

Uporabnost in zanesljivost pri razvoju in uporabi odprte kode

Primož Zevnik

Gimnazija Kranj, Koroška cesta 13, 4000 Kranj
primoz.zevnik@guest.arnes.si

V okviru odprte kode je nastalo več vrst programske opreme. Večino te opreme uporabljajo izkušeni uporabniki. Čeprav uporaba prostega programja narašča, ima povprečen uporabnik računalnika stik le z lastniško zaščiteno programsko opremo. Razlogov za takšno stanje je več; eden izmed njih je prepričanje, da so programi odprte kode manj uporabni. V tej razpravi proučujem, kako razvojni proces odprte kode vpliva na uporabnost, in predlagam metode izboljšanja uporabnosti, ki so primerne za razvoj tovrstne programske opreme na Internetu.

Ključne besede: Odprta koda, zanesljivost, problem uporabnosti, razvoj, izboljšave, skupnost.

Open Source Usability and Reliability in Development and Usage

Open source communities have successfully developed many pieces of software. Most of this software is used by technically sophisticated users, in software development or as part of the larger computing infrastructure. Although the use of open source software is growing, the average user computer user only directly interacts with proprietary software. There are many reasons for this situation; one of which is the perception that open source software is less usable. This paper examines how the open source development process influences usability and suggests usability improvement methods that are appropriate for community-based software development on the Internet.

Keywords: Open Source, Reliability, Usage problems, Development, Improvements, Community.

1. ODNOS MED ODPRTO KODO IN UPORABNIKI

Odprta koda si je pridobila sloves zanesljivosti, učinkovitosti in funkcionalnosti, ki je presenetil številne na področju razvijanja programske opreme. Internet je omogočil koordinacijo prostovoljnih razvijalcev po svetu, da so izdelali odprtokodne rešitve, ki so na svojem področju vodilne na trgu (npr. strežnik Apache). Vendar pa je večina uporabnikov teh aplikacij tehnično usposobljenih, povprečen uporabnik računalnika pa uporablja standardno komercialno lastniško programsko opremo. Za takšno stanje obstaja več razlag: počasno reagiranje, interoperabilnost, podpora uporabnikom, organizacijske odločitve glede nakupa ipd. V tej razpravi me zanima zlasti ena od možnih razlag, in sicer da je (za večino potencialnih uporabnikov) odprtokodno programje manj uporabno.

Uporabnost načeloma ocenjujemo s petimi značilnostmi: lahka priučljivost, učinkovita uporaba, čim lažje pomnjenje, pogostost in resnost napak ter osebno zadovoljstvo. Uporabnost je ločena od zmogljivosti programske opreme (t.j. da lahko opravi določeno nalogo) in drugih značilnosti, kot sta zanesljivost in stroški. Problematična pa ni uporabnost programske opreme, ki jo uporabljajo razvijalci; v pričujoči razpravi se bomo osredotočili na programsko opremo, ki je namenjena predvsem povprečnemu uporabniku (kot so urejevalniki besedil, elektronske pošte in internetni brskalniki).

Dejstvo, da pri odprtokodnih programih obstajajo problemi glede uporabnosti, ni pomembno samo po sebi; pri vsaki interaktivni programski opremi so težave. Problem je naslednji: kakšna je programska oprema, ki

nastane v postopku razvoja odprte kode, v primerjavi z drugimi pristopi? Žal ni lahko izvesti kontroliranega preizkusa, v katerem bi primerjali alternativne postopke razvoja; vendar pa je možno primerjati podobne naloge na obstoječih programih programske opreme, izdelane v različnih razvojnih okoljih.

Med dvema programoma obstaja več razlik, ki lahko vplivajo na take primerjave, npr. čas razvijanja, zrelost programske opreme, predhodni obstoj podobne programske opreme ipd. Nekateri od teh dejavnikov so značilni za razlike med odprtokodnim in komercialnim razvojem, vendar je zaradi številnih razlik težko reči, kakšna naj bi bila »pravična primerjava«. Preizkusni kamen mora biti navsezadnje uporabniško testiranje programske opreme, vendar pa, kot kaže projekt Mozille, lahko traja več let, da odprtokodni projekt postane primerjalni, zato predhodnih negativnih primerjav ne smemo jemati kot indikativne za ves primerjalni pristop. Poleg tega javna narava razvoja odprte kode omogoča vidnost začetnih verzij, medtem ko je distribucija zasnov komercialne programske opreme načeloma omejena.

Če bi bil razvoj aplikacij in namizja prvenstveno tehnični problem, bi bil rezultat nedvoumen. Vendar pa ni tako; gre za problem ergonomskega oblikovanja in psihologije vmesnika, hekerji pa so na tem področju že od nekdaj šibki. Medtem ko so zelo dobri pri oblikovanju vmesnikov za druge hekerje, pa so prešibki pri oblikovanju miselnih procesov za ostalih 95% prebivalstva, da bi lahko napisali vmesnike, ki bi jih vsi uporabniki kupili in plačali.

