرون به صرفه نیست، ولی از آنجایی که یک سرور وب پیچیدگی‌های چندان نادارد، می‌تواند مورد بررسی قرار گیرد. ایراد این روش آن است که شرکت با جدا یا بازنشسته شدن نویسنده سرور وب، دوباره با مشکلاتی روبرو خواهد شد. راه سوم - و به نظر ریموند بهترین راه - پیوستن به گروه آپاچی و استفاده از سرور وب آنهاست. این سرور وب منبع‌باز، اختصاصی نیست و کاربر می‌تواند خود به سفارشی‌سازی آن دست بزند و آن را مطابق نیازهای خود بازسازی نماید.

لفکویتز^{۱۱۹}، با رندی خاصی یادآور این نکته می‌شود که کاربرانی که مایل به نگاه کردن به کد منبع و یا تغییر آن باشند، بسیار نادرند. وی راه چهارمی را پیشنهاد می‌کند که اکثر کاربرانی که نیاز به یک سرویس مطمئن و تمام وقت دارند انتخاب می‌کنند: خرید یک سرور وب منبع‌باز از یک توزیع کننده تجاری و استفاده از خدمات آن، بدون کوچکترین نگاهی به کد منبع^{۱۲۰}. لفقویتز با ذکر یک مثال از یک طرح باز ساخت‌افزایی پیچیده این سؤال را مطرح می‌کند که چه کسی به دنبال کد منبع است؟ پاسخ او روشن است و خود نیز به آن پاسخ داده است: برنامه‌نویسان.

لفکویتز در ادامه آماری را به نقل از انجمن فناوری اطلاعات آمریکا^{۱۲۱} ذکر می‌کند که جالب توجه است. به طور تقریبی ۱۰ میلیون کارمند در سال ۲۰۰۱ در حوزه فناوری اطلاعات کار می‌کرده‌اند. ۹۲ درصد از کل این تعداد برای شرکت‌هایی غیر مرتبط با IT کار می‌کردند. برنامه‌نویسان و مهندسان ۲۱ درصد از کارمندان حوزه IT را تشکیل می‌دادند. ۲ میلیون برنامه‌نویس در ایالات متحده یعنی کمتر از یک درصد جمعیت کشور. وی می‌افزاید، به احتمال ۹۹ درصد من یک برنامه‌نویس نیستم. کدام یک از این راه‌ها بهتر است: (۱) یک برنامه دودویی که من پولی بابت آن نپرداختم و با دوبار کلیک کردن روی آن نصب و اجرا می‌شود، (۲) یک برنامه دودویی که من پولی بابت آن نپرداختم و با دوبار کلیک کردن روی آن نصب و اجرا می‌شود و به علاوه فایل‌های مرموزی روی کامپیوتر من می‌سازد، (۳) آرشویی از کد منبع برنامه که من پولی بابت آن نپرداختم و مشروط بر اینکه روش ساختن برنامه دودویی آن را بدانم و تمام ابزارهای توسعه و کتابخانه‌های لازم را در اختیار داشته باشم و به علاوه به اندازه کافی صبور باشم، می‌توانم نهایتاً آن را اجرا کنم.

منظور لفقویتز از آوردن چنین مثالی روشن است. از نظر او برای کاربر نهایی هیچ تفاوتی بین برنامه منبع‌باز و برنامه مجانی (مادامی که برنامه منبع‌باز، فایل دودویی برنامه را هم در اختیار او قرار دهد) وجود ندارد. حتی زمانی که برنامه منبع‌باز، فایل دودویی برنامه را ارائه نمی‌کند، به احتمال زیاد در حدود ۹۹ درصد کاربران خود را از دست خواهد

¹¹⁷ sale value

¹¹⁸ use value

¹¹⁹ Robert M. Lefkowitz

¹²⁰ Lefkowitz, R., Protecting the Innovation Premium, Open source software convention 2003, O'Reilly.

¹²¹ Information Technology Association of America (itaa.org)

داد، چرا که این کاربران ابزار لازم برای ساختن فایل دودویی و اجرای نهایی برنامه را در اختیار ندارند. ارزش یک برنامه منبع باز فقط نزد برنامه‌نویسان روشن است و رویکرد کاربر نهایی نسبت به آن کاملاً خنثی است.

۲،۴ مدل کسب و کار

ریموند در ادامه کتاب «کلیسای جامع و بازار»، در مقاله‌ای تحت عنوان «پاتیل جادویی»^{۱۲۲} به توصیف روش‌های کسب و کار و مباحث اقتصادی در حوزه منبع باز می‌پردازد. بحث اصلی ریموند در این مقاله، معطوف به دو ارزش مختلف برای یک نرم‌افزار است: «ارزش فروش» و «ارزش استفاده».

