

Michal Pěnka | 4IT460 – Analýza trhu IT

Open-source

Výhody, nevýhody a motivace účastníků řetězce

5/27/2010

Contents

Úvod	3
Co si pod pojmem open-source představit?	3
Jaké jsou tedy výhody?	7
...ale určitě existují i nevýhody	8
Otevřít či neotevřít.....	9
Užívat či neužívat.....	10
Příklad č. 1 – Android	11
Příklad č. 2 – OpenSolaris	12
Závěr	13
Zdroje	14

Úvod

Jmenuji se Michal Pěnka a pracoval jsem ve společnosti Sun Microsystems na pozici *open-source enthusiast*, tedy doslova jako open-source nadšenec. To musím hned na úvod zdůraznit, abych předešel jakékoliv spekulaci o mém postoji k této problematice. Říkám to na rovinu, open-source mě baví, věřím v něj, stal se dokonce mým životním stylem, již několik let jsem členem tzv. open-source komunity, mám ho prostě rád.

Přes to všechno se domnívám, že jsem schopný podat poměrně objektivní pohled na věc a naopak využít svých bohatých zkušeností pro tuto kritiku. Jsem si plně vědom mnoha výhod tohoto fenoménu, stejně tak jsem se již setkal s nevýhodami, o kterých se často ani neví a jsou spíše předmětem živých diskusí mezi „IT odborníky“, často až příliš emočně vypjatých střetů či flamů na diskusních fórech.

V této práci bych rád představil open-source člověku, který netuší, co tento pojem znamená, a doposud si nebyl schopný dát věci do souvislostí. Zmíním výhody a nevýhody, dále pak důvody proč se zapojit či proč naopak od open-source modelu ustoupit při vývoji, užívání či distribuci software. Na závěr uvedu dva velké open-source projekty a vzpomenu na jejich příběh.

Co si pod pojmem open-source představit?

Tuto otázku je třeba zodpovědět hned v úvodu, přestože si troufám tvrdit, že všeobecné povědomí o open-source určitě narůstá. Zkusím to vysvětlit na velmi zjednodušeném příkladu.

Když si zakoupíme počítač, například notebook, je zřejmé, že nám je k ničemu bez aplikací, které na něm budeme provozovat. Chceme psát dokumenty, používat tabulky, brouzdat po internetu, psát e-maily, poslouchat hudbu, bavit se. Tyto aplikace musí někdo vytvořit. Asi všichni víme, že tvůrcem aplikace je programátor. Co takový programátor dělá?

Z pohledu nezasvěceného člověka sedí na židli před obrazovkou a „mlátí“ do klávesnice, občas na ní svačí a mnohdy i spí¹. Programátoři píší tzv. *zdrojový kód* (tedy posloupnosti příkazů – udělej tohle, pak tamto a pokud vyjde takhle hodnota, pokračuj tímto...), který je základem aplikace. Pokud je programátor zručný a počítač jeho zdrojovému kódu rozumí, aplikace z něho nakonec opravdu vzejde – představme si pro tuto chvíli například grafickou kalkulačku.

Zdrojový kód je výborná věc, protože je to jakýsi jazyk, kterým spolu mluví nejen programátor s počítačem, ale i programátor s jinými programátory. No, ne že byste v kanceláři programátorů slyšeli, jak se baví o včerejším fotbalovém zápase ve zdrojovém kódu. Spíše jsem to myslel tak, že zdrojový kód může být vytvářen několika programátory najednou, mohou si ho mezi sebou vyměňovat a sdílet.

Ve škole nás nicméně učili, že počítač nepracuje ve zdrojovém kódu, ale v jedničkách a nulách, tedy *binárním kódu*. Proto musí dojít k tzv. *překladu*. Překlad je převod zdrojového kódu, kterému rozumí člověk, do kódu binárního, kterému zase rozumí počítač. Nutno dodat, že tento překlad je jednosměrnou záležitostí, tedy z binárního kódu se jen velmi těžko², pokud je to vlastně vůbec možné, vytváří zpětně zdrojový kód.

Po tom, co je zdrojový kód přeložen do binárního kódu, je konečně spustitelný. Většina koncových uživatelů (to jsme my, neprogramátoři) získá (koupí na CD, stáhne z internetu, získá od kamaráda...) tento spustitelný binární kód, spustí ho a může aplikaci (například onu kalkulačku) používat. Mohlo by se zdát, že příběh o grafické kalkulačce končí a ve většině případů tomu tak opravdu je.