Po tradiciji so uporabniki odprtokodne programske opreme strokovnjaki, ki so to opremo zgodaj sprejeli in so skorajda sinonim za razvijalce. Z vstopom odprtokodne programske opreme v komercialni svet se poudarja uporabnost in oblikovanje vmesnika, kar počneta Caldera in Corel distribuciji, ko eksplicitno posvečata glavnemu namizju največjo pozornost. Verjetnost, da bi nestrokovnjake pritegnila razpoložljiva koda je majhna, večja pa, da bi izbrali izdelke odprtokodnih programov na podlagi stroškov, kvalitete, znamke in podpore.

2. SKUPNOST ODPRTE KODE PROTI SPLOŠNIM UPORABNIKOM, KOT JO VIDI JAMES TURNER

Skupnost odprte kode ima nedvomno veliko potenciala. Poleg velikega števila razvijalcev ima gibanje za odprto kodo tudi strast in predanost svojemu delu, ki pri proizvajalcih komercialne programske opreme pogosto zbuja zavist. Vendar pa je včasih skupnost odprte kode sama sebi največji sovražnik. V nadaljevanju navajam nekaj razlogov, zakaj je temu tako.

A) Preveč razvijalcev se ubada z istim problemom

Razvijalci odprte kode prihajajo do novih idej, ker morajo razrešiti določen problem. Z drugimi besedami, do nove aplikacije so prišli zaradi neke potrebe. Težava je v tem, da se neredko pojavijo dva ali trije (ali celo več!) različnih paketov za isto stvar. Do tega je prišlo npr. pri zvočnih sistemih za Linux, kjer zdaj obstaja več konkurenčnih paketov, ki jih morajo podpirati različne distribucije. Ali pa na področju Java, kjer je še v zadnjem času nastal projekt proste Java za aplikacije, ki so bile odvisne od tega pomembnega Sunovega segmenta, ki pa ni bil prost.

Malo konkurenčnosti ne škoduje. Navsezadnje predstavlja Linux konkurenčno vizijo za operacijski sistem. Če pa obstaja preveč konkurenčnih vizij, iz katerih se v kratkem času ne izloči manjše število opcij, nastane zmešnjava konfliktnih standardov, uporabniška skupnost pa mora namestiti preobilje različnih paketov, ki vsi delajo isto stvar.

Odličen primer tega sindroma je preveliko število distribucij Linuxa. Nobenega razloga ni, da bi morale obstajati več kot dve ali tri distribucije. Ker vsaka deluje malce drugače, smo naleteli na težavo, da so aplikacije in gonilniki redko na voljo v binarni obliki, ker je treba podpirati preveč verzij prevelikega števila izdaj Linuxa.

Razvijalci aplikacije bi morali priskrbeti od 5 do 10 različnih binarnih namestitev, po eno za vsako distribucijo. Pomnožite to s petimi ali več aktivnimi izdajami distribucije, ki je lahko v obtoku, in videli

boste, zakaj je tako veliko paketov na voljo le kot vir (zlasti najnovejše izdaje paketov, ki še niso bile vključene v distribucije Linuxa).

B) RAZVIJALCI ODPRTE KODE LJUBIJO DOBER NOTRANJI KONFLIKT

BSD vs. Linux. Gnome vs KDE. Debian vs Red Hat. Za vsako zanimivo tehnologijo odprte kode obstajata dva zagrizena tabora, ki porabita prav toliko časa za udrihanje drug po drugem, kolikor za izboljšanje svojih proizvodov.

Težko si je predstavljati, koliko boljša bi bila odprtokodna programska oprema, če bi te skupine sodelovale in združile napore. Na žalost je slaba stran osebne predanosti projektom ta, da jih ljudje uporabijo kot merilo lastne vrednosti, zato tekmeci zelo težko priznajo dosežke svojih nasprotnikov.

C) RAZVIJALCI ODPRTE KODE SE POGOSTO UKVARJAJO Z NAPAČNIM PROBLEMOM

Težava pri komercialnem razvoju je, da razvijalci ponavadi niso uporabniki svojih proizvodov in zato ne občutijo bolečine ob njihovih napakah. Težava pri razvoju odprte kode pa je, da skupnost razvijalcev pogosto ne odpravi problemov ali razvije novosti, če zanje niso neposredno zanimivi.

Ponavadi to ni problematično, saj razvijalci (kot uporabniki) naletijo na enake težave kot njihova uporabniška baza. Žal pa se razvijalci odprte kode od splošne uporabniške baze razlikujejo po tem, da imajo bistveno več tehničnega znanja. To pomeni, da so pripravljeni na veliko višjo stopnjo tehnične veščine pri uporabi določene stvari, kot pa netehnični uporabniki.

