

# PREČO OTVORENÝ SOFTVÉR ZATIAĽ NEZMENIL SVET

*Bude budúci rok rokom otvoreného softvéru, ktorý  
zmení stav informačných technológií k lepšiemu?*

*Michal Tomlein*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
michal.tomlein[zavináč]gmail[.]com

**Abstrakt.** *Otvorený softvér je dnes takmer všade. Je na väčšine serverov sveta a používajú ho milióny ľudí aj na svojich počítačoch. Dá sa ale predovšetkým takému otvorenému softvéru, ktorý pre svoju činnosť nepotrebuje grafické používateľské rozhranie, alebo je určený pre skúsených administrátorov a programátorov. To, čo však zatiaľ otvorený softvér svetu nepriniesol, je výraznejší zásah do spôsobu interakcie s počítačmi, výraznejší pokrok v použiteľnosti alebo inú inováciu, ktorá by navždy zmenila náš pohľad na výpočtovú techniku. Tieto vlastnosti sa nám spájajú skôr so zatvoreným softvérom, za ktorým stojí cieľavedomé vedenie s víziou, ako zlepšiť aktuálny stav. Stojí za ním tiež veľké množstvo dizajnerskej práce, pretože dizajn a funkcionálnosť nie sú oddeliteľné veci. V tejto eseji sa zaoberám otázkou do akej miery sa otvorenému softvéru podarilo posunúť stav informačných technológií dopredu, kde a prečo zaostal a pokúsím sa identifikovať príčiny tohto problému.*

**Kľúčové slová:** *kvalita, použiteľnosť, riadenie, otvorený softvér*

## Úvod

V dobe, kedy sa kvalita stáva prioritou v softvérovom procese a používatelia sa čoraz viac dožadujú použiteľnosti a prívetivosti softvéru [1], je zaujímavé sledovať, ako sa tieto vlastnosti darí alebo nedarí nadobúdať softvéru s otvoreným zdrojovým kódom.

V posledných rokoch aj vďaka rozmachu inteligentných mobilných telefónov a tabletov s dotykovým rozhraním a s nimi neodmysliteľne spojenými aplikáciami, je

## 2 Michal Tomlein

postupne stále viac cítit zvyšujúce sa očakávania a požiadavky používateľov na kvalitu, použiteľnosť a jednoduchosť ovládania aplikácií.

Ludia si stále viac uvedomujú, že softvér môže byť intuitívny, aj keď slúži na netriviálne úlohy. Toto je dôležitá zmena v myslení, ktorá už teraz mení softvérový priemysel.

Dokáže sa jej prispôbiť aj svet otvoreného softvéru?

### Rozdiel medzi bazárom a katedrálou

Kedysi som sa snažil v čo najväčšej miere nahradiť proprietárny softvér, ktorý som pri svojej práci na počítači potreboval, softvérom s otvoreným zdrojovým kódom. Dnes už takúto snahu vôbec nemám, dokonca mám často úplne opačný prístup.

Táto moja snaha mala svoje dôvody. Otvorený softvér ponúka výhody, ktoré sú pre pokročilých používateľov veľmi zaujímavé. Odstraňuje možno celkom neopodstatnený strach zo závislosti na dodávateľovi. Dáva pocit samostatnosti a pocit seberealizácie, ktoré často plynú z času, ktorý používateľ strávi tým, že si tento softvér spojzdni a prispôbi. Často je možné nastaviť každý aspekt jeho funkčnosti alebo rozhrania tak, ako to vyhovuje používateľovi.

V čom je teda problém? Tieto priority nie sú a nemali by byť prioritami väčšiny ľudí. Je nesprávne a nelogické, aby v 21. storočí bežní používatelia trávili čas inštaláciou a konfiguráciou softvéru.

Na to, aby sme odhalili príčinu značnej odlišnosti otvoreného softvéru z hľadiska kvality a použiteľnosti, musíme porozumieť tomu, ako funguje vývoj takéhoto softvéru.

Základným princípom otvoreného softvéru je, ako už vyplýva z názvu, otvorenosť zdrojového kódu, čo znamená prístup k zdrojovému kódu, možnosť jeho redistribúcie, úprav a znovupoužitia. Tieto práva sú vo väčšine prípadov spresnené v licencií, pod ktorou je tento softvér distribuovaný, a môžu byť v rôznej miere obmedzujúce. Typickým príkladom obmedzenia je nemožnosť použiť otvorený softvér v rámci komerčného projektu alebo povinnosť uverejniť vykonané modifikácie vo forme zdrojového kódu pod rovnakou licenciou, ako je to aj v prípade veľmi rozšírenej licencie GNU GPL.

