

NÁSTROJE, KTORÉ TVAROVALI SOFTVÉR KOMUNITY OTVORENÉHO ZDROJOVÉHO KÓDU

*Bola to práve komunita otvoreného zdrojového kódu,
ktorá najvýraznejšie prispela k vývoju verziovacích
nástrojov a súčasne ich aj najviac využívala.*

Milan Šoka

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
milan.soka[zavináč]chello[.]sk

Abstrakt. Otvorený softvér, ktorý vyvíjala komunita otvoreného zdrojového kódu, by sa nikdy nestal takým úspešným, keby sa súčasne s ním nezačali vyvíjať aj verziovacie nástroje. Boli to práve verziovacie nástroje, ktoré umožnili vyriešiť najväčšie problémy komunity otvoreného zdrojového kódu. A boli to práve verziovacie nástroje, ktorých vývoj najviac urýchlila komunita otvoreného zdrojového kódu. V tejto eseji vám poviem niečo o tom, ako vznikla myšlienka otvoreného softvéru, ako to súviselo s licenciou na otvorený softvér a ako mohol začať, hlavne vďaka verziovacím nástrojom, vývoj voľného softvéru tak ako ho poznáme dnes, akým je napríklad Linux. A ako sa mohli verziovacie systémy, hlavne vďaka podpore komunity otvoreného zdrojového kódu, rozvíjať ďalej do podoby v akej ich poznáme dnes.

Kľúčové slová: komunita otvoreného zdrojového kódu, verziovacie nástroje, nástroje pre podporu vývoja softvéru, voľný softvér

Komunita otvoreného zdrojového kódu

Komunita otvoreného zdrojového kódu nevznikla len tak z nejakej spoločenskej potreby, ale o jej zrode sa hovorí v súvislosti s tvrdohlavosťou jedného pána menom Richard

Stallman¹. Prišiel s myšlienkou, že zdrojový kód by sa mal zdieľať rovnako tak, ako vzduch v miestnosti. Ak niekto nejako zdokonalí svoj systém, potom sa očakáva, že sa podeli o svoje modifikácie so všetkými, ktorí majú podobný systém, tak aby každý z toho mohol mať úžitok. Pretože s takýmto prístupom nikto nič nestratil, všetci mohli iba získať. Neskôr Richard Stallman založil neziskovú organizáciu menom „Free Software Foundation“ a navrhol aj licenciu pre ňu, ktorá navždy zabezpečovala voľný prístup k zdrojovému kódu. Dodnes je známa ako GPL (General Public License) [1].

Stallmanova myšlienka otvoreného zdrojového kódu sa ujala a ľudia začali uvoľňovať svoje zdrojové kódy pod GPL. Toto vlastne umožnilo, že programátori z celého sveta získali prístup ku zdrojovým kódom svojich kolegov. Rovnako tak vďaka GPL mohli pracovať na mnohých projektoch, ktoré si sami vybrali a mohli sa pridať v ktorúkoľvek dobu vývoja projektu. Toto bol veľmi výrazný zlom v prístupe programovania ako takého. Nová komunita, ktorá zdieľala svoje zdrojové kódy a do ktorej sa mohol pridať úplne hocikto z ktorejkoľvek časti sveta, sa začala výrazne rozširovať vo svojich počtoch [1].

Obrovský prínos prvého voľného operačného systému

Toto ale netrvalo dlho a ľudia začali pochybovať o tom, že všetok softvér má byť voľne dostupný. V tom čase sa ale objavil chlapík, ktorý výraznejšie posunul veci dopredu smerom k otvorenému softvéru a myšlienke otvorených zdrojových kódov. Objavil sa Linus Torvalds² z Fínska, ktorý sa rozhodol nanovo vytvoriť celé Unix-ové jadro, ktoré nazval Linux. K svojmu dielu zverejnil aj zdrojové kódy a publikoval ho pod GPL licenciou. Takmer okamžite sa začali k nemu pridávať ďalší programátori, ktorí začali písať vylepšenia a podporné programy pre Linux a Linux mohol začať rásť.