ارزش فروش یک برنامه، ارزش نهایی آن به عنوان یک کالا است و ارزش استفاده - در معنای نوین اقتصادی - ارزش میانی یک کالا است. برنامه نویسان به ندرت بسته به ارزش فروش حقوق دریافت می‌کنند و معمولاً صرف‌نظر از برنامه‌ای که تولید می‌کنند، حقوق ثابتی را دریافت می‌کنند.^{۱۲۳}

ریموند استدلال می‌کند که اگر ارزش واقعی یک نرم‌افزار از فروش بیت‌های آن به مشتری حاصل می‌شد، بسیاری از سیستم‌عامل‌ها و نرم‌افزارهای کاربردی که همه روزه به صورت غیر قانونی کپی و در نسخه‌هایی کاملاً مشابه نسخه اصلی به مشتریان عرضه می‌شوند، جایی برای درآمد شرکت‌های تولیدکننده باقی نمی‌گذارد. مدل درآمد را برای چنین نرم‌افزارهایی به فروش آن محدود نمی‌شود بلکه فراتر از آن به ارائه نسخه‌های به‌روز رسانی، پشتیبانی فنی و ارائه کتاب‌ها و مستندات استفاده از نرم‌افزار می‌پردازند.

از نمونه‌های بسیار موفق در عرضه تجاری نرم‌افزارهای منبع باز، شرکت ردهت^{۱۲۴} است. ردهت از مدت‌ها پیش به عنوان یکی از توزیع‌های محبوب لینوکس (به صورت مجانی) شناخته می‌شد و همچنان با پشتیبانی از پروژه فدورا^{۱۲۵} چنین جایگاهی را برای خود حفظ کرده است. ردهت علی‌رغم کمک به تولید بسیاری از نرم‌افزارهای منبع باز تحت مجوزهای نرم‌افزار آزاد (نظیر مجوز گنو)، هیچ‌گاه به دنبال سود از فروش مستقیم نرم‌افزارهای خود نبود. ردهت هم‌اکنون از طرق مختلف کسب درآمد می‌کند: نسخه‌های آزمایش شده و مطمئن لینوکس را تحت عنوان Red Hat Enterprise Linux به صورت تجاری عرضه می‌کند، برای این نسخه‌ها و حتی نسخه‌های مجانی توزیع خود پشتیبانی تجاری تعریف می‌کند، دوره‌های مجازی و حضوری آموزش لینوکس با اعطای گواهی معتبر RHCE^{۱۲۶} برگزار می‌کند، خدمات سفارشی سازی و مشاوره‌ای لینوکس را برای کاربردهای خاص به صورت مطمئن ارائه می‌کند و نهایتاً با پشتیبانی از نرم‌افزارهای یک‌تاز تجاری نظیر پایگاه داده اراکل، نفوذ خود را در بازار پررنگ‌تر می‌کند. نفوذ ردهت در سیستم عامل لینوکس به قدری پررنگ است که بسیاری از نرم‌افزارهای مطرح منبع‌بازی که در اکثر توزیع‌های دیگر لینوکس مورد استفاده قرار می‌گیرند و حتی امکانات مهمی از هسته لینوکس، توسط ردهت نوشته شده است. در اواسط سال ۲۰۰۶، ردهت برای افزایش نفوذ خود در عرصه نرم‌افزارهای منبع باز محیط جاوا، شرکت JBoss (دارای یکی از مهم‌ترین پیاده‌سازهای منبع باز Java EE) را به قیمتی حدود ۳۵۰ میلیون دلار خریداری کرد.

¹²² The magic cauldron

¹²³ Raymond, E., 'The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary, Revised Edition, 2001, pp. 118-122.

¹²⁴ Red Hat

¹²⁵ Fedora

¹²⁶ Red Hat Certified Engineer

نمونه بسیار ساده‌تر و سبک‌تر از کسب و کار در پروژه‌های منبع‌باز، چارچوب گزارش‌گیری Jasper Reports برای محیط جاوا است. این پروژه در سال ۲۰۰۱ هنگامی آغاز شد که تئودور دنسیو^{۱۲۷} با بررسی راه‌حل‌های موجود گزارش‌گیری دریافت که بودجه کافی برای استفاده از آنها در پروژه خود را ندارد. با وجود آنکه هم‌اکنون Jasper Reports تحت مجوز آزاد و منبع‌باز LGPL (که به عقیده بسیاری از GNU هم آزادتر است) منتشر می‌شود، لیکن به دلایل مختلف، تولید چنین چارچوب گزارش‌گیری به یک درآمد مناسب برای نویسنده و شرکت او (JasperSoft) مبدل شده. به دلیل سادگی و محبوبیت این نرم‌افزار، توسعه‌دهندگان در استفاده از آن تردیدی به خود راه نمی‌دهند و معمولاً به جای مراجعه به نمونه‌های تجاری (با تعداد کاربران بسیار اندک) از این ابزار استفاده می‌کنند. به دلیل نیاز شدید به چنین ابزاری برای محیط جاوا، تماس‌های توسعه‌دهندگان حتی پیش از منتشر شدن اولین نسخه Jasper Reports با نویسنده آغاز شد. هم‌اکنون درآمدزایی JasperSoft همانند Red Hat از آموزش، پشتیبانی تجاری، راه‌حل‌های هوش تجاری^{۱۲۸} و فروش مستندات و کتاب‌های آموزشی ابزارهای خود حاصل می‌شود ولی تا پیش از تأسیس این شرکت، عمده درآمد پروژه حاصل از مشاوره و فروش مستندات آن بود. جالب است که با وجود محبوبیت زیاد، همچنان مستندات استفاده از Jasper Report به صورت تجاری عرضه می‌شوند و مستندات مجانی مفصلی برای استفاده از آن وجود ندارد.

