

# PROBLÉMOVÉ TVÁRE KOMUNIKÁCIE V OPEN SOURCE PROJEKTOCH

*V dnešnej dobe sa už ľudia nerozprávajú,  
komunikujú.*

*Martin Geier*

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

**Abstrakt.** *Komunikácia medzi ľuďmi stojacich tvárou v tvár je zložitá a náročná, no zlepšujeme sa v nej každým dňom. Je na nej postavený úspech jednotlivcov i spoločnosti a tiež úspech ich projektov. Ako však úspešne komunikovať pri tvorbe projektu, ak sme od seba vzdialení stovky kilometrov a pritom nás delia rôzne jazyky a zvyky? V eseji opisujem výber komunikačného nástroja vhodného na komunikáciu v otvorených softvérových projektoch, umožňujúcich kolaboráciu neznámych ľudí bez priameho kontaktu. V ďalšej časti poskytujem pohľad na spôsoby komunikácie návrhu softvéru. V najväčšej časti eseje sa však venujem komunikácii o zdrojovom kóde, ktorý je dôležitým faktorom fungovania projektu. Analyzujem najčastejšie problémy, ktoré v komunikácii o projekte môžu vzniknúť a ktoré aj vznikajú. Následne navrhujem viac, či menej efektívne spôsoby ako komunikáciu k projektu usmerniť, alebo sa vyhnúť už tým častiam, ktoré usmernenie potrebujú.*

**Kľúčové slová:** *komunikácia, open source projekty, riadenie komunikácie*

## **Motivácia**

Komunikácia masívnym spôsobom vplyva na úspešnosť spoločnosti a tým aj na úspešnosť jej projektov. V prosperujúcich spoločnostiach výrazne dominujú iné formy komunikácie ako vo firmách menej prosperujúcich alebo neprosperujúcich. V prosperujúcich firmách

tvoril až 60% komunikácie počúvanie s porozumením a len 20% písanie, pričom v neprosperujúcich firmách bol tento pomer opačný [4].

Je možné vytvárať úspešné projekty aj bez osobného kontaktu s ľuďmi, ktorí pracujú na iných kontinentoch a v rôznych časových pásmach? Je možné na to používať len písomnú komunikáciu a aké problémy táto komunikácia so sebou prináša?

## **Aký komunikačný nástroj používať**

Svet softvéru môžeme rozdeliť na softvér, ktorý má zverejnený zdrojový kód (open source) a uzavretý zdrojový kód (close source). Vývoj programov s uzavretým zdrojovým kódom prebieha v spoločnostiach, ktorých zisk je väčšinou generovaný predajom týchto programov alebo ich podporou. Na rozdiel od toho, programy vyvíjané s otvoreným zdrojovým kódom sú často krát vyvíjané komunitou, ktorej zisk je tvorený dobrovoľným prispievaním používateľov a spoločností využívajúcich daný softvér, alebo zisky vôbec negenerujú. Existujú však aj projekty ktoré majú otvorený zdrojový kód a sú vyvíjané centralizovane. Preto nemusí platiť, že otvorený zdrojový kód znamená vývoj nezávislými programátormi, s decentralizovaným prístupom bez zisku za uskutočnenú prácu.

Vývojári pracujúci na verejných projektoch zväčša nie sú motivovaný peňažne, ale tým, že projekt rieši ich problém a takýto softvér chcú vylepšiť alebo opraviť nejaké jeho chyby. Takýmto spôsobom mu pomyselne vrátia to, čo od neho získali. Keďže je projekt verejne prístupný, nie je vývojár obmedzovaný na prácu na konkrétnej časti ale môže riešiť čokoľvek. Takýmto spôsobom pracujú rôzni ľudia na rôznych častiach, najčastejšie však na tom, čomu rozumejú, alebo čo chcú zlepšiť. Nie vždy je potrebné a vhodné samovoľne začať vyvíjať bez dohody s inými ľuďmi pracujúcimi na rovnakej časti. Tu sa začína prejavovať význam komunikácie. Ako však prebieha komunikácia v tíme tvorenom rôznymi ľuďmi pracujúcimi v rôznych časoch a na rôznom mieste?