Dôvodom, prečo mohol byť softvér s otvoreným zdrojovým kódom úspešnejší ako ten s uzavretým, veľmi pekne načrtol Eric Raymond vo svojej eseji z názvom Katedrála a Bazár (angl. The Cathedral and the Bazaar) [2], kde vysvetľuje na príklade porovnania uzavretého softvéru ako spôsobu fungovania Katedrály, a otvoreného softvéru ako fungovania bazáru. O takom bazáru kde sa skutočne môžu uplatniť rôzne vývojové techniky a štýly a tak isto o ďalšej funkcionalite môže rozhodnúť ktokoľvek, kým pri uzavretom softvérovom vývoji sú to práve biznis potreby konkrétnej spoločnosti, ktorá softvér vyvíja. Nie vždy sa pracuje na tých častiach softvéru, ktorú skutočne používatelia potrebujú najviac, alebo ktorá obsahuje najviac nedostatkov. Ďalej poukázal na to, že uzavretý vývoj softvéru má vždy svoje tvrdé ohraničenie, ktorým je hlavne rozpočet, termíny a marketingové stratégie, ktoré nijako nesúvisia s technickými problémami softvéru. Zatiaľ čo otvorený softvér má výhodu oproti uzavretému aj vzhľadom k nedostatkom alebo chybám v kóde. Linus Torvalds túto skutočnosť vysvetlil takto: „daný dostatočný počet očných buliev pohltí všetky chyby v kóde“. Tomu sa potom neskôr začalo zo žartu hovoriť Linusov Zákon.

Príchod takého operačného systému, akým bol Linux, bol skutočným darom aby sa mohli veci začať hýbať dopredu. Všetci, čo poznáte trochu Linux, viete, že je často schopný

1 Richard Stallman je považovaný za zakladateľa myšlienky otvoreného softvéru.

2 Linus Torvalds je považovaný za zakladateľa operačného systému Linux a dodnes vedie jeho vývoj.

fungovať výrazne efektívnejšie a bez nejakých jeho „pádov“. Linux otvoril možnosť a pridal inšpiráciu ľuďom k písaniu programov, ktoré by na ňom bežali a aj k tomu aby programátori zverejňovali svoj zdrojový kód. Pretože so zverejnením kódu sa projekt mohol naplno rozširovať a takto sa mohla zvýšiť jeho kvalita a vôbec zachovať jeho prežitie medzi podobnými programami ako takými.

Problémy okolo zdrojového kódu

Tak, ako sa začala rozširovať komunita otvoreného zdrojového kódu, tak začal rásť aj objem kódu, s ktorým museli jednotlivé projekty pracovať. Kým voľakedy ľudia zverejňovali svoj zdrojový kód na internetových stránkach, teraz toto už bol veľký problém, pretože zdrojového kódu bolo priveľa. V jednotlivých projektoch už nebolo toho, kto by sa rozhodol zobrať si na plecia celú logistiku okolo zdrojových kódov. Presnejšie sťahovanie zdrojových kódov zo stránok prispievajúcich programátorov, vkladanie modifikácii do súčasného projektu a súčasne s tým ešte sledovať nedostatky, ktoré by nová modifikácia mohla priniesť. Nie, toto už ďalej nebolo možné, aby toto niekto robil sám. Zvlášť pokiaľ hovoríme o komunite otvoreného zdrojového kódu, ktorej členmi sú nadšenci a vo svojej podstate dobrovoľníci, ktorí chcú hlavne vylepšovať existujúci kód svojimi vlastnými vylepšeniami.

Projekty s otvoreným zdrojovým kódom museli začať fungovať inak aj v oblasti zverejňovania svojich zdrojových kódov. Kým dovtedy sa zverejňovali iba zdrojové kódy stabilnej a dobre otestovanej verzie, teraz museli ísť ďalej. Pretože pre prispievajúcich vývojárov bolo často veľmi frustrujúce, ak si stiahli uvoľnenú verziu zdrojových kódov, začali hľadať chyby v zdrojovom kóde a potom nakoniec keď chybu našli a poslali ju späť do projektu zistili, že danú chybu už pred časom našli a aj opravili. Preto sa projekty s otvoreným zdrojovým kódom rozhodli najprv pre čiastočné riešenie tohto problému. Každý deň zverejňovali na svojich stránkach aktuálny stav celého projektu. Čo bolo určite lepšie, pretože ľudia, ktorí chceli prispieť, mohli mať prehľad o tom, aká funkcionálnosť sa práve vyvíja. Na druhú stranu to bolo ale veľmi ťažkopádne riešenie. Stačilo, že sa zmenil jeden riadok v ktoromkoľvek súbore a už bol dôvod dávať znova celý, len máličko pozmenený projekt na stiahnutie.