¹ Autorova osobní zkušenost s programátory, která se může značně lišit od všeobecné skutečnosti

² Převodem binárního kódu do zdrojového kódu se zabývají reverzní inženýři, jejichž práce za prvé velmi zdlouhavá a neefektivní a za druhé především nepochopitelná pro běžné smrtelníky, dokonce i ty nejlepší programátory

Určitě by ale pozorného čtenáře napadla myšlenka, zda by nebylo koncovým uživatelům vhodné distribuovat ne spustitelný binární kód, ale rovnou ten zdrojový. Jistě, mělo by to určitě mnoho výhod, které budeme rozebírat v následující kapitole, ale je tu jedno velké ALE (vlastně jich je více, ale o tom také budeme hovořit dále)... Většina koncových uživatelů neumí nebo nemá vybavení na to, aby převáděla zdrojový kód na kód binární. Navíc je to pro mnoho z nich zbytečné a je to opravdu neefektivní - proč ztrácet čas, když to udělá programátor jednou pro tisíce uživatelů?

Open-source přístup přichází s řešením – programátor nabídne všem uživatelům binární kód. Vedle toho však poskytne i zdrojový kód.

Ve většině případů mají dnešní open-source projekty webové úložiště zdrojového kódu, do kterého mohou uživatelé či jiní programátoři nahlížet přes internet. Vedle tohoto webového úložiště však existuje mnoho distribučních kanálů (například opět web, ale i krabice s CD atp.) pro binární kód.

Software, který se výše uvedeným přístupem řídí, je tzv. *open-source software*, tedy software s otevřeným zdrojovým kódem. Často se uvádí jen zkratka OSS. Pokud se jedná o tzv. *svobodný software* (free software), zkratka je poté rozšířena na FOSS či, aby byl zřejmý význam slova free, přidává se i slovo libre a vzniká FLOSS.

Jaký je rozdíl mezi svobodným a nesvobodným open-source software? Svobodný je ten, který dává nabyvateli (tedy koncovému uživateli, programátorovi, organizaci atp.), zpravidla formou licence, svobodu ve smyslu další distribuce, úprav či užití. Většinu FLOSS tedy získáme „zdarma“, kamarádovi ji „zdarma“ a bez jakéhokoliv postihu přepálíme na CD či mu ji rovnou nainstalujeme na počítač a pokud máme programátorské schopnosti, můžeme ji dále upravovat. Tato svoboda je samozřejmě relativní a je vymezena právě licencí (doporučuji přečíst

alespoň český překlad nejpoužívanější FLOSS licence *GPL* v její třetí verzi na adrese <http://www.gnu.cz/article/32/>).

Nesvobodný software je poté ten, který buď vůbec zdrojové kódy neposkytuje, nebo jen za velmi přísných podmínek (například smíme nahlížet, nesmíme upravovat či dále distribuovat atp.). Nesvobodný software se často označuje také jako *proprietární software*.

Kategorizaci svobodného a nesvobodného software lze dále lépe prostudovat na <http://www.gnu.org/philosophy/categories.cs.html>.

Je třeba ještě zdůraznit, že open-source software neznamena, že je zdarma. Ostatně sám velký Richard Stallman, zakladatel open-source „hnutí“ GNU, prohlásil³:

"Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer."

Je mnoho open-source software, především toho svobodného, který lze bez poplatku získat, distribuovat a upravovat. Existují však i projekty, kdy lze za peněžní sumu „přikoupit“ zdrojový kód.

Na závěr uvedu nejznámější příklady OSS i proprietárního software.

OSS	Proprietární
Linux, FreeBSD, OpenSolaris	MS Windows, Apple Mac OS X ⁴
Apache web server	MS IIS
Gimp, Inkscape	Adobe Photoshop, Corel Draw
VLC	Apple iTunes, Windows Media Player
Mozilla Firefox, Thunderbird et al.	MS Internet Explorer, Apple Safari ⁵

³ Nemá smysl překládat, protože překladem do češtiny se vytrácí význam; v podstatě jde o dvojí význam slova free – zdarma (pivo zdarma) a zároveň svobodný (svobodný projev).

⁴ Jádro systému, Darwin, má otevřený kód

⁵ Opět část systému, tzv. renderovací jádro WebKIT, má otevřený kód

Jaké jsou tedy výhody?

Je zřejmé, že na výhody a nevýhody je třeba nazírat z pohledu jednotlivých účastníků řetězce. Já jsem si vybral následující:

- programátor či firma vyvíjející software,
- distributor,
- koncový uživatel či firma užívající software.

