VERZIOVACÍ SYSTÉM – CENTRALIZOVANÝ ALEBO DISTRIBUOVANÝ?

Odpoveď nie je taká jednoznačná, ako by sa mohlo zdať.

Peter Sivák

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

Abstrakt. *Použitie verziovacieho systému v akomkoľvek projekte sa považuje za nevyhnutnosť. Kedysi boli k dispozícii iba centralizované typy systémov, ale postupom času sa vyvinuli distribuované prístupy k verziovaniu – treba si vybrať jeden z nich. Distribuovaný systém sa stáva čoraz populárnejším, čo ale neznamená, že si ho automaticky vyberiem. Cieľom mojej eseje je ukázať, že obidva prístupy majú svoje výhody aj nevýhody a až po dôkladnej analýze sa dá zodpovedne zvoliť jeden z nich, ktorý bude pre projekt prínosnejší. Analyzujem tu rôzne aspekty, ktoré môžu mať vplyv na konečné rozhodnutie. Dotýkam sa tu problému prístupu k repozitáru, čo sú klady a zápory distribuovania, aké veľké tímy kooperujú na danom projekte a ktoré riešenie je pre nich lepšie. Ďalej popisujem potenciálne riziká vyplývajúce z nepravidelného ukladania verzií na server a ku koncu eseje sa venujem prechodu medzi verziovacími systémami a aké prínosy a náklady to so sebou prináša. Treba si taktiež uvedomiť, ktorý aspekt má pre vývojový tím väčšiu prioritu a aj to zahrnúť do finálneho rozhodnutia.*

Kľúčové slová: *verziovací systém, manažment podpory vývoja, repozitár, kolaborácia*

Verziovací systém – samozrejmosť pri vývoji

Vývoj softvéru sa skladá z viacerých fáz a môže prebiehať vo viacerých iteráciách. Vývojári musia počas týchto fáz implementovať množstvo kódu, aktualizovať ho,

zdokonaľovať a udržiavať. Preto je jasné, že celý softvér, resp. systém sa nedá vyvinúť na jeden krát. Je potrebné uchovávať viacero verzií zdrojového kódu, buď ako zálohu, alebo pre prípad, keby bolo potrebné vrátiť sa k starším verziám programu. Je preto nevyhnutné používať verziovacie systémy, ktoré poskytujú kompletnú správu verzií súborov a uľahčujú prácu vývojárom.

Ktorý si ale vybrať?

V súčasnosti sú k dispozícii dva hlavné typy verziovacích systémov – centralizovaný a distribuovaný, resp. decentralizovaný. Každý z týchto systémov má rôzne prípady použitia a každý z nich má svoje výhody aj nevýhody. Na webe je avšak možné nájsť množstvo článkov, kde sa píše, prečo je lepšie používať distribuované systémy, aké majú výhody oproti centralizovaným systémom a prečo prejsť na distribuovaný systém, ak už práve používate centralizovaný.

Mohlo by sa zdať, že odpoveď na túto otázku je jednoznačná a nie je čo riešiť. Cieľom tejto eseje je vyvrátiť toto tvrdenie a ukázať, že má zmysel uvažovať nad každou z týchto možností. Zároveň sú v nej spísané určité myšlienky, ktoré by mali napomôcť pri výbere zo spomínaných dvoch verziovacích systémov.

Prístup k repozitáru

Vždy keď vývojár dokončí nejakú ucelenú časť svojej práce, je bežná prax, že svoje zmeny uloží do verziovacieho systému. Pri centralizovanom systéme musí mať k dispozícii internetové pripojenie, pretože všetky údaje sú uložené na jednom centrálnom repozitári. Naopak, pri distribuovanom type systému internetové pripojenie nepotrebuje, keďže každý člen tímu má uloženú kópiu všetkých potrebných údajov aj ich verzií na svojom lokálnom počítači. Toto je dosť veľká výhoda, keď vývojár cestuje vlakom alebo lietadlom a nemá k dispozícii internet a potrebuje urobiť nejaké zmeny a uložiť ich do verziovacieho systému.

Limitácia aktualizovania projektu je ešte výraznejšia, keď sa ukladajú zmeny na server veľmi často. Zlaté pravidlo verziovania dokumentov znie: ukladať zmeny do repozitára skoro a často. Keby vývojári neukladali svoje zmeny na server často, ale neskoro po obrovských zmenách, oveľa ťažšie by sa následne vykonávala integrácia ich zmien so zmenami ostatných členov tímu. Keď sa ukladajú zmeny často, ostatní programátori v dostatočnom predstihu vidia, ako sa systém vyvíja a môžu sa tomu lepšie prispôsobiť. Pretože treba sa zamyslieť nad tým, že pokiaľ sa zdrojový kód nenachádza vo verziovacom systéme, ako keby neexistoval. Pre samotného autora kódu možno áno, ale pre tím ako celok nie a to je o dosť dôležitejšie. Preto je dobré ukladať zmeny na server často [2].

