

VÝZNAM, NÁKLADY A RENTABILITA SPOJENÉ S VERZIOVANÍM SOFTVÉRU

*„Najzložitejšiou cestou ako dospieť k výsledku
problému je vyriešiť ho!“*

Richard Sámela

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

Abstrakt. Vývoj softvéru je vo všeobecnosti veľmi náročný proces z finančného, časového i intelektuálneho hľadiska. To je aj jedným z hlavných dôvodov, prečo sa v dnešnej dobe využívajú rôzne podporné systémy pri jeho vývoji. Jeden z najdôležitejších je práve verziovací systém, ktorým sa zaoberá táto esej. Avšak často nie sú vývojári dostatočne oboznámení ako, prečo a či sa vôbec oplatí tento systém používať. Uvažujeme nad stanoviskami na základné otázky: Vznikajú náklady spojené s verziovaním softvéru, ak áno, aké? Je rentabilita pri verziovaní softvéru vyššia ako samotné náklady? Vedia vývojári dostatočne efektívne využívať funkcionality, ktorú ponúkajú? Dôležitým faktorom je, aby členovia vývojového tímu nepovažovali používanie verziovacích systémov za samozrejmosť, či za akýsi štandard. Mali by sa k tejto téme postaviť zodpovedne, správne vyhodnotiť, či je daný systém nutné použiť a vybrať čo možno najvhodnejší verziovací systém.

Kľúčové slová: verziovací systém, náklady a rentabilita, manažment podpory vývoja

Úvod

Len málokedy nastane reálna situácia, že by sa od začiatku až do konca procesu vývoja softvéru nič nezmenilo, alebo sa v celom priebehu nevyskytol žiaden problém. Aj preto si členovia vývojových tímov zaužívali pri procese tvorby softvéru používať, či už jeden

globálny, alebo viacero menších podporných systémov. Tie môžu byť medzi sebou navzájom poprepájané. Zrejme žiaden vývoj sa nezaobíde bez réžie spätej s požiadavkami zákazníka, plánovaním jednotlivých cyklov vývoja softvéru, či písaním zdrojového kódu programu.

Zdrojový kód je zrejme najdôležitejšou súčasťou softvérového projektu, ktorú je vhodné verziovať. Práve verziovací systém je tou zložkou podporného systému, ktorá nám to umožňuje.

Verziovanie – áno, či nie?

Myslím, že obe odpovede na túto otázku sú správne, len je dôležité určiť si kedy. Nie vždy je to ale také jednoduché, ako sa na prvý pohľad môže niekomu zdať.

Je veľa rôznych vedeckých názorov o verziovaní softvéru, ktoré istým spôsobom navádzajú vývojárov verziovať najmä zdrojový kód. Napríklad, že deštrukcia histórie dizajnu softvéru je intelektuálny vandalizmus [1]. Na jednej strane s tým názorom ako používateľ verziovacích systémov súhlasím, ale je potrebné verziovať softvér vždy?

Úplne vždy to podľa môjho názoru nie je nutné. Bolo by treba pri tom zohľadniť viacero faktorov, ktoré toto rozhodnutie priamo ovplyvňujú. Jedným z nich je rozsiahlosť projektu. Ako vývojár si viem predstaviť dôležitosť zachovávanie histórie zmien v softvérovom projekte. Samotný softvérový návrh je náročný proces a v závislosti od dizajnu sa počas celej dĺžky vývoja mení [1]. Každá takáto zmena môže so sebou priniesť rôzne chyby, ktoré sú prevažne tým závažnejšie, čím väčší zásah do projektu vykonáva. V prípade, ak sa pri vývoji rozsiahlejšieho projektu používa verziovací systém, nemusí byť odstraňovanie podobných chýb príliš náročné. Preto si myslím, že verziovanie softvéru pri rozsiahlejších projektoch je priam nutné.

Ale čo v prípade, ak ide len o nejaký jednoduchý projekt? Je aj vtedy verziovanie softvéru nutné? Na to je trochu ťažšie odpovedať, pretože je zložité presne definovať, čo znamená "jednoduchý" projekt. Osobne si pod tým predstavujem projekt, ktorého návrh sa od inicializácie až po ukončenie nemení. Vtedy nedochádza ku žiadnym zmenám, a preto ani história dizajnu softvéru pri tomto type projektu nemá podstatný zmysel. Myslím teda, že z hľadiska zachovania zmien v jednotlivých verziách nie je nutné ani verziovanie podobných softvérových projektov. Samozrejme, ale verziovanie podobných projektov nezavrhum. Možno by stálo za zváženie, čo iné okrem zachovávanie histórie zmien v projekte verziovací systém môže poskytnúť a rozhodnúť sa na základe toho.