Na výber je z viacerých spôsobov komunikácie, a to:

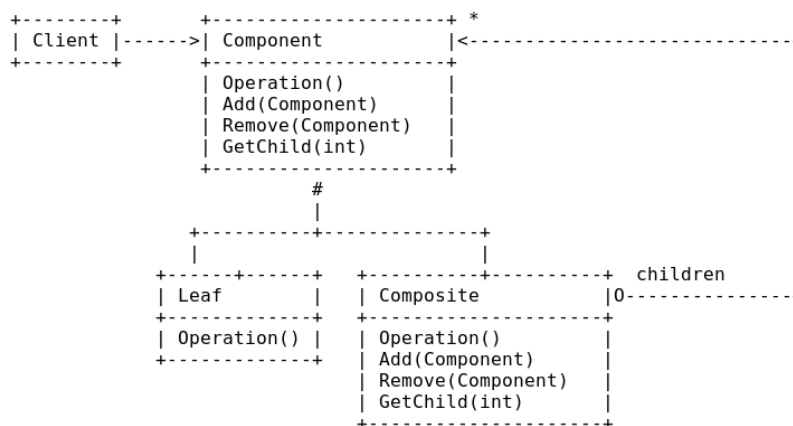
- emailová komunikácia medzi členmi,
- verejný mailing list, alebo jeho ekvivalent,
- IRC komunikačný nástroj, alebo
- wiki s udržovaním aktuálneho stavu vývoja.

Zásadným problémom pri vytváraní projektu v taktom prostredí, je spolupráca členov. Ako efektívne udržiavať aktuálne informácie o tom, kto pracuje na ktorej časti projektu a ako zabrániť prepisovaniu rovnakých častí programu viacerými vývojármi súčasne. Podľa [3] je riešením prístup „robme veci verejne“. Základným prístupom je teda verejná komunikácia medzi členmi tímu, ktorú môže ktokoľvek čítať, poprípade sa do nej zapojiť. Takýmto spôsobom sa k určitej téme vyjadrujú iba ľudia, ktorí na téme pracujú a ostatní si už ľahko vytvoria obraz o spolupracovníkoch. Tento prístup preto vyžaduje komunikáciu cez otvorený nástroj, ktorý udržiava históriu, ako je mailing list alebo jemu ekvivalentné nástroje. Tento nástroj ponúka množstvo výhod, oproti iným riešeniam. Umožňujú verejné diskutovanie o návrhu, čo email nemôže ponúknuť, umožňuje prácu v rôznom čase, čo nemôže poskytnúť IRC a dosahuje dynamiku, ktorá je omnoho vyššia ako u wiki stránok. Ak sa pozrieme na veľké otvorené projekty, zistíme, že uprednostňujú práve komunikáciu prostredníctvom mailing listu, napr. LibreOffice, Eclipse, Apache, atď. V nasledujúcich

kapitolách sú opísané aj nevýhody ktoré tento spôsob prináša, avšak dané problémy sa týkajú aj bežnej komunikácie.

### Vytvárame návrh pomocou diagramov

Vývoj softvéru stojí na kvalite návrhu softvéru a bez neho nie je takmer možný. Návrh je takmer vždy reprezentovaný modelom. Pri centralizovanom vývoji je možné vytvoriť návrhy v jednotnom softvéri, kresliť ich na tabuľu, alebo vytlačiť a ručne popísať. Pri distribuovanom vývoji sú všetky tieto metódy obmedzené alebo nefunkčné. Vývojári aj tak cítia potrebu nejako diagramy a modely distribuovať. Jedným z najjednoduchších prístupov je použiť ASCII umenie (ASCII art), čiže vykreslenie diagramu pomocou znakov, ktoré poskytuje ASCII tabuľka – Obr. 1. Tento typ diagramov sa taktiež často používa aj v hotových normách, ktoré vyžadujú textový zápis. Je ich možné bezproblémovo distribuovať a zobrazovať. Úprava je možná prakticky v akomkoľvek textovom editore, no jednoduchosť úpravy je diskutabilná. Druhým prístupom je využívanie diagramov kreslených vo voľne distribuovaných nástrojoch, čo umožňuje ich používanie pre širokú škálu používateľov, no i tu je možné sa stretnúť s problémami ako je prenositeľnosť na všetky platformy. Posledným prístupom, využívaný na túto komunikáciu, je vytvorenie výsledného diagramu v ľubovlnom nástroji a následná distribúcia len vo forme obrázku. Síce si každý môže diagram prehliadať, jeho úprava je náročná a takmer vždy končí vytvorením nového diagramu.



Obr. 1. Vzor kompozit vykreslený metódou ASCII art.

Prínosom do tejto oblasti sú webové aplikácie, umožňujúce vytváranie diagramov, nevýhodou je však častá nutnosť platiť za kvalitu, alebo možnosti uloženia a komerčného využitia. Preto si myslím, že vývoj v tejto oblasti by mohol byť zaujímavý. Vytvoriť webovú aplikáciu, ktorú by bolo možné nasadiť na server určený na kolaboráciu programátorov, ako je mailing list, fórum atď. a efektívnym spôsobom umožniť previazanie týchto platforiem.