Pro **programátora** může open-source přístup nabízet mnoho výhod. Za prvé, po tzv. *otevření zdrojového kódu*, například jeho vystavením na webu, může získat spolupracovníky z celého světa, kteří mohou mít zájem podílet se na vývoji. Tím se vývoj mnohonásobně urychlí, zefektivní a velmi často zkvalitní. I v rámci programátorů existují specialisté – někdo se zaměřuje na řízení projektu, jiný zase na vývoj aplikační logiky, další je například specialistou na uživatelská rozhraní. Je tak dost pravděpodobné, že si programátoři k sobě najdou chybějícího specialistu z té oblasti programování, ve které nejsou sami tak dobří.

Dále si představme situaci, kdy chceme svým firemním partnerům darovat krátké video představující nový produkt. Abychom našim partnerům usnadnili práci s přehráváním videa, přidáme k tomu navíc i aplikaci – videopřehrávač. Dostáváme se tedy do role **distributora**. Protože dodavatel (programátor) videopřehrávače zveřejnil zdrojové kódy a umožnil nám software dále modifikovat a distribuovat, můžeme si přehrávač upravit tak, aby rovnou po vsunutí CD do mechaniky počítače, na kterém budeme video distribuovat, spouštěl ono video představující produkt. Zároveň máme jistotu, že software, který našim partnerům spolu s videem distribuujeme, neobsahuje viry, protože jsme celý zdrojový kód analyzovali předem.

Jak z open-source těží **uživatelé**? Za prvé je to rychlost vývoje (viz výše výhody programátora), takže se k němu dostávají žhavé novinky mnohem rychleji, než kdyby na software pracoval jeden programátor či jedna firma.

Za druhé je to zmiňovaná bezpečnost. Přestože je to tak trochu sporný bod, o kterém se vedou diskuse, je velmi pravděpodobné, že kód, který používá mnoho uživatelů (včetně samotných programátorů) a který je umístěn na viditelném místě někde na webu, je revidovaný mnoha programátory a bezpečnostními specialisty. Za třetí, software s otevřeným kódem je zpravidla značně populární mezi programátory i uživateli, takže se okolo něj vytváří tzv. komunita. Ta je v případě potřeby (například při potřebě nápovědy, při zjištění chyby atp.) ochotná pomoci.

...ale určitě existují i nevýhody

Představme si firmu, která investuje do vývoje bankovního systému, který zpracovává bankovní transakce. Pokud by tato firma zveřejnila zdrojový kód, přišla by tím pravděpodobně „na buben“ – jiné firmy by kód převzaly, začaly vyvíjet a prodávat bankám levněji, protože velkou část nákladů nese původní firma vyvíjející tento systém.

Je také zřejmé, že v bankovním sektoru nemůžeme tolerovat jedinou bezpečnostní chybu. U většiny důležitých open-source aplikací existuje bezpečnostní tým, který je upozorněn na bezpečnostní chybu vždy jako první a teprve po vydání bezpečnostní záplaty je celá záležitost zveřejněna. V případě banky by však mohlo dojít k tomu, že by si útočník nechal celou věc pro sebe a klienty banky připravil o jejich úspory. Přenesme se na chvíli pryč od bankovního software.

Uvedme další příklad. Představme si, že naše konkurenční firma vyvíjející podobný systém zveřejnila část svých zdrojových kódů. Získá tak všechny výhody jmenované výše – rychlejší vývoj, větší přívaz uživatelů i programátorů pro její software atd. Z tohoto kroku však my těžít nemůžeme vůbec – zveřejněný kód nepotřebujeme, protože jsme si již napsali vlastní. Naopak přicházíme o velké množství uživatelů, konkurence bude vždy rychlejší a staneme se tak pro trh nezajímavými. Možná to není

až taková nevýhoda, spíše nešťastný důsledek pro některé dodavatele proprietárního software.

Ted trochu z praxe. Několik let užívám operační systém GNU/Linux. Vývoj je neuvěřitelně rychlý, spolupracuje na něm obrovské množství programátorů. S tím však souvisí problém, že rychlý vývoj s sebou nese značné množství chyb. Občas se tedy stane, že se aplikace nechová úplně spolehlivě. Někteří lidé dokonce open-source software odmítají užívat se slovy „nebudu jim dělat testera“ či „je to jeden velký bastl, nemá to koncepci“. Pravdou je, že žádný software není bez chyb. Ten, který řadíme do kategorie open-source, jich měl v minulosti opravdu mnoho. A stále jich i mnoho má. Dle mého názoru a především každodenních zkušeností se však situace neustále zlepšuje.