Ďalším faktorom, ktorý priamo ovplyvňuje rozhodnutie, či verziovať, je aj počet spolupracovníkov v tíme. Verziovací systém zlepšuje kolaboráciu ľudí v tíme [3]. Ako vývojár som už viackrát pracoval vo väčšom tíme pracovníkov. Zatiaľ vždy sme pri vývoji softvéru používali niektorý z verziovacích systémov a musím priznať, že neviem a ani si nechcem predstaviť, ako by sme mohli ako tím inak spolupracovať. Myslím si, že ak by sa na vývoji softvéru podieľal vývojár sám, alebo v najkrajnejšom prípade vo veľmi malopočetnom tíme pracovníkov, bolo by možné sa použitiu verziovacieho systému vyhnúť. Ale pri tejto možnosti by opäť stálo za uváženie, aká je rozsiahlosť projektu. No v inom prípade mi nezostáva nič iné, ako súhlasiť s týmto tvrdením. Osobne si myslím, že v tíme vývojárov, ktorí majú medzi sebou spolupracovať, je používanie verziovacieho systému potrebné.

Verziovacie systémy neslúžia len na lepšiu spoluprácu ľudí v tíme, ale ich cieľom je aj zefektívňovať vývoj jednotlivcov [3]. Rozhodnutie, či vývojár jednotlivec bude používať verziovací systém, spočíva na ňom samotnom. Myslím si, a čerpám aj z osobných skúseností, že vývojári jednotlivci zväčša nepoužívajú pri vývoji verziovacie systémy. Predpokladám, že dôvodom je akási spätosť pojmu verziovania softvérového projektu s pojmom vývoja v tíme. Osobne som práve preto nikdy svoj zdrojový kód neverzioval v projekte, na ktorom som pracoval samostatne. Ale prikláňam sa i k názoru, že si vyvojári ani neuvedomujú výhody verziovania softvéru, aj keď na jeho vývoji sa podieľajú sami. Myslím si, že verziovací systém môže aj samotným vývojárom poskytnúť viacero výhod, akými sú napríklad rýchlejšia identifikácia možnej chyby v projekte, či v niektorých prípadoch aj zálohovanie softvérového projektu.

Verziovanie softvéru - nutnosť, či voľba?

Zrejme stojí za úvahu, ktorá z možností je vhodnejšia. Na jednej strane môže byť prospešné, ak vývojár dostane verziovanie softvéru príkazom, no na strane druhej môže asi aj prísť o niektoré poznatky. Podľa môjho názoru je pravda niekde uprostred.

Najmä praktickosť verziovania sa uchytila vo firemnom prostredí. Verím, že každá spoľahlivá firma zaoberajúca sa vývojom softvéru disponuje verziovacím systémom, ktorý reálne v praxi aj používa. Inak by zrejme nemal zmysel. V opačnom prípade, ak softvérová firma pri vývoji nepoužíva ani najprimitívnejší verziovací systém, býva označená ako "garážovka". Určite je na mieste, keď si vo firemnom prostredí uvedomujú rozsiahle výhody. Verziovanie softvéru je začiatkom úspešnému softvérovému produktu [2]. Schválne, ktorá zo softvérových firiem, by nechcela aby práve ich produkt bol úspešným? Zrejme žiadna.

Ale čo s takými zamestnancami? Je pre nich výhodnejšie, že musia, či môžu verziovať softvér? Pri tom zohrávajú pomerne podstatnú úlohu skúsenosti vývojára s verziovacími systémami. Myslím, že jeho rozhodnutie pre verziovanie by bolo pozitívne. Ale čo ak vývojár nemá postačujúce, prípadne žiadne poznatky ohľadne verziovania softvéru? Takýto vývojár má budúcnosť v prostredí softvérovej firmy, len ak sa doučí verziovať softvér. Ale túto možnosť osobne už považujem za nanútenú. A zrejme je to už len na vývojárovi samotnom, ako sa k tomu postaví. Či sa naučí len základy, ktoré nevyhnutne potrebuje, aby mohol vo firme existovať, alebo v lepšom prípade ho to podnieti k štúdiu a vo vlastnom záujme si bude vyhľadávať viac a viac informácií o verziovaní softvéru. Moje subjektívne poznatky ale hovoria skôr o prvom prípade.

Odhliadnuc od firemného prostredia a vo všeobecnosti od prostredia, v ktorom verziovanie softvéru nie je nutné, uvedomujú si aj tak vývojári výhody verziovania softvéru? Opäť, ak vnímame vývojára ako znalého a pokročilého v oblasti verziovania, zrejme mal dôvod sa tomu venovať a vzdelávať sa v tejto oblasti. Nakoľko to pre neho nebolo nutné, predpokladám, že ho k tomu viedla jeho húževnatosť a túžba po vedomostiach. Podľa mňa je práve tento prípad najlepším, pretože vo všeobecnosti človek, ktorý má cieľ a chuť sa niečo naučiť, je ochotný aj vynaložiť väčšie úsilie preto, aby ho dosiahol. Aj práve preto predpokladám, že vývojár, ktorý sa samostatne naučil verziovať softvér, často disponuje aj lepšími vedomosťami o rôznych verziovacích systémoch, princípoch verziovania, či výhod a opačne.

No aj ten najskúsenejší vývojár, ktorý vie výborne verziovať softvér, bol raz nováčikom v tejto oblasti. Ako verziovanie softvéru vníma taký nováčik? Vidí v tom stratu času alebo schopnosť verziovať softvér vníma ako investíciu do budúcnosti?