مورد دیگری که در فرآیند کسب و کار حول فعالیت‌های منبع‌باز باید به آن توجه کرد، موقعیت کاری و شغلی بهتر است. مطلبی که تقریباً در تمامی پیمایش‌های روانشناسی - اجتماعی، از فعالان این حوزه پرسیده می‌شود و بعضاً هم در کنار انگیزه‌های دیگر شخصی چون «شهرت یافتن» و یا «پول درآوردن»، نقش پررنگ‌تری دارد^{۱۲۹}.

در پیمایش جامعی که دانشگاه ماستریخت هلند در سال ۲۰۰۲ انجام داد، ۳/۴۶٪ از پاسخ‌دهندگان بیان کرده‌اند که هیچ نفع مالی از مشارکت بر روی پروژه‌های منبع‌باز نمی‌برند در حالی که مابقی به صورت مستقیم، غیر مستقیم و یا هر دو حالت از پروژه‌های منبع‌باز نفع مالی می‌برند^{۱۳۰}. به نظر می‌رسد آمار ارائه شده در این پیمایش (که مربوط به سال ۲۰۰۲ است) در حال حاضر به سمت منفعت مالی بیشتر متمایل شده باشد چرا که هر روز شرکت‌های تجاری بیشتری برای افزایش سود و یا از دست ندادن نفوذ خود در بازار، هوشمندانه به قافله منبع‌باز می‌پیوندند^{۱۳۱}.

۳ جامعه‌شناسی نرم‌افزار آزاد

مباحث اجتماعی گوناگونی حول منبع‌باز مطرح است. از ماهیت جنبش-گونه آن گرفته تا تغییر در ساختارهای تجاری کلان و همچنین ساختار تفکر مدیران. چیزی که نرم‌افزار آزاد به کاربران خود اعطا می‌کند، هویت و چیزی که به جامعه کاربران اطراف خود می‌دهد انسجام و یکپارچگی است، حال این یکپارچگی به دلیل مخالفت با شرکت‌های بزرگ تجاری (چون مایکروسافت) باشد و خواه به دلیل توافق بر سر موضوعات مختلف.

¹²⁷ Teodor Danciu

¹²⁸ Business Intelligence

¹²⁹ Ghosh, R., Glott, R., Krieger, B., Robles, G., Free/Libre and Open Source software: survey and study, 2002, p. 45.

¹³⁰ همان منبع، ص. ۶۴-۶۵

¹³¹ به عنوان نمونه با آنکه استیو بالمر، مدیر عامل شرکت مایکروسافت در سال ۲۰۰۱ لینوکس را سرطان خوانده بود، هم‌اکنون خود مایکروسافت از تعداد زیادی از پروژه‌های منبع‌باز حمایت می‌کند و حتی با شرکت Novel برای هماهنگی هر چه بهتر لینوکس و ویندوز همکاری می‌کند. رجوع شود به:

Ballmer: "Linux is a cancer", http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer, 2001
<http://www.microsoft.com/opensource>

کسب انسجام اجتماعی - در معنای دورکیمی^{۱۳۲} آن - را می‌توان به سادگی از نوشته‌های ایدئولوگ نرم‌افزار آزاد، استالمن برداشت کرد:

این آزادی‌ها به صورت حیاتی اهمیت دارند. آنها ضروری‌اند، نه به خاطر کاربران منفرد بلکه به خاطر اینکه انسجام اجتماعی ما را بهبود می‌بخشند.^{۱۳۳}

۳.۱ جنبه‌های هویتی و روانی منبع‌باز

بی‌تردید از مهم‌ترین جنبه‌های روانی و اجتماعی جنبش منبع‌باز، جنبه هویت بخشی آن نه فقط به توسعه‌دهنده بلکه به جامعه مصرف‌کننده آن است. نگاهی اجمالی به سمینارها، تالارهای گفتگو و گروه‌های کاربری لینوکس^{۱۳۴} نشان از برقراری انسجامی نسبی در بین کاربران و احساس هم‌فکری بین آنان است. تعداد فراوان توزیع‌های لینوکس - ولو با تعداد کاربران اندک، حس مخالفت با مایکروسافت و نرم‌افزارهای اختصاصی، استفاده از نرم‌افزارهای منبع‌باز و آزاد به جای نمونه‌های بسیار مناسب‌تر ولی غیر آزاد (و لو مجانی) و مواردی از این دست همه حکایت از هویت بخشی است که کاربر نرم‌افزار آزاد از آن کسب می‌کند.

