

# PODPORNÉ PROSTRIEDKY PRE VERZIOVANIE: VHODNÝ VÝBER PRE NÁŠ TÍM?

*Budúcnosť je jasná, budúcnosť sú distribuované  
verziovacie systémy... alebo centralizované???*

Balázs Nagy

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

**Abstrakt.** *Vývoj softvérových produktov v dnešnej dobe je proces, ktorý trvá zvyčajne mesiace, ale niekedy aj niekoľko rokov. Takéto projekty potrebujú nielen dostatočné množstvo ľudských zdrojov, ale aj podporné prostriedky, ktoré im umožnia spoluprácu a efektívne využitie pracovnej doby. Jeden z týchto prostriedkov je verziovací systém, ktorý je zodpovedný za spoluprácu najmä na zdrojových kódach. Vybrať si z týchto prostriedkov však nie je také jednoduché, ako sa to môže zdať, existuje ich totiž mnoho. V tejto eseji sú spomenuté ich výhody a nevýhody, vlastné skúsenosti s nimi. Esej sa taktiež zaoberá vhodným výberom a samotným spôsobom výberu systému pre verziovanie.*

**Kľúčové slová:** *podporné prostriedky, verziovanie, verziovacie systémy,*

## **Verziovanie – Kedy? Prečo? Ako?**

Cieľom eseje je ponúknuť čitateľovi prehľad o známejších verziovacích systémoch, vysvetlenie ich hlavných výhod a nevýhod, a taktiež poukázať na dôležité aspekty pri výbere verziovacieho systému pre konkrétne tímy.

## Začíname s verziovaním

Podporné prostriedky sú neoddeliteľnou súčasťou vývoja každého softvérového projektu. Použitie jednotlivých systémov je treba stanoviť už na začiatku vývoja, pričom tento proces robí často ťažké hlavy manažérom. Existujú však komplexné riešenia, ktoré obsahujú všetky potrebné podporné prostriedky v jednom balíčku. Tieto systémy sú ale zvyčajne nesmierne drahé, a preto nedostupné pre malé firmy a pre študentov. Jeden z týchto prostriedkov je aj systém pre verziovanie, ktorý ponúka nasledujúce prínosy pri vývoji softvéru:

1. viacerí môžu robiť súčasne na tom istom zdrojovom kóde
2. súčasná práca na viacerých verziách
3. zálohovanie bez zbytočne vynaloženého úsilia
4. jednoduchý návrat k predošlým verziám v prípade poškodenia súčasnej verzie
5. sledovanie zmien a aktuálneho stavu produktu
6. sledovanie práce jednotlivých vývojárov

Hoci všetky tieto vlastnosti sú dôležité a majú svoj prínos pri vývoji, v praxi sa využíva často len niekoľko z nich. Dôvodom toho je buď neinformovanosť manažérov, alebo nepotrebnosť pri menších projektoch.

Pri verziovaní je dôležité vedieť, čo všetko je možné verziovať a čo nie. Niektoré typy súborov sú totiž nevhodné na verziovanie, ako sú napríklad prekladaný kód, automaticky vygenerovaný kód a dokumentácia alebo súbory obsahujúce bezpečnostné informácie. Naopak môžeme verziovať zdrojový kód, HTML kód, obrázky, konfiguračné súbory, dokumentáciu, a všetko ostatné, čo je potrebné pre vchod projektu.

Od vzniku prvého verziovacieho systému prešli tieto prostriedky evolúciou, ktorá spôsobila, že súčasne existuje niekoľko prístupov k verziovaniu [1]. Poznáme viacero softvérových riešení s týmto účelom (pozri Tab. 1). Hoci tento fakt znamená pre manažéra podpory vývoja prácu navyše kvôli preštudovaniu alternatív, výhodou je možnosť vybrať si vhodný systém šitý na mieru tímu.

**Tab. 1.** Vlastnosti niektorých verziovacích systémov s rôznym prístupom k verziovaniu.

Názov systému	Licencia	Prístup	Riadenie konfliktov	Manažovanie histórií
Subversion	Apache/BSD (Voľne dostupný)	Centralizovaný	Zlúčenie alebo uzamknutie	Snímka a súbor zmien
Team Foundation Server	MSDN (Kommerčný)	Centralizovaný	Zlúčenie alebo uzamknutie	Súbor zmien
Git	GPL (Voľne dostupný)	Distribučný	Zlúčenie	Snímka

Rôzne zdroje, ale aj vlastné skúsenosti hovoria o tom, že táto oblasť softvérových produktov ponúka kompletné riešenia aj v podobe voľne dostupných softvérových systémov. Softvéry ako Git, Bazaar, Mercurial alebo Subversion sú všetky voľne dostupné, a používa ich mnoho firiem a tímov. Komerčné systémy sú použité najmä vo veľkých

firmách, kde sú ochotný zaplatiť ich cenu a vedia využiť funkcionality ktoré ponúkajú navyše.