## **Komunikujeme úpravy kódu**

Jedným z najčastejšie riešených vecí na vývojárskych mailing listoch je zdrojový kód, pridávanie nového kódu, odoberanie starého kódu a diskusia o kvalite kódu. Problém tejto komunikácie tkvie práve v samotných odpovediach diskutujúcich a odpovediach na tieto odpovede, ktoré sťažujú získavanie užitočných informácií. Množstvo príspevkov si nevyžaduje odpoveď, respektíve ani neboli myslené ako otázka. Túto skutočnosť vystihuje aj zoznam najčastejších správ, ktoré sa objavujú na mailing listoch a to sú [2]:

- príspevky, ktoré navrhujú netriviálne úpravy alebo rozšírenia projektu,
- príspevky, ktoré vyjadrujú súhlas alebo nesúhlas na príspevok niekoho iného a
- príspevky, ktoré zhromažďujú fakty z predchádzajúcich príspevkov.

Žiaden z týchto príspevkov nevyžaduje našu odpoveď. Ak by aj odpoveď bola vhodná, určite ju vyjadrí niekto iný, ktorý si nie je vedomý toho, že odpoveď sa neočakáva. Rovnako veľkým problémom ako je znepriehľadňovanie komunikácie a pridávanie takzvaného šumu, je odstránenie tohto problému.

Medzi problémy, ktoré by sme mohli zaradiť tiež medzi dôležité pri komunikácii o projekte sú:

- opakujúce sa argumenty od rovnakých používateľov, ktorí si myslia, že ich názor bol prehliadnutý a
- diskusia o netriviálnych záležitostiach, ktorá nevedie k riešeniu aktuálneho problému v softvéri.

Takéto vlákna je potrebné zastaviť alebo usmerniť účastníkov tak, aby dosiahli pokrok v riešení problému. Diskusia o netriviálnych zmenách v softvéri, však môže viesť aj k novým nápadom na rozšírenia projektu a to hlavne v prípade, ak sa problematika rozloží na jednoduchšie časti. Ukončenie však vôbec nie je triviálna záležitosť, pretože „násilné“ ukončenie témy môže odrezať riešenie, ktoré by časom mohlo vzniknúť. Ukončenie príspevkom, ktorý sa snaží usmerniť prispievateľov je tiež možné, no ukončenie nie je okamžité a ešte nejaký čas vzniká dodatočný šum.

Jedným zo špecifických prípadov znepriehľadňovania a zahlcovania komunikácie je oblasť nazývaná sväté vojny.

## **Čo ak stretneme nepriateľov**

Termín sväté vojny sa dostal do povedomia už v roku 1980 [1]. Termín vznikol ako reakcia na nejednotnosť ukladania poradia bajtov v pamäti, takzvaný „Little-Endian“ a „Big-Endian“. Vo všeobecnosti ide o maličkosť, ktorú však diskutujúci vnímajú tak silno, že chcú správnymi argumentmi presvedčiť opačnú stranu o svojej pravde. Problém sa prenáša aj do súčasnosti a vo veľkej miere sa mu darí v oblasti informatiky. Môže sa týkať toho „pravého“ operačného systému, programovacieho jazyka, linuxovej distribúcie, atď.

Sväté vojny môžu hovoriť o programovacom jazyku, knižnici alebo spôsobe programovania a častokrát sú reakciou na aktuálne riešený problém. Ak sa už takáto vojna začne, je potrebné ju ukončiť, aby sa problém mohol správne vyriešiť. Ak vojnu nezastavíme, prispievatelia zaplnia neúžitočnými informáciami celé vlákno správ,

v ktorom sa následne užitočná informácia takmer vždy stratí a problémová oblasť ostane nevyriešená.

Sväté vojny so sebou môžu prinášať aj dezinformovanie čitateľov, ktorí cieľovú oblasť nemusia dostatočne poznať a môžu si o problematike vytvoriť mylnú predstavu. Následne sa pokúšať problém riešiť neoptimálnym prístupom. Týmto prístupom sa znižujú výhody verejného prístupu:

- nie je možné jednoznačne určiť kto, na danej problematike aktívne pracuje a kto sa nechal strhnúť argumentmi a
- aké je optimálne riešenie problému.