ایدئولوژی استالمن در استفاده نکردن از نرم‌افزارهای اختصاصی^{۱۳۵} و ابداع copyleft (در مقابل copyright) به منظور اشاعه هرچه بیشتر فرهنگ نرم‌افزار آزاد، نشان از هویت‌یابی او و علاقمندان نرم‌افزار آزاد از استفاده از آن است. شاید بتوان در بسیاری از موارد - صرف‌نظر از وجه عقیدتی آن - استفاده نکردن از نرم‌افزارهای غیر آزاد ولی در عین حال مجانی را با استفاده از متد تحلیلی پیر بوردیو^{۱۳۶} و مفهوم تشخیص اجتماعی^{۱۳۷} وی بیان کرد: کاربران سخت‌گیر نرم‌افزارهای آزاد، با استفاده نکردن از هرگونه نرم‌افزار غیر آزاد (تجاری یا غیر تجاری) تمایز خود را از دیگران روشن می‌کنند و منزلت خود را به عنوان یک فرد با آگاهی خاص ارتقا می‌بخشند. دیگرانی که حتی معمولاً چیزی در مورد نرم‌افزار آزاد نمی‌دانند.

برای نویسندگان نرم‌افزارهای آزاد، تنها انگیزه‌های ایدئولوژیک، تشخص و یا حتی موقیت‌های مالی و شغلی بهتر مطرح نیست، بلکه بخش عمده‌ای از انگیزه توسعه نرم‌افزارهای آزاد توسط جامعه کاربران به توسعه‌دهنده القا می‌شود. پس از ارائه نسخه‌های اولیه و یافتن شهرت نسبی در بین کاربران، ایمیل‌های حاوی درخواست برای امکانات جدید و یا گزارش نواقص و خطاهای موجود در برنامه به سوی توسعه‌دهنده و یا جامعه توسعه‌دهندگان (از طریق میلینگ لیست‌ها) سرازیر می‌شود. نفس اینکه برنامه توسعه‌دهنده دارای کاربران مشتاق به آن باشد، به او امید می‌دهد و او را به سمت ادامه توسعه و رفع نواقص تشویق می‌کند. ریموند در مقاله کلیسای جامع و بازار به این نکته اشاره می‌کند و تابع منفعت^{۱۳۸} در توسعه پروژه‌های منبع‌باز را/رضای نفس و کسب شهرت در بین توسعه‌دهندگان دیگر می‌داند.

¹³² امیل دورکیم، جامعه‌شناس شهیر فرانسوی سده ۱۹

¹³³ Richard Stallman, Why Open Source misses the point of Free Software, <http://www.gnu.org/philosophy/open-source-misses-the-point.html> (accessed 17 Sep 2007)

¹³⁴ سایت‌های LUG: Linux User Group

¹³⁵ ر.ک. بخش نرم‌افزار آزاد، مفاهیم

¹³⁶ Pierre Bourdieu

¹³⁷ social distinction

¹³⁸ utility function

۳،۲ کسب و کار

همان‌طور که پیشتر در مورد مدل‌های کسب و کار در حوزه منبع‌باز توضیح داده شد، از علل اساسی موفقیت نرم‌افزار آزاد، اقبال شرکت‌های تجاری به آن و مدل توسعه منبع‌باز است. مدلی که برای کاربر نهایی همان مزایایی را به دنبال دارد که از یک نرم‌افزار آزاد انتظار می‌رود، یعنی امکان تغییر آزادانه کد منبع، توزیع مجدد آن و مواردی از این دست.

فرایند نوین کسب و کاری که در محیط توسعه منبع‌باز شکل گرفته است، اولاً باعث منسوخ شدن مدل‌های پیشین در مورد نرم‌افزارهای عام-منظوره و اختصاصی و حتی مدل فرایند توسعه آن‌ها می‌شود و ثانیاً به رشد و فراگیر شدن نرم‌افزارهای منبع‌باز از سویی و کم‌اهمیت شدن پنهان‌سازی کد منبع از سوی دیگر کمک می‌کند.

در کنار این مسائل، توسعه‌دهندگان منبع‌باز عمدتاً بیش از دیگر توسعه‌دهندگان پروژه‌های تجاری درگیر بخش‌های مختلف نرم‌افزار از طراحی و مهندسی نیاز گرفته تا پیاده‌سازی و تست و حتی ارتباط با مشتری می‌شوند. این موضوع از طرفی نگران‌کننده و از طرف دیگر مایه امیدواری است. نگران‌کننده به این خاطر که توسعه‌دهنده باید در بسیاری از مسائل وارد شود و درگیر بخش‌های مختلف فرایند تولید نرم‌افزار شود. این مسئله به خلاقیت و توانایی زیادی نیاز دارد و همان‌طور که اشاره خواهیم کرد تنها اقبال نخبه جامعه از پس آن بر می‌آیند. اما امیدوار کننده به این دلیل که نویسنده برنامه یک آدم جزئی-کار نخواهد شد و عملاً بر کل فرایند تولید نظارت و دخالت دارد. دخالت او در بخش‌های مختلف فرایند تولید به رشد و شکوفایی شخصی او و توانایی او در حل مشکلات و چند بعدی شدن کمک می‌کند.

در نگاه مارکسیستی، کارگری که صرفاً قسمتی از فرایند تولید را به عهده می‌گیرد، کار و یا زمان خود را به صورت مجرد به کارفرما می‌فروشد. در این صورت نه تنها خود را تولید کننده محصول نهایی نمی‌داند و هیچ علقه‌ای به آن ندارد، بلکه دچار از خود بیگانگی از کار می‌شود و به دلیل اینکه تولید کار را فراتر از توانایی خود می‌یابد، برای آن قائل به تقدسی کاذب می‌شود. با کمی اغماض می‌توان این رویکرد را به تولید نرم‌افزار نیز بسط داد. تا وقتی که برنامه‌نویسان نرم‌افزارهای آزاد، بت ناممکنی تولید نرم‌افزارهای پیچیده را نشکنند، سایه برتری و تسلط این نرم‌افزارها بر سر آنان سنگینی می‌کند.

