

# SVN (subversion)

Subversion je FREE/Open-source nástroj pre správu a kontrolu verzií projektu. SVN sa stará o adresáre a súbory v stromovej štruktúre projektov. Hlavným dôvodom používania SVN je, že si pamätá históriu zmien súborov, dokáže identifikovať človeka, ktorý zmenu vykonal, poskytuje informácie o vykonanej zmene a je schopný vrátiť projekt do predošlých stavov. V SVN poznáme revízie, ktoré sú akýmsi číselným vyjadrením zmeny projektu.

Filozofia používania SVN sa opiera o sadu jednoduchých pravidiel. Subversion má svoju vlastnú stromovú štruktúru, ktorá nám pomáha definovať stav projektu. Rozlišujeme tri základné adresáre a to: trunk, branches a tags.

## Trunk

Adresár, v ktorom sa nachádza vývojová vetva projektu. Pri vývoji nových častí projektu intenzívne pracujeme s trunkom. Všetci developeri pridávajú svoju funkcionálnosť do trunku, riešia problémy, ktoré vznikajú, a testujú projekt. Po ukončení vývojových prác a deployovaní projektu na živý server sa z trunku vytvorí produkčná vetva (branch), ktorá sa nasadí do produkcie, pričom trunk zostáva v stave, v akom bol predtým.

Pri potrebe ďalšieho vývoja vývoj pokračuje v trunku a produkčnej vetvy sa pri programovaní nových funkcionalít nedotýkame.

Mnoho začínajúcich firiem a programátorov robí tú chybu, že trunk je zároveň aj ich produkčnou vetvou. Toto je zlé rozhodnutie, ktoré môže viesť a aj vedie k fatálnym problémom a chybám.

## Branches

Je adresár, ktorý obsahuje vetvy, ktoré slúžia buď na vývoj špeciálnej funkcionality, alebo sú tam produkčné verzie systému. Ak v trunku náš projekt dosiahne stav, v ktorom chceme spraviť release, vytvoríme branch, ktorý sa nasadí na živý server. Do tohto branchu následne robíme iba hotfixy, opravy chýb a prípadnú nutnú údržbu. Vývoj novej funkcionality prebieha v trunku.

## Tags

V tomto adresári sa nachádzajú tagy, ktoré sú ako branches, kópie kódu. Označujú určitý stav kódu, v ktorých sa projekt v čase tagovania nachádzal. Tagy sa využívajú aj na identifikovanie stavu pred určitou zmenou kódu a po nej.

## Užitočné príkazy a workflow

Pri práci s projektom, ktorý beží pod správou SVN, je nutné si projekt najskôr checkout-núť k sebe do počítača a vytvoriť si tzv. working copy. Na novovytvorenej working copy pracujeme, meníme ju - či už vývojom novej funkcionality alebo hotfixami. Podľa charakteru práce vytvárame aj working copy. V prípade, že ideme opravovať chybu, ktorá je v produkcii, tak vytvoríme working copy z release branch, ktorá sa nachádza v adresári branches. V prípade regulárneho vývoja naša working copy je kópia trunku z adresára trunk.

Po úspešnom ukončení našej práce musíme vytvorené zmeny odoslať do centrálného repozitára. Po úspešnom odoslaní všetkých zmien si tieto zmeny môžu premietnuť do svojich working copies aj ostatní členovia tímu pomocou príkazu update. Tento spôsob nám umožňuje bezproblémový tímový vývoj, aktuálnosť, prehľad i celkový prehľad o novej funkčnosti projektu.

### svn checkout (svn co) – Vytvorenie working copy

Príkaz nám zabezpečí vytvorenie (skopírovanie) projektu z centrálného repozitára. Na takto vytvorenom projekte môžeme následne pracovať. Naše zmeny kontroluje SVN, ktoré si pamätá zmeny v projekte. Checkout môžeme použiť na skopírovanie adresára, súboru i celého projektu.

```
svn checkout URL PATH
```

```
svn co URL PATH
```

### svn update (svn up) – Aktualizovanie našej pracovnej kópie

Pomocou príkazu svn update si premietneme všetky ostatné zmeny z centrálného repozitára a naša working copy sa stane aktuálnou. Update nám však neprepiše naše zmeny, ktoré už máme vytvorené, iba aktualizuje tie prvky projektu, ktoré sú na centrálnom repozitári a do našej working copy sa ešte nedostali.