Primer: razvijalcu odprte kode se ne zdi nič posebnega, če se od uporabnikov zahteva, naj ustvarijo in konfigurirajo dokument XML, da bo nekaj delovalo, medtem ko bi končni uporabnik potreboval čarovnika za delo.

D) V SKUPNOSTI ODPRTE KODE SI »ZA NAS ALI PROTI NAM«

Skupnost odprte kode se običajno pritožuje, da prodajalci lastniške programske opreme uporabljajo legalna sredstva za utišanje kritik. Vendar zna biti skupnost odprte kode prav tako neprizanesljiva do notranjih kritikov. Poskusi poudarjanja napak ali možnosti izboljšav ponavadi vodijo k obrambnemu zanikanju, osebnim napadom kritikov ali popolnemu zavračanju tehtnosti problema.

E) SKUPNOST ODPRTE KODE IMA ZA VRATOM MICROSOFT

Čeprav je zadnje čase priljubljena tarča tudi SCO, že samo omemba MS deluje kot rdeča ruta na bika. Resničnost je pač taka, da ima Microsoft levji delež v svetu namiznih aplikacij in baznih postaj. Če je prva stvar, ki jo rečeš lastnikom Windowsov, da so nevedni, to ni dobra osnova, da se jih lahko spremeni.

Če nam je všeč ali ne, uporabniki Windowsov morda ne marajo strašljivega modrega zaslona smrti in Microsoftovih cen ter licenc, vendar pa znajo Windowse uporabljati in običajno dobijo aplikacije in strojno podporo, ki jo zanje potrebujejo. Linux ima celo vrsto orodij, zaradi katerih bi se mnogi obrnili k Linuxu, vendar bo potrebna večja prepričljivost, da se bodo za to dejansko določili. Povedati bo treba, *zakaj je Linux boljši* (in zanje tudi res mora biti boljši), ne le *zakaj so Windowsi zanič*.

Še zlasti je slab odnos »koga briga X«, pri čemer je X lahko strojna oprema brez podpre, neobstoječe igre, komplicirana multimedijaska podpora ali karkoli drugega, pri čemer ima Linux težave ali naj bi jih imel. Če nekdo želi početi nekaj drugega kot ti, to še ne pomeni, da je to manj pomembno. Ko sem se pritoževal, da imam težave s povezavo DVD predvajalnika na prenosni računalnik, da bi gledal komercialne filme, sem poslušal komentarje kot npr. »zakaj pa gledaš DVD-je na prenosnem računalniku?!« Nekateri so mi celo predlagali, naj kupim prenosni DVD predvajalnik. Če pustim ob strani, koliko težav bi imel s prenašanjem še

enega kosa elektronike čez carino, kadar bi hotel gledati filme na letalu, so takšni komentarji najbolj nesmiselni, ker temeljijo na sledečem mišljenju: če Linux trenutno ne omogoča nečesa tako kot Windowsi, napadimo potrebo po tem, da bi to sploh počeli.

Zaključim lahko, da največja težava skupnosti odprte kode pri prenosu odprte kode na višjo raven ni kakšen pravni izziv ali Microsoftova tržna kampanja, temveč nezrelost in negotovost nekaterih članov skupnosti.

3. RAZVOJ PROTI UPORABNOSTI

Skupnost prostega programja je v ta namen razvila več orodij in tehnik: metode pregledovanja uporabnosti, vodnike za vmesnik, metode testiranja, interdisciplinarne ekipe ipd. Glede na to, da je skupina odprte kode vse bolj pozorna na uporabnost, je videti, da prehaja skozi enako fazo kot lastniško zaščitena programska oprema v 80. letih prejšnjega stoletja.

Ker so uporabniki programske opreme vse bolj raznoliki in manj tehnično podkovani, so proizvajalci programske opreme začeli uporabljati bolj uporabniško naravnane metode, da bi njihove proizvode uspešno prevzeli novi uporabniki. Medtem ko ima veliko uporabnikov še naprej težave s programskimi aplikacijami, so strokovnjaki, ki so jih zaposlila podjetja, precej izboljšali izkušnje uporabnikov.

Ko se bo baza uporabnikov odprtokodne programske opreme razširila na številne nerazvijalce, bodo morali projekti vsebovati uporabniku prijazne tehnologije, da bi programsko opremo ta uporabljal. Videti je, da nekateri odprtokodni projekti v zadnjem času prevzemajo tehnike iz predhodno lastniško zavarovanega dela, kot so eksplicitne smernice za uporabniški vmesnik za razvijalce aplikacij.