Spôsobov vývoja otvoreného softvéru je viac, líšia sa medzi iným aj rôznou mierou riadenia. Dva hlavné modely vývoja otvoreného softvéru, ktoré opisuje Eric S. Raymond vo svojej knihe *The Cathedral and the Bazaar*, t.j. katedrálový a bazárový model, sú diametrálne rozdielne práve v miere centralizovaného riadenia, ktoré sa v nich vykonáva [2]. Miera riadenia je kľúčová, pretože definuje vlastnosti, ktoré s vysokou pravdepodobnosťou výsledný produkt nadobudne.

O aké vlastnosti ide? Mám na mysli predovšetkým použiteľnosť, zameranie na hlavnú úlohu, ktorú má softvér plniť, množstvo funkcionality a spôsob jej sprístupnenia v používateľskom rozhraní.

Katedrálový model sa vyznačuje kontrolovaným, riadeným a uzavretým spôsobom vývoja, ktorý na konci jednej iterácie produkuje verziu softvéru sprístupnenú vo forme zdrojového kódu verejnosti. Je to model, ktorým sa často vyvíja aj zatvorený, proprietárny softvér.

Bazárový model, ktorý sa za posledné desaťročie stal populárnym medzi otvorenými projektmi, bol spopularizovaný projektom Linux, ktorý v deväťdesiatych rokoch ako jeden

z prvých väčších projektov tento model aplikoval do praxe. Aj bazárový model samozrejme umožňuje kontrolovať a riadiť smerovanie projektu, v porovnaní s katedrálovým modelom však v menšej miere. Bazárový model totiž dáva možnosť zapojiť sa do vývoja takmer komukoľvek na svete.

V prípade Linuxu drží stráž nad projektom a jeho smerovaním zakladateľ Linus Torvalds a niekoľko ďalších ľudí, ktorým dôveruje. Torvalds to opisuje ako hierarchiu dôvery. Kým on akceptuje kód len od niekoľkých ľudí, ktorým dôveruje, títo ľudia akceptujú kód od ďalších ľudí, ktorým dôverujú oni. Kým sa do oficiálnej verzie Linuxu dostane zmena od nového prispievateľa, musí tak prejsť cez niekoľkých ľudí na rôznych úrovniach.

Distribúovaný verziovací systém Git, ktorý Torvalds vyvinul pôvodne pre potreby projektu Linux, tento prístup k vývoju priamo podporuje. Aj vďaka Gitu a projektom ako je portál GitHub sa začal bazárový model vývoja rozširovať medzi mnohé ďalšie otvorené projekty.

Mohlo by sa zdať, že bazárový model je jednoznačne lepší ako katedrálový. Nič však nie je čiernobiele.

## Prevrátené pyramídy

Bazárový model môže veľmi ľahko prehĺbiť problém, ktorý je charakteristický pre otvorený softvér: slabé riadenie.

Som presvedčený, že bez silného vedenia sa úspechy dajú dosiahnuť len šťastnou zhodou okolností. Softvér, za ktorým nestojí vízia alebo aspoň dobrá myšlienka, zväčša nie je softvér, ktorý by som rád používal.

Otvorený softvér však veľmi často žiadne silné vedenie za sebou nemá. Výsledky, ktoré otvorený projekt dosahuje, sú založené často na konfliktných víziách každého jedného prispievateľa. Každý v otvorenom projekte vidí svoj vysnívaný softvér a najradšej by ho celý prerobil podľa svojich predstáv.

Často sa žartuje, že keď sa stretnú traja právnici, v diskusii padnú najmenej štyri názory. Často si však neuvedomujeme, že to isté platí o všetkých ľuďoch, programátorov nevyímajúc. Programátori majú radi slobodu, a preto ich priťahujú otvorené projekty.

V tomto prístupe vidím zásadný problém. Produkuje softvér, ktorý je zľúčeninou čiastkových príspevkov ľudí rôznych názorov, ktorí namiesto toho, aby projekt ťahali jedným smerom podľa vízie jeho vedenia, vidia v projekte príležitosť, ako realizovať svoje predstavy.