Nováčik vs. verziovanie softvéru

Zrejme každý vývojár si pri prvom kontakte s verziovacím systémom položil otázku, či je pre neho výhodné začať verziovať práve teraz. Pri otázke rozhodovania sa, či verziovať alebo nie, zohráva nemalú úlohu strach pred niečím novým. Obavy pri začatí s verziovaním často ani nesúvisia s obavami o to, ako sa naučiť používať systém. Najväčšou prekážkou býva principiálna neznalosť spôsobu ako verziovanie funguje.

Myslím, že vývojárov, ktorí začínajú verziovať softvér odradí práve nutnosť investovať čas do výučby. Čerpám trochu i z vlastných skúseností. Ochota investovať čas zvyčajne postupne klesá, pretože vývojár zo začiatku nemusí vidieť žiaden prínos z verziovania, ani iné výhody, ktoré by z neho mali plynúť. Nehovoriac o tom, keď sa dostavia niektoré z prvotných problémov. Avšak tieto vzniknuté problémy ani nemusia spočívať v samotných vývojároch, ako v zlom výbere verziovacieho systému. Aj verziovacích systémov je niekoľko druhov, od jednoduchších po zložitejšie. Na výbere vhodného verziovacieho systému záleží [4]. Na prvý pohľad sa vývojárovi môže zdať, že ak sa ako prvý naučí zložitý systém, odpadá mu povinnosť postupne sa učiť stále niečo nové. Myslím si, že voľba vybrať si na začiatok ten najťažší systém nie je práve najlepšia. Keď ešte vývojár nemá dostatočné skúsenosti, je oveľa väčšia pravdepodobnosť, že nevedomky urobí začiatočnícke chyby, ktoré ho môžu neskôr odradiť od verziovania. Problémy tohto druhu je možné aspoň mierne eliminovať vybratím a použitím vhodného verziovacieho systému. Správny výber verziovacieho systému šetrí čas vynaložený členmi tímu na vývoj softvérového produktu [4].

Programátori bývajú vo všeobecnosti vnímaní ako skupina ľudí, ktorá veľmi pružne reaguje na nové technológie a trendy, no súčasne máme vo veľkej obľube vytvoriť si vlastný svet, kde používame iba svoje obľúbené postupy. Preto si myslím, že zaučenie nováčika, na istý verziovací systém môže byť jednoduchšie, ako meniť zaužívané zvyklosti pokročilého vývojára. Práve tu je rozhodnutie o tom, či použiť nový verziovací systém oveľa zložitejšie. No podľa môjho názoru, každý dobrý vývojár by sa rozhodne nemal brániť novým veciam.

Záver

Pretože je vývoj softvéru vo všeobecnosti časovo, finančne aj intelektuálne náročný projekt, je veľmi výhodné používať verziovací systém. Nie len z dôvodu, že je prostriedkom pre prácu v tíme, ale najmä pre vyhliadky do budúcnosti. Vývoj softvéru prechádza viacerými iteráciami, ktoré môžu obsahovať rôzne zmeny, pridané funkcionality a iné. Práve tu sa dostaví výsledný efekt, prečo je verziovanie softvéru potrebné a efektívne.

Ako pri nováčikoch v oblasti verziovania, tak i pri pokročilých, vždy treba myslieť na budúcnosť. Dobrá schopnosť verziovať softvér tiež napomáha k osobnostnému rastu vývojára. Aj to je dôvod prečo investovať čas do naučenia sa, správne verziovať softvér.

Hoci sa niekedy môže zdať používanie niektorého z verziovacích systémov nevýhodným, aj tak sa ho v konečnom dôsledku oplatí používať. Preto môj postoj k verziovaníu znie: "Určite áno!".

Použitá literatúra

1. Cliff B. Jones: Reflections on, and predictions for, support systems for the development of programs, *School of Computing Science*, Newcastle University, UK, 2008
2. T. Zimmermann, P. Weißgerber, S. Diehl, and A. Zeller, Mining Version Histories to Guide Software Changes, *Proc. Int'l Conf. Software Eng.*, pp. 563-572, 2004
3. Rita Nienaber, Elsabe Cloete: A Software Agent Framework for the Support of Software Project Management, *University of South Africa*, 2003
4. Annie T.T. Ying, Gail C. Murphy, Predicting Source Code Changes by Mining Change History, *IEEE Transactions on Software Engineerings*, vol. 30, no. 9, 2004

Annotation

Importance, costs and profitability associated with versioning software

Software development in general is a very difficult process of financial, time and intellectual viewpoint. This is also one of main reasons why nowadays there are used various support systems by software development. One of the most important support systems is just version control system, which is reflected in this essay. However, developers are only for a few times familiar how, why and whether is convenient use this system. Let's reflect about the views of the fundamental questions: Does exist any emerging costs associated with software versioning and if does, so what? Is it profitability of software versioning higher than the actual cost? Can developers enough effectively use the functionality what version systems offer? It is important to members of the development team do not consider usage version control systems for granted, or as a kind of standard. They would be responsible in this situation, they should evaluate correctly whether version control system must be used and then they should choose the most appropriate one.