Težko je dati končni odgovor na vprašanje, ali obstaja problem uporabnosti odprte kode. Mnenja številnih komentatorjev in dejavnosti podjetij, kot je vpletenost Suna v GNOME, nakazujejo, da problem obstaja, čeprav znanstvena literatura molči. Da bi lahko predlagali pristope, ki zadevajo praktične in socialne značilnosti razvijalcev odprte kode (ter uporabnikov), je potrebno proučiti aspekte razvojnega procesa, ki bi lahko imeli negativen vpliv na uporabnost.

4 PROBLEMI UPORABNOSTI PROTI PROBLEMU FUNKCIONALNOSTI

Probleme funkcionalnosti je lažje določiti, oceniti in modularizirati kot probleme uporabnosti. To so vsi atributi, ki poenostavljajo decentralizirano reševanje problemov. Nekatere (a ne vse) probleme uporabnosti je precej težje opisati. Odpravljanje problema lahko zahteva natančen pregled celotnega vmesnika – kar seveda ni majhen prispevek k neprestanemu oblikovanju. Če bi v oblikovanje vmesnika vključili več kot enega oblikovalca, še zlasti če bi delali samostojno, bi to vodilo k nekonsistentnosti oblikovanja in s tem manjši splošni uporabnosti.

Modularnost projektov odprtokodne programske opreme prispeva k učinkovitemu pristopu, ker jim omogoča oddaljitev od Brooksovega zakona. Različen dele je mogoče zamenjati in nadomestiti z višjimi moduli, ki se jih vgradi v naslednjo verzijo. Vendar pa je glavni kriterij za uspeh pri uporabnosti konsistenca oblikovanja. Manjše variacije vmesnika med moduli in različne verzije modulov so lahko moteče in povzročajo zmedo, s čimer pokvarijo celotno izkušnjo uporabnika.

Priznati moramo, da projekti odprtokodne programske opreme uspešno rešujejo določene kategorije problemov uporabnosti. Eden od priljubljenih pristopov k oblikovanju vmesnika OSS je vzpostavitev različnih uporabniških frontendov: izmeničnih vmesniških ravni, ki dramatično vplivajo na celoten izgled aplikacije, vendarle malo spreminjajo naravo osnovne interakcijske dinamike. Podoben pristop je internacionalizacija programske opreme, kjer je jezik vmesnika (in kulturno specifičnih ikon) preveden. Oba pristopa sta primerna za modularen pristop k odprtokodni programski opremi, medtem ko poskus reševanja globljih interakcijskih problemov s preoblikovanjem ne vodi tako enostavno v obvladljiv projekt. Razlog za razliko je, da reševanje globljih problemov interakcije vpliva ne le na celoten vmesnik, temveč povzroča tudi spremembe različnih elementov funkcionalnosti.

Naslednja velika kategorija, ki zadeva uspešnost pri uporabnosti odprtokodne programske opreme, je namestitev programske opreme (predvsem GNU/Linux). Celo tehnično podkovani so imeli težave pri

nameščanju začetnih verzij GNU/Linuxa. Projekt Debian je nastal zato, da bi izboljšali distribucijo, ki je olajšala namestitve, in podjetja so ta trend nadaljevala. Taki projekti rešujejo problem uporabnosti, vendar na način, ki je v skladu s tradicionalnim razvojem odprtokodne programske opreme. Niz ročnih operacij je avtomatiziran, tako da za končnega uporabnika nastane črna skrinjica, ki je ne želi raziskovati. Ker je to odprt projekt, črno skrinjico seveda lahko odpre, preišče in spremeni vsak, ki ima za to voljo in znanje.

5. ZAKLJUČEK

Iz tega, kar imamo, moramo torej potegniti najboljše. Imamo razvojni model, ki je zmožen izdelata najboljšo programsko opremo na svetu, in skupnost odličnih, zagnanih in nesebičnih razvijalcev. Dolžni smo jim zagotoviti, da bo njihovo delo ostalo svobodno. Medtem ko čakamo na ustrezno zakonodajo, je dober prvi korak, da temeljito premislimo o obliki licence, ki bo izbrana za projekt programske opreme, in o tem, kdo bo lastnik avtorskih pravic.

LITERATURA

[1] <http://www.opensource.org/docs/definition.php>.

[2] <http://www.gnu.org/licenses/why-gfdl.html>; <http://www.gnu.org/licenses/fdl.txt>.

[3] <http://www.gnome.org/about/>.

[4] <http://www.kde.org/whatiskde/>.

[5] <http://www.linuxworld.com.au/index.php/id;1226331430;fp;4;fpid;2>.

***Primož Zevnik** je profesor slovenskega jezika in knjižničar na Gimnaziji v Kranju. Je avtor številnih razprav s področja literarne teorije in računalniške stroke. Na Gimnaziji v Kranju je vpeljal projekt odprtokodnega programja pri uporabi informacijskih tehnologij. Predava na seminarjih po Sloveniji in tujini.*