۳،۳ نخبه‌گرایی

اساساً منبع‌باز جنبشی متعلق به نخبه‌ها و افرادی با منزلت علمی بالا در جامعه است. صرف‌نظر از پیشینه دانشگاهی آن (MIT) و حمایت بسیاری از محافل معتبر علمی از آن، کاربران آن‌را هم اقبال نخبه جامعه تشکیل می‌دهند. یک کاربر نهایی عادی کامپیوتر ترجیح می‌دهد در قبال پرداخت مبلغی، نرم‌افزاری تست شده و به فرم دودویی بخرد و یا لااقل از فرم دودویی و قابل اجرای یک نرم‌افزار کد-بسته مجانی استفاده کند تا اینکه نگران کامپایل کردن کد منبع برنامه و ارضای پیش‌نیازهای باشد.

موفقیت‌های اخیر منبع‌باز در جذب بسیاری از کاربران عادی دقیقاً به دلیل حرکت او به سمت تعدیل رویکرد خود برای جامعه کاربران عادی کامپیوتر است. هم‌اکنون اوبونتو^{۱۳۹} محبوب‌ترین توزیع لینوکس است که در حدود ۳۰٪ از

از کاربران لینوکس از این توزیع استفاده می‌کنند^{۱۴۰}. در مقابل رویکرد رسمی استالمن، اکثر توزیع‌های محبوب کنونی بیشتر به راحتی کاربر نهایی پایبندند تا رعایت اصول اساسی «بنیاد نرم‌افزار آزاد». به عنوان نمونه بر روی انبار بسته‌های نرم‌افزاری اوبونتو، انواع نرم‌افزارهای اختصاصی و کد-بسته که امکان توزیع رایگان داشته باشد، وجود دارد و به صورت گسترده مورد استقبال کاربران قرار می‌گیرد.

۳،۴ آیا منبع‌باز یک انقلاب است؟

آیا جنبش منبع‌باز یک حرکت انقلابی است و آیا این انقلاب پیروز شده است؟ چرا برخی منبع‌باز را انقلاب می‌دانند؟ منبع‌باز جایگزینی برای کدام سلف است؟ اینکه منبع‌باز یک جنبش مؤثر و قدرتمند ایجاد کرده است، کمتر مورد تردید قرار گرفته است، لکن استفاده لفظ انقلاب برای تغییراتی که جنبش منبع‌باز در مدل توسعه، درآمد و به طور کلی بازار نرم‌افزار ایجاد کرده است، نوعی تسامح در تعبیر به نظر می‌رسد.

پیش از هر چیز بهتر است مفهوم انقلاب را تعریف کنیم و تفاوت‌های آن با واژه‌هایی نظیر کودتا، اصلاح و یا شورش را بیان کنیم.

برای تبدیل شدن مجموعه‌ای از رویدادها به یک انقلاب، باید مجموعه‌ای از عوامل گرد هم آیند^{۱۴۱}:

۱۱. رویدادهایی انقلاب به شمار می‌آیند که متضمن یک جنبش اجتماعی توده‌ای^{۱۴۲} باشند. این شرط مواردی را که حزبی از طریق فرایندهای انتخاباتی به قدرت می‌رسد، یا گروه کوچکی مانند رهبران ارتش، قدرت را تصرف می‌کنند، مستثنی است.

۱۲. انقلاب به فرایندهای عمده اصلاح یا دگرگونی می‌انجامد. جان‌دان یادآور گردیده است که کسانی که قدرت را تصاحب می‌کنند باید به طور قطع بیشتر از آنهایی که سرنگون شده‌اند توانایی حکومت بر جامعه‌ای را داشته باشند که کنترل آن را به دست می‌گیرند؛ رهبری باید حداقل توانایی دستیابی به بعضی از هدف‌هایش را داشته باشد. در جامعه‌ای که یک جنبش در به‌دست آوردن زمام رسمی قدرت موفق می‌شود، اما نمی‌تواند به‌طور مؤثر حکومت کند، نمی‌توان گفت که انقلابی رخ داده است؛ چنین جامعه‌ای بیشتر احتمال دارد که جامعه‌ای دستخوش آشوب و هرج و مرج، یا در خطر فروپاشی باشد.

۱۳. انقلاب متضمن تهدید یا کاربرد خشونت از جانب کسانی است که در جنبش توده‌ای شرکت می‌کنند. انقلاب‌ها تغییرات سیاسی‌ای هستند که در برابر مقاومت مقامات نظام پیشین به‌وجود می‌آیند که حاضر نیستند بدون تهدید به استفاده از وسایل خشن یا کاربرد واقعی این وسایل از قدرتشان صرف‌نظر کنند.