Vždy pred začatím práce na projekte by sme mali spustiť príkaz svn update, aby sa všetky nové zmeny premietli aj k nám a predišli sme tak nežiaducim konfliktom.

```
svn update
```

```
svn up
```

### svn commit (ci) – Odoslanie našich zmien do centrálného repozitára

Príkaz commit slúži na odoslanie nami vykonaných zmien do centrálného repozitára. Takto odoslané zmeny si môžu update-núť iní členovia tímu.

Existuje nepísané pravidlo, že v repozitári sa má nachádzať iba a jedine funkčný kód. Pravidlo nám zaručí, že keď si iný člen tímu premietne posledné zmeny z repozitára do svojej working copy, jeho kópia projektu bude funkčná. Predchádzame tak situácii, keď jeden človek nemôže pracovať kvôli chybe iného.

Povinným údajom pri commite je commit message. Je to správa, ktorá sa pripojí k odoslaným zmenám. Táto správa sa nachádza pri čísle revízie, pod ktorou sa nachádza daný commit. Správy by mali byť jednoduché, výstižné a samozrejme vecne popisujúce obsah commitu.

```
svn commit -m "správa o commite"
```

```
svn ci -m "správa o commite"
```

## svn add – Pridanie adresára alebo súboru pod správu svn

Vývoj projektov častokrát so sebou prináša aj rôzne kopírovanie súborov. Ideálnym príkladom je nahadzovanie vizuálnej stránky projektu. Táto činnosť je väčšinou spojená s kopírovaním obrázkov a rôznych iných súborov. SVN však nevie, ktoré súbory boli pridané, a preto mu to musíme povedať.

Príkazom `svn add` pridáme chcený súbor alebo adresár pod správu `svn` v našej working copy. Pri následnom commite sa pridané veci odošlú do centrálného repozitára.

V prípade, že by sme spravili commit nových súborov bez predošlého spustenia príkazu `svn add`, súbory sa na server neodošlú a zostanú iba v našej lokálnej working copy.

```
svn add /cesta/k/suboru
```

## svn delete – Zmazanie súboru/adresára z centrálného repozitára

Obdobne ako v prípade predošlého príkazu, zmazaním súboru z našej working copy sa tento súbor neodstráni aj z centrálného repozitára. Spustením príkazu `svn delete` povieme SVN, aby po commite odstránilo označený súbor aj z centrálného repozitára.

```
svn delete /cesta/k/súboru
```

## svn status – Zobrazenie stavu

Pojmom status projektu rozumieme aktuálne zmeny, ktoré ešte neboli commitnuté. Príkaz `status` nám zobrazí stav nášho projektu. Zobrazí všetky súbory a adresáre, v ktorých bola vykonaná nejaká zmena. Pomocou tohto príkazu sme pred commitom schopní si jednoduchým spôsobom skontrolovať, čo všetko ideme commitnúť a čo všetko sa zmení aj na centrálnom repozitári.

```
svn status
```

Príkaz `svn status` obsahuje jeden prepínač, ktorý je dobré spomenúť, a síce `-u`. Prepínač nám zobrazí aj tie súbory a adresáre, ktoré sa budú na aktuálnej working copy meniť. To znamená, že na spomenuté prvky budú premietnuté zmeny z centrálného repozitára.

```
svn status -u
```

## **svn info – Zobrazenie základných informácií o checkoutnutej verzii projektu**

Svn info poskytne základný prehľad o informáciách, ktoré so sebou nesie daná working copy. To znamená adresu, z ktorej bola checkoutnutá, čas posledného commitu, meno človeka, ktorý vytvoril daný projekt, ...

svn info