Pri centralizovanom systéme bez internetového pripojenia je ukladanie zmien do repozitára veľkým problémom. Preto, keď sa jedná o prístup k repozitáru, je jasným víťazom distribuovaný prístup.

Distribuovanie – dobré či zlé?

V predchádzajúcej časti som rozoberal, ako sa pri distribuovanom prístupe kopíruje celý repozitár na jednotlivé lokálne počítače všetkých vývojárov. Teraz sa môžeme zamyslieť, či je vhodné mať skopírovaný celý repozitár a vlastne skoro to isté na viacerých počítačoch? Nie je to zbytočné? Nezaberá to príliš veľa miesta na disku? V dnešnej dobe sa zdá, že problém s miestom na disku by už nemal byť taký rozsiahly, ako tomu bolo kedysi. Dávnejšie boli kapacity diskov rádovo menšie a bolo treba viac šetriť miestom. Momentálne sú už kapacity diskov niekoľkonásobne väčšie, takže na nich môžeme mať umiestnených oveľa viac súborov. Takže problém s diskom pri mnohonásobnom kopírovaní repozitárov by už mohol byť odstránený.

Na jednej strane kapacity diskov sú čoraz väčšie, na druhej strane ešte aj verziovacie systémy podporujú kompresiu súborov, takže aj tým odstraňujú problém s miestom. Problém tu ale nastáva, ak máte verziovaných veľa obrovských binárnych súborov [4], ako napríklad obrázky a videá, tenzory komprimujúce video, trojrozmernú grafiku v skomprimovanom formáte, rôzne trénovacie vzorky pre strojové učenie a tak ďalej. Tie sa nedajú verziovať tak inteligentne, ako je to napríklad u zdrojového kódu programov. Väčšinou sa musí pre každú verziu systému skopírovať celý binárny súbor, čo môže byť nesmierne nevhodné pri rádovo tisíckach až desaťtisíckach súborov. Vtedy zabraný priestor na disku narastá exponenciálne – s tým, že sa to ešte musí diať na každom počítači zúčastneného vývojára. Preto si myslím, že v takomto prípade je lepšie použiť centralizovaný prístup oproti distribuovanému.

Ale zase to nie je také jednoznačné. Kopírovanie takmer tých istých údajov na niekoľko počítačov môže byť aj užitočné, keď sa jedná o zálohovanie. Pri centralizovaných systémoch existuje len jedna „oficiálna“ kópia verziovaných údajov, takzvané centrálne úložisko a je tu veľké riziko, že disk sa môže poškodiť, či už fyzicky, alebo niekto nedopatrením vymaže časť údajov z neho a tieto údaje sa stratia a už nie je možné sa k nim vrátiť. Síce niektorí vývojári si môžu neoficiálne robiť zálohy, ale nie je to tak organizované ako pri distribuovanom prístupe. V tomto smere by som dal teda prednosť distribuovaným verziovacím systémom.

Veľkosť projektu

Ďalším dôležitým aspektom pri výbere verziovacieho systému je veľkosť projektu. Veľkosťou projektu myslím to, že čím je projekt rozsiahlejší, tým viac zdrojového kódu a dokumentov sa v ňom musí vytvárať a tým väčší tím ho tvorí. Väčší tím znamená aj väčšiu kolaboráciu medzi jednotlivými členmi.

Existujú aj také prípady, že členovia tímu sú rozmiestnení na rôznych miestach zemegule a ťažšie sa im vtedy medzi sebou komunikuje. Myslím si, že v tom prípade je lepšie použiť distribuovaný systém s tým, že jednotlivé lokálne tímy by pracovali nezávisle na svojich úlohách, pravidelne by ich aktualizovali na svojom repozitári a keď by prácu dokončili, zmeny z jednotlivých lokálnych repozitárov by sa potom ľahšie medzi sebou integrovali, nakoľko jednotlivé verzie prác lokálnych tímov by boli stabilnejšie a vznikalo by menej konfliktov. Naviac tu môže existovať viacero úrovní repozitárov, kde programátori ukladajú svoje verzie do lokálnych repozitárov, ktoré sa potom integrujú do

repozitárov na vyššej úrovni, ktoré môžu mať opäť nad sebou ďalšiu úroveň a tak ďalej. Vzniká tu akási hierarchia a preto sa takéto typy systémov nazývajú hierarchické distribuované systémy.