اگر این سه معیار را با یکدیگر جمع کنیم می‌توانیم انقلاب را به این صورت تعریف کنیم: «انقلاب عبارت است از تصرف قدرت دولتی از طریق وسایل خشن به‌وسیله رهبران یک جنبش توده‌ای، به منظور استفاده بعدی از آن برای

¹⁴⁰ 2007 Desktop Linux Survey results revealed, <http://www.desktoplinux.com/news/NS8454912761-.html>

¹⁴¹ آنتونی گیدنز، جامعه‌شناسی، ۱۳۸۴، تهران، ص ۶۶۰

¹⁴² mass social movement

ایجاد اصلاحات عمده اجتماعی^{۱۴۳}». پس از بیان این تعریف برای روشن شدن کارکرد جنبش منبع‌باز به تطبیق آن با تعریف فوق می‌پردازیم.

جنبش منبع‌باز با مورد اول بسیار سازگار است، چرا که حقیقتاً با یک جنبش اجتماعی (یا مجازی) توده‌ای همراه بوده است. توده‌ای بودن را در این تعریف به معنای حمایت بخشی از اجتماع کاربران کامپیوتر می‌گیریم، و این روشن است که منبع‌باز جامعه کاربران گسترده‌ای دارد. این جامعه کاربری، محدود به جمعیت خاصی (مثلاً اعضای بنیاد گنو) نمی‌شود هم‌اکنون در همه کشورهای جهان مبلغین و طرفدارانی دارد.

درباره تطبیق موارد دوم و سوم باید کمی با احتیاط برخورد کرد. برآستی آیا منبع‌باز به فرایند عمده اصلاح انجامیده؟ حداکثر می‌توان ادعا کرد منبع‌باز به عنوان یک جایگزین برای نرم‌افزارهای غیر آزاد و اختصاصی مطرح شده است، ولی نکته‌ای که باید مد نظر داشت وسعت این اصلاح است. هرچند سهم منبع‌باز از بازار سرورهای وب قابل توجه است (سرور وب آپاچی حدود ۶۰ درصد این بازار را در اختیار دارد)، لکن سهم منبع‌باز از بازار سیستم عامل و یا حتی نرم‌افزارهای تخصصی نظیر محاسبات مهندسی، شبیه‌سازها، نرم‌افزارهای گرافیکی و سایر نرم‌افزارهایی که جنبه عام ندارند بسیار ناچیز و بعضاً هیچ است. اصلاحی که منبع‌باز ایجاد کرده است، حداکثر باعث در نظر گرفتن آن در محاسبات بازار و به حساب آوردن آن به عنوان یک رقیب جدی برای نرم‌افزار اختصاصی می‌شود. در دنیای منبع‌باز ما زمامدار رسمی امور نداریم و اگر هم نهایتاً بنیاد گنو یا پیشگامان منبع‌باز را زمامدار آن بدانیم، با اختلافات ریشه‌ای ایدئولوژی آنها چه می‌کنیم؟ اگر منبع‌باز زمامدار رسمی داشته باشد، دچار همان مشکلی می‌شود که نرم‌افزار اختصاصی شده و آن هم انحصار^{۱۴۴} است. متکثر بودن زمامداری منبع‌باز است که آن را از نرم‌افزار اختصاصی متمایز می‌سازد. آیا منبع‌باز به صورت مؤثر حکومت می‌کند؟ بعید به نظر می‌رسد بتوان حکومتی برای نرم‌افزار آزاد متصور شده، چرا که نرم‌افزار آزاد همواره به عنوان رقیب و نه یکه‌تاز عرصه نرم‌افزار مطرح بوده است، لذا مشکل بتوان حکومتی برای آن تصور کرد.

در روند فراگیری نرم‌افزار آزاد، خشونت و دفعیتی در کار نبوده، بلکه جنبش نرم‌افزار آزاد به صورت کاملاً نرم و آهسته و پیوسته به مقابله با نرم‌افزار اختصاصی و مدل‌های انحصاری برخاسته است. هرچند این جنبش تهدیدی برای مدل‌های اختصاصی نرم‌افزاری محسوب می‌شود، لکن - حداقل تا کنون - این تهدید به اصلاح روندها و رقابت برای کسب بازار بین منبع‌باز و نرم‌افزارهای انحصاری تبدیل شده است. سازندگان نرم‌افزارهای تجاری هر روز برای مقابله با تهدید منبع‌باز، به گنجانیدن امکانات جدید و اصلاح نرم‌افزارهای خود اقدام می‌کنند.

تیم آراییلی^{۱۴۵} - مدیرعامل انتشارات ارایلی - برای بیان این تغییر تحت تأثیر کتاب معروف تامس کوهن^{۱۴۶} «ساختار انقلاب‌های علمی^{۱۴۷}» از واژه انتقال انگارامی یا *paradigm shift* استفاده می‌کند^{۱۴۸}. کوهن این واژه را برای بیان تغییر انگاره‌های ذهنی (قالب‌ها و الگوها) برای برخورد با قضایای علمی تجربی استفاده کرد. پارادایم از نظر کوهن - در

^{۱۴۳} همان منبع

^{۱۴۴} monopoly

^{۱۴۵} Tim O'Reilly

^{۱۴۶} Thomas Kuhn

^{۱۴۷} The structure of scientific revolution

^{۱۴۸} Wynants, M., Cornelis, J., How Open is the Future?, 2005, pp, 85-109

معنای اعم - مشتمل است بر مفروضات کلی نظری و قوانین و فنون کاربرد آنها که اعضای جامعه علمی خاصی آنها را برمی‌گیرند^{۱۴۹}.