## Stačia nám centralizované systémy?

Prvé významnejšie systémy pre verziovanie vznikli 90-tých rokoch, a boli založené na princípe centralizovaných systémov. Niektoré z nich sú použité a vyvíjané dodnes. Dokážu plniť všetky požadované funkcionality, ktoré sú na tieto systémy kladené.

Problémom pri týchto systémoch bolo najmä to, že staršie verzie nepodporili riadenie konfliktov a eliminovali možnosť vzniknutia konfliktov uzamknutím jednotlivých súborov pre jedného používateľa [2]. Bolo to jednoduché riešenie, ale nie príliš efektívne. Neskôr prišli systémy s možnosťou editovania súborov naraz viacerými používateľmi, čo však priniesol so sebou problém konfliktov, ktoré bolo treba pri zlúčení odstrániť.

Medzi centralizované systémy patrí aj Team Foundation Server (TFS), s ktorým mám aj vlastné skúsenosti. Je to komerčný nástroj spoločnosti Microsoft, ktorý ponúka rozsiahle riešenie nielen pre verziovanie, ale aj pre manažovanie úloh, pre reportovanie, pre dátový sklad, atď. Systém bol použitý v relatívne malom tíme, a keďže celý systém je vynikajúco zdokumentovaný, jeho inštalácia, nastavenie a použitie je naozaj jednoduché. Jeho ďalšou výhodou je kompatibilita a spolupráca s nástrojmi ako MS Office Excel, Share Point Server ale aj priama podpora komunikácie TFS s Ultimate verziou vývojového prostredia Visual Studio. TFS ponúka taktiež webové rozhranie na prehľadávanie medzi rôznymi verziami zdrojového kódu, úlohami a kompilovanými verziami produktu. Webové rozhranie je užitočné najmä členom tímu, ktorí nerobia priamo vo vývojovom prostredí, ale majú úlohy, ktoré s týmito časťami projektu súvisia.

Team Foundation Server by som odporučil najmä firmám a tímom, ktoré vyvíjajú vo vývojovom prostredí Visual Studio a v programovacích jazykoch podporovaných týmto prostredím. Je to kompletne riešenie pre malé, stredné a veľké projekty. Jediným dôvodom na nevyužitie tohto softvéru môže byť iba jeho cena, ktorá sa pohybuje od 500 \$ až do 10 000 \$ v závislosti od obsahu balíka. Ďalšie komerčné riešenie sú napríklad ClearCase a Rational Team Concert od IBM alebo StarTeam od Borland-u.

Okrem TFS však existujú aj voľne dostupné alternatívy pre centralizované verziovanie, ako sú napríklad CVS a Subversion. Tieto majú tiež svojich podporovateľov a používateľov.

Ak chceme odpovedať na otázku, či nám stačia centralizované systémy pre verziovanie, odpovedal by som, že určite áno. Všetky produkty ktoré boli spomenuté ponúkajú základnú funkčnosť, ktorá sa od nich očakáva. Niektoré navyše obsahujú funkcionality ktoré sú čerešničkou na torte. Na koniec by som v niekoľkých bodoch zhrnul dobré rady pre používateľov týchto systémov.

1. Získanie najnovšej verzie vždy pred začatím práce
2. Časté a skoré odovzdanie (angl. commit) zmien
3. Odovzdanie častí rozumnej veľkosti
4. Odovzdanie vždy iba funkčného kódu
5. Komunikácia s členmi tímu.

## Distribučované systémy - redundancia podporujúca bezpečnosť

V poslednej dobe sa čoraz viac začínajú presadzovať distribučované systémy, ktoré sa zdajú byť budúcnosťou verziovacích systémov. Ich sila spočíva najmä v nasledujúcich vlastnostiach:

1. Každý vývojár má k dispozícii plnohodnotný repozitár s projektom
2. Jednoduché automatické spojenie verzií
3. Podpora zmien na experimentálnej úrovni
4. Práca s repozitármi bez pripojenia na server

Často najsilnejším argumentom na prechod z centralizovaného systému na distribučovaný je práve prvý bod. Ďalšou výhodou, ktorá z tohto bodu vyplýva je to, že na klientskych počítačoch sú uložené zálohy (klientske úložiská) [3]. Tie môžu byť veľmi užitočné pri poškodení dát na jednej z klientskych staníc. Mohli by sme povedať že je to plytvanie pamäťou, a že redundancia je zvyčajne nežiadaná v informačných technológiách, ale aj tento prípad svedčí o tom, že niekedy môže byť aj užitočná.