Na druhej strane treba ešte zvážiť, že sa budú kopírovať repozitáre medzi jednotlivé počítače. Ako som už v predchádzajúcich častiach tejto eseje spomínal, distribuovanie veľkých súborov medzi jednotlivé počítače býva často problém. Treba preto brať do úvahy aj tento aspekt a prehodnotiť, čo a koľko toho sa bude kde ukladať.

Pre menšie tímy, ktoré sú väčšinou aj vo fyzickom kontakte, je podľa mňa použitie distribuovaného systému redundantné. Systém zvykne byť v tom prípade jednoduchší a nie je potrebná taká veľká kolaborácia ako pri veľkom tíme. Väčšinou vtedy stačí mať jeden centrálny repozitár, cez ktorý komunikujú všetci členovia tímu.

„Antikolaboratívny“ prístup

Keď som v jednej z predchádzajúcich častí tejto eseje rozoberal prístup k repozitáru, spomínal som, akú dôležitú rolu hrá pri odovzdávaní softvérového artefaktu do verziovacieho systému internetové pripojenie. Uvádzal som, že nedostupnosť internetového pripojenia oveľa lepšie riešia distribuované systémy, pretože odovzdávanie verzie softvéru na server pri nich obsahuje ešte medzikrok uloženia zmien do lokálneho repozitára, kde nie je potrebný prístup k internetu. Programátor preto nie je nútený odovzdávať svoj príspevok priamo na server a paradoxom je, že to môže byť výhoda aj nevýhoda.

Existuje tu totiž riziko, ktoré nie je také zrejmé a nemusí si ho každý uvedomovať. Programátor si môže totiž povedať, že nie je nútený ukladať zmeny do verejného repozitára a preto ich tam ani ukladať nebude. Môže v pokoji, samostatne a bez konfliktov pracovať na implementácii svojej úlohy a na server uloží svoje zmeny až vtedy, keď už bude mať všetko hotové. Nebude totiž odovzdávať svoju úlohu po malých príspevkoch, odovzdá všetko naraz. Komunita okolo centralizovaného verziovacieho systému *Subversion* nazýva tento fenomén „zhodenie bomby“ a je považovaný za antisociálny a antikolaboratívny [3]. Takýto veľký príspevok sa potom ťažšie integruje, horšie udržuje, keďže treba vytvárať viac zložitejších testovacích prípadov a takisto takáto veľká zmena je oveľa menej čitateľná, ako keby sa ukladali zmeny na server po častiach.

Centralizované systémy tento problém úplne odstraňujú, pretože nútia programátora ukladať svoje zmeny priamo do verejného repozitára, kde všetci ostatní zúčastnení majú možnosť dané zmeny hneď vidieť a prispôsobiť tomu aj svoju ďalšiu prácu. Pri distribuovanom prístupe tento problém môže vznikať, ale keď na to programátor myslí, ľahko sa tomu vyhne. Stačí si len povedať zlaté pravidlo verziovania dokumentov – ukladať zmeny do repozitára skoro a často.

Prechod medzi verziovacími systémami

Doteraz som v eseji opisoval rôzne scenáre, ktoré nám napomáhajú k rozhodnutiu, ktorý verziovací systém je pre nás lepší. Môže ale nastať aj situácia, že vývojový tím už používa jeden typ verziovacieho systému, ale rozhodol sa prejsť na druhý typ systému, pretože sa javí pre daný tím výhodnejší. Mohlo by sa zdať, že situácia je tým pádom vyriešená,

dokonca že stav sa ešte zlepšil, pretože sa našla lepšia alternatíva k momentálnemu riešeniu. Toto je opäť mylný resp. neúplný predpoklad, pretože sa uvažoval len samotný výsledok a neboli brané do úvahy dôsledky vyplývajúce zo samotného prechodu medzi systémami. Treba totiž zvážiť, či celkový prínos z nového verziovacieho systému je väčší ako celkové náklady spojené so samotným prechodom medzi systémami. Až potom si tím môže povedať, že prechod sa oplatí. Keďže prechod z distribuovaného systému na centralizovaný sa v praxi vyskytuje len málokedy, popisujem teda iba prínosy a problémy spojené s prechodom z centralizovaného systému na distribuovaný.