هرچند پارادایم هم لزوماً بهترین واژه برای توضیح این تغییر نیست، لکن نکته آموزنده این اصطلاح در وجه ذهنی بودن تغییر رویکردها و ساختارها است. تغییر پارادایم، به معنای تفاوت عمده‌ای است که هم‌اکنون در ذهن افراد شکل گرفته: منبع‌باز نه تنها به عنوان رقیبی برای نرم‌افزار اختصاصی مطرح شده بلکه در برخی عرصه‌های نرم‌افزاری یکه‌تاز است^{۱۵۰} و حتی مدل (یا مدل‌های) توسعه و کسب و کار معینی دارد که امتحان خود را پس داده است^{۱۵۱}. با این بیان اگر برای تغییر پارادایم از لفظ/انقلاب هم استفاده شود، حدود آن روشن است و این معنا به اصلاح شبیه‌تر است.

۴ نتیجه‌گیری

در این پژوهش روند تدریجی رشد و توسعه مفهوم نرم‌افزار آزاد و منبع‌باز در جامعه نرم‌افزاری را بررسی کردیم و نشان دادیم که خاستگاه اولیه نرم‌افزار آزاد، محافل دانشگاهی بوده‌اند^{۱۵۲}. در بررسی روند تاریخی نکته قابل توجه سیر دوری نرم‌افزارها بود که بدواً به صورت آزاد، منبع‌باز و غیر اختصاصی تولید می‌شدند (دهه‌های ۵۰ و ۶۰)، پس از آن روند تولید نرم‌افزار به مدل تجاری و اختصاصی بالغ‌تری تغییر یافت که در آن محصول نرم‌افزاری به عنوان کالایی برای فروش درآمد. در این مدل - به دلیل اختصاصی بودن - کد منبع نرم‌افزار از کاربران نهایی و توسعه‌دهندگان پنهان می‌ماند تا شرکت تولید کننده مزیت رقابتی خود را حفظ کند. نهایتاً در دوره حاضر، روند تولید نرم‌افزار به مدلی مختلط (ترکیبی از مدل‌های آزاد و همچنین اختصاصی) مبدل شده است.

در ادامه، با معرفی مدل‌های مختلف تولید در عرصه نرم‌افزار آزاد، ملهم از کتاب «کلیسای جامع و بازار» اریک ریموند، با رویکردی انتقادی به بررسی دقیق و موشکافانه میزان موفقیت هر یک از این مدل‌ها پرداختیم مشکلات و مزایای هر یک را ذکر کردیم. در این بخش به نکاتی در باب مدیریت و راهبری پروژه‌های منبع‌باز و همچنین نرخ شکست این پروژه‌ها پرداختیم و روش‌ها و ابزار مدیریتی مناسب را برشمردیم. نهایتاً برای جمع‌بندی و مطابقت مدل‌سازی ارائه شده با پروژه‌های جهان واقع، سه نمونه پروژه موفق را برشمردیم که دو مورد آن مستقیماً پروژه نرم‌افزاری بودند (هسته لینوکس و اکلیسی) و یک مورد آن نیز یک دانشنامه اینترنتی بود که آزادانه توسط اشخاص مختلف ویرایش می‌شد (ویکیپدیا). نشان دادیم که مدل‌سازی‌های ساده‌انگارانه چه در حوزه اقتصادی و چه توسعه نرم‌افزار، پاسخگوی پرسش‌های کنجکاوانه در مورد موفقیت‌های نرم‌افزار آزاد نیست، و به صورت کاملاً غیر متمرکز نمی‌توان پروژه‌های کوچک را مدیریت کرد، چه رسد به پروژه‌های متوسطی و بزرگی که بیش از ده‌ها نفر در تولید آن دخیل هستند. تلاش کردیم با دیدگاهی چندجانبه‌تر و با در نظر گرفتن کاربر نهایی و مهم‌تر از آن کسب و کار و سود تجاری حول نرم‌افزار آزاد به تبیین دقیق‌تر موفقیت‌های آن بپردازیم.

نهایتاً با رویکرد یک جنبش به نرم‌افزار آزاد، و با استفاده از مفاهیم و نظریات جامعه‌شناسی، مفاهیمی چون هویت، نخبه‌گرایی، کسب و کار و انقلاب را در این عرصه بررسی کردیم و نشان دادیم از عمده‌ترین دلایل موفقیت و

¹⁴⁹ آلن چالمرز، چستی علم، ترجمه سعید زیباکلام، ص ۱۰۸.

¹⁵⁰ رجوع کنید به مورد لمپ (LAMP): لینوکس، آپاچی، مای‌سکوئل، پی‌اچ‌پی.

¹⁵¹ رجوع کنید به مورد شرکت ردهت (RedHat)

¹⁵² ر.ک. تاریخچه پروژه گنو (ریچارد استالمن) و همچنین پروژه BSD.