Riešenie prebiehajúcej diskusie však nie je možné riešiť uspokojivo [1]. Pre pozorovateľa je jasné, že ani jedna strana neustúpi, no pre účastníka je opačná strana veľmi tvrdohlavá. Ústup nie je možný, pretože by sa chápal ako ustúpenie argumentov protistrany. Ak sa už svätá vojna začne, nie je možné ju uspokojivo vyriešiť a jediná možnosť je ju ukončiť „násilne“.

### **Ako zvíťaziť**

Vhodným prístupom k vláknam, ktoré neprinášajú žiadne užitočné poznatky, je ich hodnotenie. Po zlom hodnotení príspevkov je možné automaticky zhoršovať celkové hodnotenie vlákien, čo môže spôsobiť odradenie ďalších čitateľov prichádzať na neproduktívne vlákna a tým umlčať aj prispievateľov. Základným problémom riešenia, ktoré je bežne nasadzované, je, že v niektorých typoch príspevkov nemožno určiť objektívneho hodnotiteľa príspevkov. Príkladom sú sväté vojny, kde jedna skupina bude hodnotiť svoje príspevky kladne a príspevky druhej skupiny záporne. Potrebné je však sledovať aj množstvo prispievateľov a počet hodnotení. Pri malom množstve hodnotení môže relatívne ľahko pár zlých hlasov ohodnotiť vlákno veľmi negatívne.

Ďalším doplnkovým riešením je odporúčanie čitateľovi len tých vlákien, ktoré kladne hodnotili iní čitatelia, ktorí majú podobné hodnotenia vlákien alebo článkov ako on. Táto metóda sa nazýva kolaboratívne odporúčanie [5].

Zaujímavou oblasťou výskumu, ktorý by mohol viesť k automatickému rozpoznávaniu neefektívnych vlákien, môže byť aj automatické vyhľadávanie komunikačných vzorov na základe obsahu a hodnotenia vlákien. Napríklad sväté vojny sa môžu vyznačovať dvoma rozdielnymi skupinami ľudí, ktorí pridávajú kladné hodnotenia vlastnému názoru a záporne iným.

### **Záver**

V eseji sa podarilo opísať niekoľko závažných problémov pri komunikácii na softvérových projektoch. Niektoré problémy sa však dotýkajú aj všeobecnej komunikácie a nie len komunikácie v softvéri, no tieto problémy sú také závažné, že museli byť spomenuté.

V eseji bola taktiež snaha navrhnúť riešenia opísaných problémov, ako je vytvorenie webovej aplikácie, slúžiacej na tvorbu diagramov, ktorú by bolo možné voľne distribuovať. Taktiež je tu popísaný návrh na nasadenie kolaboratívneho odporúčania pre zefektívnenie vyhľadávania informácií pre konkrétneho používateľa. Zaujímavý je taktiež

návrh smeru vývoja algoritmov pre vyhľadávanie vzorov poukazujúcich na neefektívne vlákna.

Aj keď je komunikácia v distribuovaných projektoch náročná, ukazuje sa, že stále je možná a dôkazom toho sú aj veľké úspešné projekty ako GNU/Linux.

## Použitá literatúra

1. Cohen, D.: *On holy wars and a plea for peace*, [online], 1980 [citované 04.12.2012]. Dostupné z: <http://www.ietf.org/rfc/iem/iem137.txt>
2. Fogel, K.: *Producing Open Source Software*, 150-190, 2005, ISBN:0-596-00759-0.
3. Gutwin, C., Penner, R., Schneider, K.: Group Awareness in Distributed Software Development. In: *2004 ACM conference on Computer supported cooperative work*, ACM Press, New York (2004), 72 – 81.
4. Hybels, S., Weaver, R. L.: *Communicating effectively*. 1988.
5. O'Mahony, M., Hurley N., Kushmerick, N., Silvestre G.: Collaborative Recommendation: A Robustness Analysis. In: *ACM Transactions on Internet Technology*, ACM Press, New York (2004).

## Annotation

### *The faces of communication problems in open source projects*

*The communication between people standing face-to-face is many times complicated and difficult, but we are improving in this sphere day by day. The success of every single man and society is based on communication, as well as their projects. But how can be the communication in project creation successful, when we are distant from each other hundreds of kilometers and dispense us different languages and traditions? In this essay I describe the choice of communicational tools suitable for communication in open software projects that make possible collaboration between foreign people without direct contact. In next part of essay I present a look on communication ways by designing a project. The biggest part of this essay is dedicated to communication about a source code, which is important factor in functioning of whole project. I do analyze the most frequented problems, which can appear in communication about a project and which really appear. Thereafter I suggest more or less effective ways to how the communication guide to the project or evade from those parts, which need the direction.*