Druhým nástrojom, s ktorým mám skúsenosti, je Git. Na rozdiel od TFS je to voľne dostupné riešenie, a má plne distribučovaný prístup k verziovaniu. Bol vytvorený Linusom Torvaldom pre vývoj jadra Linuxu, čo hneď naznačuje jeho silu pri tvorbe rozsiahlych projektov. Pri vytvorení sa inšpirovali systémami ako je BitKeeper a Monotone, neskôr sa Git stal kompletným systémom na správu revízií a v poslednej dobe je používaný v mnohých projektoch.

Git som použil pri vývoji softvérového produktu v programovacom jazyku Java vo vývojovom prostredí NetBeans. Pre Git existuje rozšírenie v NetBeans, ktoré umožňuje jednoduché odovzdanie zmien vykonané programátorom, ktoré sa objavujú iba na lokálnom úložisku. Po otestovaní a overení riešenia môže programátor potlačiť (angl. push) vykonané zmeny do hlavnej vetvy produktu v príkazovom riadku použitím vhodných príkazov.

Celkovo som mal pozitívny dojem z práce s Git-om. Zdôraznil by som rýchlosť odovzdania zmien, čo vyplýva najmä z toho, že sa zmeny uložia lokálne, a netreba sa vždy pripojiť k serveru. Pri centralizovaných systémoch sa to vždy uskutoční, čo môže byť pre používateľa mierne otravné. Pri použití Git-u sa objavili aj jeho ďalšie výhody ako napríklad jeho nenáročnosť na pamäť, jednoduché použitie vetvenia a pokročilé riadenie konfliktov.

## Výber vhodného systému - najdôležitejšie aspekty

Pri výbere nového systému pre verziovanie, alebo pri prechode na nový systém treba zvážiť a identifikovať požiadavky tímu ktorý bude systém používať. Treba identifikovať taktiež znalosti a predchádzajúce skúsenosti členov tímu s podpornými prostriedkami. Na základe týchto údajov môžeme následne vybrať softvér, ktorý nám najviac vyhovuje. Jeden z najčastejších dôvodov, prečo sú členovia tímu proti prechodu na nový systém je ich pohodlie a odpor naučiť sa nové technológie. V takomto prípade nie je celkom jasné rozhodnutie, či sa oplatí riskovať úspech projektu nútením členov robiť so systémom s

ktorým nemajú skúsenosti. Manažér podpory musí mať v takom prípade závažné dôvody na zavedenie nového systému.

## Záver

Cieľom tejto eseje bolo stručné ale nie povrchné vysvetlenie úloh verziovacích systémov, porovnanie niektorých z nich a podeliť sa s vlastnými skúsenosťami s týmito systémami. Môžeme skonštatovať, že verziovacie systémy budú naďalej jednou z najdôležitejších spomedzi ostatných podporných prostriedkov, a ich vývoj bude pokračovať. V budúcnosti sa určite objavia ďalšie prístupy, ktoré možno nahradia súčasne známe centralizované a distribuované systémy. Tendencie však zatiaľ ukazujú čoraz väčšie presadenie distribuovaných systémov.

## Použitá literatúra

1. De Alvis, B., Sillito, J.: Why are software projects moving from centralized to decentralized version control systems? In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering (CHASE '09)*. IEEE Computer Society, 2009, 36-39.
2. Jotov, V.: An investigation on the approaches for version control systems. In *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing (CompSysTech '08)*, Boris Rachev and Anger Smrikarov (Eds.). ACM, 2008, Article 73.
3. Clifton, C., Kaczmarczyk, L.C., Mrozek, M.: Subverting the fundamentals sequence: using version control to enhance course management. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education (SIGCSE '07)*. ACM, 2007, 86-90.

## Annotation

*Support tools for version control: suitable choice for our team?*

*Software development today is a process that usually takes months, but sometimes several years. Such projects do not need only sufficient human resources, but also support tools to facilitate employees to co-operate and work effectively during working time. One of these tools is the version control system, which is responsible for cooperation, in particular with the source code. Choose from these funds is not as easy as it may seem, because there is a lot of systems like that. In this essay are mentioned the advantages and disadvantages, and some own experiences with this systems. The essay also deals with a suitable choice and with the way of the selection versioning systems. The reader should read the following to get a basic overview of this field and issues.*