Bazárový model im v tomto smere nebráni, ale priam pomáha. Je ako prevrátená pyramída: rozhodnutia, na čom sa bude pracovať, pochádzajú od jednotlivých programátorov. Tí pracujú na funkcionalite, ktorú od nich preberú programátori na vyššej vrstve, od nich tí na ešte vyššej vrstve, až kým sa nedostane úplne hore a nie je zaradená do hlavnej vetvy softvéru.

Samozrejme, zmeny, ktoré sú na prvý pohľad zlé, sa hore nedostanú. Dokonca sa hore nemusia dostať ani zmeny, ktoré nie sú úplne v súlade s cieľmi projektu. Veď na každej úrovni sú ľudia, ktorí za prebraté zmeny budú niesť zodpovednosť, a tak si možno dôkladne premyslia, či ich prijímú.

#### 4 Michal Tomlein

Nie je to však také jednoduché. Nie každé rozhodnutie je na prvý pohľad zlé. Projekty padajú skôr na veľkom množstve menších, nie celkom správnych rozhodnutí. Tie je ťažké spozorovať a ešte ťažšie zamietnuť.

Dobré vedenie musí vedieť, kedy povedať nie. Dokonca je veľakrát potrebné hovoriť „nie“ častejšie ako „áno“. Bazárový model a jeho prevrátená pyramída však na toto nemá dobré nástroje.

Výsledok je softvér, ktorý má stovky rôznych funkcií a nastavení, neadekvátne používateľské rozhranie a chýba mu zameranie na úlohu, ktorú by mal riešiť.

### **Rozdiel medzi programátorom a dizajnérom**

Bazárový model stavia programátorov až príliš často do úlohy dizajnérov. Programátori ale vo všeobecnosti nie sú dobrí dizajnéri. Myslia analyticky a nemajú problém používať takmer akékoľvek počítačové rozhranie. Nerozmyšľajú ako používatelia, a preto väčšinou nie sú vhodní kandidáti na tvorbu rozhraní.

Dokonca ani programátori, ktorých dizajn a použiteľnosť rozhraní trápia a zaujímajú, sa len ťažko dokážu vžiť do úlohy používateľa. To, čo sú schopní vyprodukovať, teda nedosahuje kvalít práce skutočných dizajnérov.

Nanešťastie dizajnérov je v otvorených projektoch málo. Dizajnéri nezdediajú zapálenie programátorov pre otvorený softvér. Nestotožňujú sa s predstavou, že by výsledok ich práce, ktorý často vnímajú ako svoje dielo, mohol niekto iný len tak zmeniť. Celkom logicky preto nájdeme dizajnérov skôr v komerčných, zatvorených projektoch, kde je ich práca lepšie ocenená.

### **Zatvorené katedrály**

Poviete si, že problémy, o ktorých hovorím, nie sú špecifické pre softvér, ktorý je vyvíjaný bazárovým modelom. A máte pravdu, naozaj nie sú. Rovnako nie sú špecifické ani pre otvorený softvér.

Bazárový model je však na rozdiel od katedrálového podľa môjho názoru viac náchylný k týmto problémom. Ako podloží toto tvrdenie?

Netreba sa ani príliš zamyslieť a napadne vám hneď niekoľko rôznych softvérových produktov, ktoré takpovediac zmenili svet informačných technológií. Skúste sa ale zamyslieť, či medzi nimi sú aj otvorené softvérové produkty.

Možno vás napadnú projekty ako GNU, Linux, Apache, FreeBSD, Mozilla, MySQL a iné. Tieto projekty dosiahli veľké úspechy, ale nie úspechy druhu, ktorý mám na mysli. Okrem projektu Mozilla, ktorého produkt je webový prehliadač, ide projekty, ktoré si našli uplatnenie skôr na serveroch a v IT sfére. Skúste sa ale spýtať priemerného Jožka, ktoré z týchto projektov pozná. Ak vo svojej odpovedi uvedie okrem projektu Mozilla aj Linux, pravdepodobne máte dočinenia s nadpriemerným Jožkom.