همه‌گیر شدن نرم‌افزار آزاد، نه منبع‌باز بودن آن بلکه شکل‌گیری یک مدل متکامل تجاری و کسب و کار حول آن است. برای تشریح دقیق‌تر این مدل نمونه‌ای از شرکت‌های موفق این حوزه را ذکر کردیم.

۵ منابع

- [1] Mulgen G., Steinberg T., Salem O., *Wide Open: Open Source Methods and their future potential*, 2005, Demos, London.
- [2] Transcript of Richard Stallman on GPLv3 in Brussels, Belgium; 1st of April 2007: <http://fsfeurope.org/projects/gplv3/brussels-rms-transcript> (accessed 16 Sep 2007).
- [3] Wikipedia, the free encyclopedia.
- [4] Bruce M. (2006), Why you should use a BSD style license for your Open Source Project, http://www.freebsd.org/doc/en_US.ISO8859-1/articles/bsd-gpl/index.html (accessed 6 Jun 2007).
- [5] Raymond, E., *The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary*, Revised Edition, 2001, O'Reilly, United States of America.
- [6] Stallman, R., Transcript of Richard Stallman on GPLv3 in Brussels, Belgium; 1st of April 2007, <http://fsfeurope.org/projects/gplv3/brussels-rms-transcript.en.html> (accessed 6 Jun 2007).
- [7] Stallman, R., Why Open Source misses the point of Free Software, GNU Project, <http://www.gnu.org/philosophy/open-source-misses-the-point.html> (accessed 7 Jun 2007).
- [8] Raymond, E., Goodbye, “free software”; hello, “open source”, 2004, <http://www.catb.org/~esr/open-source.html> (accessed 7 Jun 2007).
- [9] Fogel, K., *Producing Open Source software*, 2005, O'Reilly, United States of America.
- [10] Moon, J. Y. and Sproull, L., Essence of distributed work: the case of the Linux kernel, *First Monday*, volume 5, number 11, November 2000, http://firstmonday.org/issues/issue5_11/moon/index.html (accessed 22 Jun 2007).
- [11] DiBona, C., Cooper, D., Stone, M., *Open Sources 2, the continuing evolution*, 2nd ed, 2005, O'Reilly, United States of America.
- [12] Wynants, M., Cornelis, J., *How Open is the Future? Economic, social & cultural scenarios inspired by Free & Open-Source software*, 2005, VUB Brussels University Press, Brussels, Belgium.
- [13] Benussi, L., *Analysing the technological history of the Open Source phenomenon. Stories from the Free Software evolution, working paper, version 3.0*, 2005, Department of Economics, University of Turin.
- [14] Robles, G., *A Software Engineering approach to Libre Software*, *Open-Source-Jahrbuch 2004*, Berlin (Lehmanns Media) 2004.
- [15] Brooks, Frederik P., *The mythical man-month: essays on software engineering*, 20th anniversary ed., 1995, Addison Wesley, United States of America.

- [16] Lawler, C., Wikipedia as a learning community: content, conflict and the 'common good', Open Source Jahrbuch 2006, 2006, Germany.
- [17] Schroer, J., & Hertel, G., Voluntary engagement in an open web-based encyclopedia: wikipedians, and why they do it, version 1.0, Jan. 2007, University of Wuerzburg.
- [18] Nov, O., What motivates Wikipedians, or how to increase user-generated content contribution, 2007, Communications of the ACM.
- [19] Mockus, A., Fielding, R., Herbsleb, J., Two case studies of open source software development: Apache and Mozilla, 2002, ACM transactions on software engineering and methodology, vol. 11, no. 3, pp. 309-346.
- [20] Lefkowitz, R., Protecting the Innovation Premium, Open source software convention 2003, O'Reilly, http://conferences.oreillynet.com/cs/os2003/view/e_sess/3940 (accessed 24 Aug 2007).
- [21] Bezroukov, N., (Slightly skeptical) annotated collection of quotes from Eric Raymond's CatB book, 2000, http://www.softpanorama.org/OSS/ESR_Interviews.shtml (accessed 24 Aug 2007).
- [22] Bezroukov, N., A second look at the cathedral and bazaar, First Monday, vol. 4, no. 12, 1999), http://firstmonday.org/issues/issue4_12/bezroukov/index.html (accessed 25-8-2007).
- [23] David Bretthauer, Open Source software: A history, <http://www.lita.org/ala/lita/litapublications/ital/2101bretthauer.cfm>, (accessed 17 Sep 2007).
- [24] Ghosh, R., Glott, R., Krieger, B., Robles, G., Free/Libre and Open Source software: survey and study, International Institute of Infonomics, University of Maastricht, The Netherlands, 2002.

[۲۵] آنتونی گیدنز، جامعه‌شناسی، ترجمه منوچهر صبوری، ۱۳۸۴، تهران، نشر نی، چاپ چهاردهم.

[۲۶] آلن اف. چالمرز، چیستی علم، درآمدی بر مکاتب علم‌شناسی فلسفی، ترجمه سعید زیباکلام، ۱۳۸۴، تهران، سازمان مطالعه و تدوین کتب علوم انسانی دانشگاه‌ها (سمت).