Před pěti lety, kdy byla open-source komunita o mnoho menší, bylo také méně uživatelů, kteří ony „testery“ dělali. Zároveň bylo i méně zainteresovaných programátorů. Většina software byla ve verzi 0.x, tedy menší než 1 (číslo 1 znamená první vydání, které je odladěné a s nejlepším svědomím programátory prohlášené za stabilní).

Dnes do open-source investuje (poskytuje pracovní sílu, finance, svůj zdrojový kód atp.) mnoho velkých firem – Intel, Google, Red Hat, Novell, Canonical, SUN Microsystems (odkoupený firmou Oracle), IBM a další. Kvalita open-source software tedy obrovskou měrou stoupla.

Otevřít či neotevřít

...toť otázka. Zmínil jsem výše, že někdy je z bezpečnostních důvodů lepší vyvíjet software pod pokličkou.

Existují však názory, že tím programátoři (nebo firmy vyvíjející software) jen dávají najevo, že jejich kód nestojí za nic, že je v něm mnoho chyb (včetně těch bezpečnostních) či že je napsán neúhledně.

Tento názor však lze vyvrátit jistě ospravedlnitelným a plně pochopitelným tvrzením, že je pro programátora důležité, aby řídil směr vývoje sám.

Otevřením kódu však lze získat konkurenční výhodu, jak jsem již popisoval výše.

Firmy si často „půjčují“ od open-source software a pak se rozhodnou svoji přidanou hodnotu zpětně vrátit. Proč to dělají? Asi těžko jako odměnu (přestože to tak většinou prezentují). Spíše dále očekávají, že se projekt stane zajímavější a opět přilákají nové programátory a urychlí tak vývoj.

Užívat či neužívat

Když Google stavěl svoje první servery, které měly indexovat dostupný internet, použil na nich open-source software – operační systém GNU/Linux, webový server Apache nebo Lighttpd, hodně jeho aplikací bylo napsáno v Javě, PHP či jiných open-source programovacích jazycích. Vzhledem k jeho současnému úspěchu si troufám tvrdit, že jeho orientace na open-source software je pro něj výhodná a naprosto zásadní.

Bohužel, v minulosti zkoušelo open-source software (například operační systém Ubuntu, který je založený na GNU/Linux) mnoho koncových uživatelů a programátorů, kteří měli problém s množstvím chyb v něm obsažených. To je samozřejmě odradilo a proprietární software je pro ně zárukou jistoty. Přestože se situace mnohonásobně zlepšila, jen velmi těžko je bude open-source software získávat na svou stranu.

Mnoho open-source software není kompatibilní s dostupným a všeobecně používaným proprietárním software. Nejčastěji je dáván příklad Adobe Photoshop⁶ a GNU/Linux. Pak je samozřejmě pochopitelné, že je uživatel

⁶ Je zajímavé pozorovat, jak tento příklad používají k argumentaci uživatelé-laici, kteří ve Photoshopu nevyužijí ani setinu jeho funkcionality nebo nemají cit pro grafiku

ochotný používat proprietární software, přestože je k tomu vlastně donucen.

Toto donucení je patrné také při výměně dokumentů – uživatel kancelářského balíku Microsoft Office nechápavě přijímá dokument ve formátu ODT vyprodukovaný balíkem OpenOffice.org a naopak.

Ještě patrnější je to pak při styku s veřejnou správou⁷, projekt datové schránky je tomu důkazem.

Příklad č. 1 – Android

Android je operační systém pro mobilní telefony a chytrá zařízení (PDA, navigace, smartphone atp.) vyvinutý společností Google, která zdrojové kódy předala sdružení firem Open Handset Alliance.

Android je založený na populárním systému Linux a aplikace je možné psát díky tzv. Android SDK (vývojové prostředí pro programátory) – tedy v rozšířeném jazyce Java.

Přestože je to operační systém založený teprve nedávno (5. listopadu 2007), těší se obrovské oblibě.

K dnešnímu dni (27. května) je podle společnosti AdMob 75 % všech mobilních zařízení v USA provozováno právě s Androidem. V Evropě a Asii je to o mnoho slabší (12 % každá oblast). Toto měření je třeba brát trochu s rezervou, nicméně o úspěchu Androida to rozhodně svědčí.

Podle jiných měření by měl být Android na mobilních telefonech v USA druhým nejpoužívanějším operačním systémem, hned po BlackBerry.