Prínosov je hneď niekoľko. Ako som už viac krát spomínal, zlepšila by sa podpora pre vývoj softvéru aj bez možnosti internetové pripojenia, avšak treba si dať opäť pozor na antikolaboratívny prístup. Ďalej by sa podporil prístup aj pre programátorov, ktorí nie sú priamymi účastníkmi vývoja. Potenciálni prispievatelia bez práva na odovzdávanie verzií pri centralizovanom prístupe môžu pracovať len veľmi ťažko. Často to vyúsťuje do situácie, kde si musia vytvárať paralelné repozitáre, aby mohli riadiť väčšie zmeny. Pri distribuovaných systémoch má každý prispievateľ svoj vlastný repozitár, takže môže inkrementálne ukladať svoje zmeny, čo veľmi zjednoduší celý vývoj [1]. Ďalším prínosom je zlepšenie automatického zlučovania vývojových vetiev, ktoré je oveľa efektívnejšie a rýchlejšie oproti centralizovaným systémom. Z toho vyplýva aj väčšia podpora pre experimentálne zmeny. Keďže je zlučovanie rýchlejšie, programátori sa nemusia báť vytvárať si rôzne vlastné vývojové vetvy, na ktorých môžu robiť experimentálne zmeny, s tým že keď sa vetvy zlúčia dokopy, trvá to oveľa menej ako pri centralizovanom prístupe.

Avšak s uvedenými prínosmi prichádzajú aj náklady, ktoré treba brať pri prechode do úvahy. Nechávam na čitateľa, aby sa pokúsil zodpovedať niekoľko otázok, ktoré môžu vplývať na samotné rozhodnutie. Sú všetky spomínané výhody naozaj také podstatné, že sa oplatí prejsť na nový verziovací systém? Naozaj prechod zredukuje bariéry pre nových prispievateľov? Ako sa zmenia vývojové procesy? Ako sa tím vysporiada s iným štýlom číslovania a označovania verzií? Aké nové komplikácie sa môžu objaviť použitím distribuovaného prístupu? Jedná sa o *open-source* alebo *closed-source* typ projektu? Toto všetko závisí od konkrétneho vývojového tímu a konkrétneho projektu a treba dôkladne zvážiť celkový dopad naň pri podstúpení prechodu z jedného verziovacieho systému na druhý.

Záver

V tejto eseji som rozoberal dva hlavné prístupy k verziovaniu dokumentov a zdrojového kódu – centralizovaný a distribuovaný. Na webe môžete z viacerých zdrojov nájsť, že distribuované systémy sa stávajú čoraz populárnejšími v dnešnej dobe a mohli by ste si mylne bez uvažovania vybrať distribuovaný verziovací systém pre váš projekt. V tejto eseji som poukázal na to, že to nie je také jednoznačné a treba uvažovať nad oboma možnosťami.

Každá z nich má svoje pre a proti. Treba sa zamyslieť, čo je pre váš projekt prednejšie – bezpečnosť a zálohovanie, rýchlosť, zabrané miesto na disku, nezávislosť na internetovom pripojení alebo jednoduchosť? Ak už jeden typ verziovacieho systému používate a rozhodli ste sa prejsť na druhý typ, zamysleli ste sa nad tým, aké sú celkové

prínosy a náklady s tým spojené? Až po zodpovedaní týchto otázok sa môžete zodpovedne rozhodnúť pre jeden alebo druhý z týchto typov systémov.

Použitá literatúra

1. ALWIS, B.D., SILLITO, J.: Why Are Software Projects Moving From Centralized to Decentralized Version Control Systems?. Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, Chase '09, 2009.
2. ATWOOD, J.: Check In Early, Check In Often [online]. Coding Horror, 2008. http://www.codinghorror.com/blog/2008/08/check-in-early-check-in-often.html.
3. COLLINS-SUSSMAN, B.: The Risks of Distributed Version Control [online], 2005. http://blog.red-bean.com/sussman/?p=20.
4. LIONETTI, G.: What is Version Control: Centralized vs. DVCS [online]. Atlassian DVCS Guide, 2012.
   http://blogs.atlassian.com/2012/02/version-control-centralized-dvcs/.

Annotation

Version Control System – Centralized or Distributed?

Using version control system in any project is necessary. In the past there were only centralized types of systems but distributed approaches were developed later – you have to choose one of them. Distributed systems become more popular today but it does not mean that you automatically choose them. The goal of my essay is to show you that both principles have advantages and disadvantages and only after proper analysis you can responsibly choose one of them, which will better suit to your project. I analyze here various aspects which can have impact on the final decision. I illustrate here repository access problem, which are pros and cons of distributing, how large teams cooperate on the project and which solution is better for them. Later I describe potential risks resulting from irregular committing of versions to server and to the end of this essay I talk about the transition between the version control systems and what are the benefits and costs of it. You also have to realize, which aspect has for the team more priority and include also that to the final decision.