Vplyv verziovacích systémov

Úplným riešením problému okolo zdrojového kódu, ktoré je aktuálne aj dnes, je použitie verziovacieho systému. Verziovací systém, tak ako ho poznáme dnes, je poloautomatizovaný nástroj na správu verzii. Na jednej strane vývojárom, ktorí majú registrovaný účet dovoľuje sťahovať a odosielať svoje opravy a vylepšenia zdrojového kódu, ale aj iných súčastí projektu. Na druhej strane, hocikto si môže anonymne stiahnuť aktuálny stav zdrojových kódov, takže tak či tak, zdrojový kód zostáva dostupný pre každého. Ovládanie verziovacieho systému bolo už v jeho začiatkoch veľmi jednoduché z hľadiska princípu. Zvyčajne stačil jeden príkaz na to, aby ste aktualizovali svoje zdrojové kódy projektu, ale tiež stačil jeden príkaz na to, aby ste si ich mohli stiahnuť všetky nanovo. Odovzdávanie zmien súborov už nemuselo byť také jednoduché, ale aj toto sa s vývojom verziovacích systémov postupne zlepšovalo.

Príchod verziovacích systémov otvoril znova, tak ako pred ním príchod myšlienky otvoreného softvéru (GNU), otvorenej licencie na softvér (GPL) a prvého otvoreného operačného systému (Linux), nové možnosti. Projekty otvoreného zdrojového kódu začali veľmi rýchlo prechádzať na použitie verziovacích systémov. Vývojári teraz dostali skutočne silný nástroj ako podstatne efektívnejšie pracovať na svojich projektoch s otvoreným zdrojovým kódom.

Myslím si, že čím efektívnejšie dokážete pracovať na veciach a to teraz nemyslím len vývoj softvéru, ale všeobecne, tak tým viac sa do vecí môžete zahĺbiť a viac sa do nich ponoriť. Čím viac sa viete do svojej práce zahĺbiť, tým väčší vzťah si k nej viete vytvoriť. Čím väčší vzťah a lásku máte k svojej práci, tým sa vám stáva príjemnejšia a tým viac máte chuť v nej pokračovať znova aj nabadúce. A nie len to, ale získate aj motiváciu povedať o svojich úspechoch, deliť sa o skúsenosti s ostatnými a inšpirovať ďalších aby skúsili fungovať tak ako vy sami fungujete. Tento proces je podľa mňa samo-naplnujúci a samo-rozširujúci. Možno aj kvôli tomuto sa stali verziovacie nástroje také úspešné. Možno aj kvôli tomuto a verziovacím nástrojom sa stal Linux taký úspešný a vôbec otvorený softvér ako taký.

Vývoj verziovacích systémov

Keď sa pozrieme na vývoj verziovacích systémov, môžeme s istotou povedať, že ho výrazne poznačil otvorený softvér a všetko, čo sa udialo už od vzniku myšlienky otvoreného softvéru Richarda Stallmana, cez vznik otvorenej licencie na softvér (GPL), až po vznik prvého otvoreného operačného systému Linux. Dokonca aj v súčasnosti je vzájomné prepojenie medzi komunitou otvoreného zdrojového kódu a verziovacími systémami veľmi veľké. Navzájom jeden druhého potrebujú a vylepšujú. Možno hovoriť až o symbióze.

Keď prvé verziovacie systémy začínali, išlo prevažne o centralizované systémy. Centralizované systémy s jedným hlavným verziovacím skladovým miestom pre kód. Toto riešenie bolo jednoduché a efektívne. Hlavne pokiaľ išlo o menšie projekty. Takéto centralizované verziovacie systémy boli napríklad Subversion a CVS. Povedal by som, že hnacou silou ich vzniku bolo hlavne to, aby sa vôbec nejaké verziovanie a teda zlepšenie práce so zdrojovým kódom mohlo dať urobiť.

Mohli by však nástroje ako je Subversion, alebo CVS dobre poslúžiť k vývoju napríklad Linuxového jadra, ktoré má veľa vetiev a vyvíja ho množstvo ľudí z celého sveta? Určite by to bolo prinajmenšom priveľmi náročné na správu takého verziovacieho skladu. Preto v takých prípadoch je dobré siahnuť radšej po nástroji, ktorý sa už implicitne venuje decentralizovanému prístupu.

Prechod z centralizovaných na decentralizované verziovacie systémy

Takéto nástroje v komunite otvoreného zdrojového kódu existovali. Jedným z nich bol aj GNU arch. Schopnosť tohto nástroja potvrdil aj David A. Wheeler, ktorý o ňom povedal:

„GNU arch je plne decentralizovaný, čo ho robí veľmi vhodným pre prácu decentralizovaného vývoja, akým je napríklad vývoj Linuxového jadra.“ [3]

David A. Wheeler ďalej vo svojom článku hovorí o fakte, že GNU arch má hlavnú nevýhodu, ktorá ho stále odsúva do druhej kategórie verziovacích systémov a tou

nevýhodou je, že nefunguje dobre na Windows systémoch. Dokonca ani tvorcovia GNU arch už od začiatku nemali záujem na tom, aby sa toto niekedy zlepšilo. Navrhovali svoj verziovací systém tak, aby fungoval hlavne na „POSIX kompatibilných“ operačných systémoch. Keďže operačný systém Windows takým nie je, tak ani fungovanie ich verziovacieho systému na ňom nie je také, aké je na iných operačných systémoch, ktoré sú „POSIX kompatibilné“, ako je napríklad Linux.