Po události Mobile World Congress z 12. února 2008, kde Google představil první prototypy, reagovaly společnosti HTC, Texas Instruments,

⁷ Ikdyž je to neuvěřitelné, orgány veřejné moci jsou v tomto směru opravdu zaostalé - <http://www.abclinuxu.cz/clanky/mfcr-bsa-open-source>

Intel, LG, Broadcom Corporation, Motorola, Nvidia, Qualcomm, Samsung, T-mobile a další obrovským zájmem.

Někteří výrobci hardware se dokonce vyjádřili, že budou v nejbližší době dodávat své produkty pouze s operačním systémem Android.

Android částečně využívá úspěchu Apple iPhone – nabízí „revoluční“ dotykové ovládání, poměrně přívětivé uživatelské rozhraní, obchod s aplikacemi (většina je open-source) a přístup k internetu v co největším možném měřítku. Odborníci se domnívají, že jsou to právě uzavřenost a značná omezení programátorů ze strany Apple, která umožnila společnosti Google dostat se tak vysoko a samotný úspěch iPhone předčít.

Příklad č. 2 – OpenSolaris

OpenSolaris je otevřený operační systém založený na systému Solaris společnosti SUN Microsystems (nyní součást společnosti Oracle). Nejedná se o linuxový systém, ale systém typu Unix.

Jedná se o obrovský projekt, který však nesklidil takový úspěch, jako GNU/Linux nebo Android. Byl ohlášen v roce 2004, v roce 2005 byla první část kódu uvolněna pod licencí CDDL a systém se tak začal postupně otevírat.

V roce 2007 SUN Microsystems najal Iana Murdocka, zakladatele jedné z nejpopulárnějších linuxových distribucí Debian, který měl pracovat na Projektu Indiana. Jeho úkolem bylo začlenit GNOME (uživatelské grafické prostředí), software z projektu GNU a balíčkovací systém do OpenSolarisu.

Jednou z nejlepších vlastností tohoto systému je souborový systém ZFS, který umožňuje vytvářet snapshoty (obrazy) souborového systému v čase.

V roce 2008 uzavřel SUN Microsystems kontrakt se společností Toshiba America Information Systems, která začala prodávat některé své notebooky předinstalované s operačním systémem OpenSolaris.

Bohužel, na rozdíl od Androidu, OpenSolaris vstupoval do vysoce koncentrovaného trhu – většinu instalací tzv. desktopového operačního systému tvoří stále MS Windows, Apple Mac OS X a dnes již velmi kvalitní GNU/Linux. OpenSolaris stále trpí „dětskými nemocemi“, které by se snad časem mohly vyléčit. Po odkupu SUN Microsystems společností Oracle však došlo k jisté stagnaci na projektu (Oracle investuje do GNU/Linux a další systém zkrátka nepotřebuje). Jeden čas to dokonce vypadalo, že se projekt zastaví úplně.

Jak to vypadá s projektem do budoucna, je velká otázka. Z projektu odešlo (Oracle propustil) několik důležitých vývojářů. Odkup SUNu Oraclem situaci ještě zhoršil a dle mého názoru tím OpenSolaris utrpěl zásadní ránu, ze které se bude jen těžko vzpamatovávat, jestli se mu to vůbec povede.

Závěr

Open-source má světu co nabídnout. Nese s sebou značné výhody, stejně tak i nepřehlédnutelné nevýhody a úskalí.

Google, jedna z největších technologických společností dneška, na open-source přístupu založila svůj úspěch. Těží z toho nejen ona, ale i mnoho dalších firem v řetězci, stejně tak jako uživatelé, ke kterým se dostává výborný a navíc otevřený software.

Na druhou stranu jsem na příkladě OpenSolarisu ukázal, že ne každý open-source projekt končí úspěchem a je třeba velmi důkladně zvážit, jakou strategii při výběru open-source přístupu vybrat a jestli se touto cestou vůbec vydat.

Tento článek byl určený převážně laikům, kteří nevěděli, jaký je rozdíl mezi otevřeným a uzavřeným, resp. svobodným a proprietárním software. Uznávám, že jsou některé části psány až příliš polopaticky a pro (byť jen trochu) zasvěceného člověka mohlo být čtení nudné. Věřím však, že se mi podařilo nastavit takovou úroveň, abych neodradil přílišnou odborností.

Děkuji za pozornost, těším se na případnou diskusi pod článkem na <http://michal.penka.name/open-source-vyhody-nevyhody-motivace-ucastniku-retezce>

Zdroje

- [1] AdMob Mobile Metrics. Our Insights on the Mobile Ecosystem [online]. April 2010 [cit. 2010-05-27]. April 2010 Mobile Metrics Report. Dostupné z WWW: <<http://metrics.admob.com/2010/05/april-2010-mobile-metrics-report>>.