Samozrejme, tieto projekty mohli zmeniť svet informačných technológií aj bez toho, aby o nich priemerný Jožko vedel. A do istej miery aj zmenili. Posunuli ho však výrazne dopredu? Mozilla a Mozilla Firefox sú webové prehliadače. Hlavne Firefox dosiahol úspech v tom, že poskytoval lepšiu alternatívu k prehliadaču od Microsoftu, ktorý mal na trhu monopol a brzdil tak napredovanie webu. Firefox ako softvér však neprinesol

výraznejší pokrok voči tomu, čo už vtedy ponúkali iné alternatívne prehliadače. Linux je operačný systém, ktorý poskytuje podobnú funkcionálnosť ako väčšina ostatných na trhu, nakoniec ide o klon Unixu. Apache je webový server, ktorý sa veľmi rýchlo rozšíril na väčšinu serverov sveta. Dosiahol tak veľký úspech v liberalizácii serverovej technológie. FreeBSD je, podobne ako Linux, tiež len jeden z mnohých operačných systémov. MySQL je databázový systém, ktorého úspech bol pre svet prínosom v podobnom zmysle ako úspech Apache a FreeBSD.

Posunuli teda tieto projekty stav informačných technológií dopredu? Dá sa povedať, že z určitého pohľadu áno. Keďže na základe niektorých z nich boli vybudované ďalšie – aj zatvorené – projekty, ktoré možno dosiahli väčší prínos a ovplyvnili životy väčšieho množstva ľudí, môžeme usúdiť, že tieto projekty nepriamo posunuli svet IT dopredu o viac, ako by sa na prvý pohľad mohlo zdať.

Otázka však je, či by situácia bola iná, keby neexistovali. Keby nebol Linux, Apache a MySQL, existovali by dnes webové portály s miliónmi návštevníkov, ktoré sú na týchto technológiách postavené? Myslím si, že áno. Jednoducho by boli postavené na iných technológiách.

Z toho teda vyplýva, že tieto projekty, aj napriek nemalým úspechom, ktoré dosiahli, sú až príliš nahraditeľné. To nie je vlastnosť, ktorú si človek spojí s projektom, ktorý posunul svet výrazne vpred.

Pýtam sa, kde je otvorený projekt, ktorý urobil pre liberalizáciu informačných technológií to, čo Microsoft Windows a Microsoft Office? Kde je otvorený projekt, ktorý priniesol svetu takú zmenu v interakcii s výpočtovou technikou, ako prvý Macintosh alebo neskôr iPhone a iPad?

Každý rok sa môžeme dočítať, že ten budúci bude rokom otvoreného softvéru. Rokom, kedy otvorený softvér konečne dosiahne úspechy takéhoto kalibru. Neostáva iné, než čakať.

## Záver

Je viac ako pravdepodobné, že budúci rok nebude rokom otvoreného softvéru. Ba dokonca ani ten ďalší. Otvorenému softvéru v dosiahnutí tohto cieľa bráni kombinácia niekoľkých faktorov.

Niektoré z jeho problémov plynú z bazárového modelu vývoja, ktorý dominuje medzi otvorenými projektmi. Neposkytuje vedeniu projektu nástroje, ktoré by mu umožnili postaviť dizajn produktu na prvé miesto, čo si vyžaduje často hovoriť prispievateľom „nie“.

Iné problémy plynú z nedostatku dizajnérov, ktorí sú nahradzovaní programátormi. Programátori nie sú vhodní kandidáti na túto úlohu, čo v konečnom dôsledku spôsobuje nízku kvalitu používateľských rozhraní, slabú použiteľnosť a sústredenie sa na nesprávne aspekty celkového riešenia.

Kým sa tento stav nezmení, nemôžeme očakávať od otvoreného softvéru také výsledky, na aké sme zvyknutí z komerčných a zatvorených produktov a riešení.

## Použitá literatúra

1. Gaby, A.: *QA InfoTech Announces Top Trends in Software Quality for 2011*. PRWEB, 12. január 2011, URL: <http://www.prweb.com/releases/2011/01/prweb4959164.htm> [25. 11. 2011]
2. Raymond, E.S.: *The Cathedral and the Bazaar*. O'Reilly (1999), ISBN 1-56592-724-9.

## Annotation

### *Why Open Source Has Failed to Change the World*

*Open-source software can be found almost everywhere, including on most of the world's servers and the computers of millions of users. However, the most successful open-source projects are the ones that produce software without a graphical user interface or software whose intended audience are administrators and developers. What the open-source movement has failed to give the world so far is a major breakthrough in the way we use and interact with computers, a major advancement in usability or any other innovation that would forever change the way we look at technology. These are characteristics we generally associate with closed software supported by strong leadership with a clear vision of how a change for the better could be accomplished. This kind of software also requires an enormous amount of design work, which open-source projects are generally not known for. In this essay, I ponder the question to what extent open-source software has pushed information technology forward, conclude where it falls short and try to identify the reasons behind this problem.*