Ak by sme sa pozreli na GNU Arch zo všeobecného pohľadu, uvidíme, že vo všeobecnosti nie je takým vyspelým nástrojom ako je Subversion. Túto vyspelosť je vidieť hlavne v otázke súborového systému, optimalizácii, ktorá sa musí niekedy robiť ručne, ale tiež aj v dokumentácii [3].

K všetkým týmto nedostatkom zrejme prispel fakt, že išlo o mladý projekt, ktorý sa nevyvíjal tak dlho ako napríklad spomínaný Subversion. Napriek tomu ale jeho samotný koncept bol veľmi flexibilný. Medzi jeho hlavné výhody patrí určite jednoduchosť, otvorený návrh a silná schopnosť spracovať náročné situácie.

Keď sa pozreli na GNU Arch vývojári, videli v jeho návrhu dobré ideály, avšak už nie v jeho implementácii. Dôsledkom toho bol vznik jeho nástupcov, akými boli napríklad Arx, Bazaar a neskôr potom aj Bazaar-NG. Pozrime sa ďalej na jedného z jeho nasledovníkov, na Bazaar-NG.

Bazaar-NG vychádzal z GNU Arch, avšak je to v zásade nový projekt. Je to distribuovaný distribuovaný verziovací systém. Mal lepšie používateľské rozhranie ako jeho predchodca. Zlepšenie nastalo aj v dokumentácii, ktorú hlavný vývojár Bazaar-NG píše súčasne s kódom. Čo je rozdiel oproti GNU Arch, ktorý dobrú dokumentáciu potreboval ako soľ, pretože jeho operácie boli priveľmi komplikované a mnoho krát ťažkopádne.

Ďalšou veľkou výhodou oproti GNU Arch je v tom, že Bazaar-NG je vyvíjaný v Python, čo mu umožňuje sa veľmi ľahko preniesť na iné systémy. Aj keď Python je skriptovací jazyk a niekto by mohol povedať, že takáto skutočnosť spôsobí pomalšie fungovanie systému. To je však vecou iba vnímania. Aj v skriptovacom jazyku ako je Python je možné optimalizovať na rýchlosť spracovania.

Je ešte niekoľko ďalších distribuovaných verziovacích systémov komunity otvoreného zdrojového kódu, ktoré zasiahli a stále zasahujú do vývoja otvoreného softvéru.

Zhrnutie

Myšlienka otvoreného zdrojového kódu, voľnej licencie ku kódu (GPL) a voľný operačný systém Linux fungujú v dnešnej dobe aj naďalej a tešia sa stále väčšiemu úspechu spolu s verziovacími nástrojmi, ktoré tvoria ich stálu a neodmysliteľnú súčasť. Všetky tieto veci vytvorili jeden veľký celok na ktorom budú môcť stavať ďalšie generácie mysliteľov a programátorov komunity otvoreného zdrojového kódu. Všetci z nich určite nikdy nepoprú významnú úlohu, ktorú zohrali verziovacie nástroje vo vývoji komunity otvoreného zdrojového kódu. Mnohí z nich budú možno práve pokračovať v ich vývoji, ktorý sa momentálne nachádza na distribuovaných verziovacích systémoch a posunú tento vývoj ešte o kúsok ďalej, tak aby sa ľudia mohli aj naďalej tešiť z kvalitného otvoreného softvéru, ktorý pomaly ale isto sa začína posúvať viac do povedomia celého sveta.

Použitá literatura

1. Moshe Bar and Karl Fogel: Open Source Development with CVS, 3rd edition, ISBN: 1-932111-81-6, (2003)
2. Eric Steven Raymond: The Cathedral and the Bazaar, (1999), dostupné online: <http://catb.org/~esr/writings/homesteading/cathedral-bazaar/>
3. David A. Wheeler: Comments on Open Source Software / Free Software (OSS/FS) Software Configuration Management (SCM) / Revision-Control Systems, (2005), dostupné online: <http://www.dwheeler.com/essays/scm.html>

Annotation

Tools that shaped the open source software

The Version Control systems have experienced significant increase over the last few years. It was thanks to open source, why Version Control systems has become more developed. And it was thanks to the Version Control systems, why open source can get to where it is today. So we can say that the Version Control systems shaped the development of open source in highly significant manner.