Zdieľanie súborov v tíme

- nad jedným projektom reálne pracuje množstvo vývojárov
- potrebujeme sledovať zmeny
- potrebujeme mať zálohu projektu
- kopírovanie projektu cez USB kľúče nám asi nepostačí ;-)
- riešenie: **systém pre správu revízií** (revision control system)

Zdieľanie súborov v tíme

- prakticky používané množstvo systémov
- CVS, SVN, Git, Mercury, Bazaar, Source Safe...
- v open source svete najpopulárnejšie
  - **CVS** (Concurrent Version System)
  - **Subversion** (SVN)
  - CVS a SVN majú podobnú filozofiu, SVN opravuje drobné nedostatky
  - vidieť postupnú migráciu CVS -> SVN

Ako to funguje?

- **repository** na centrálnom serveri obsahuje aktuálny stav
- každý vývojár má vlastnú kópiu – **working copy**
- vývojár upravuje working copy a odosiela (**commituje**) zmeny na server
- zmeny od ostatných vývojárov preberá k sebe - **updatuje** si lokálnu kópiu

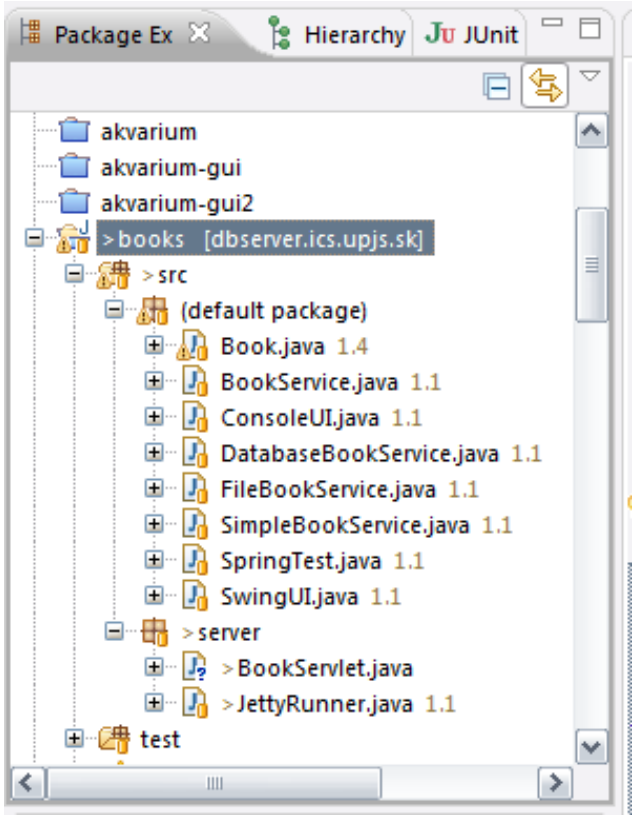
Praktické použitie

- CVS je priamo integrované v Eclipse
- SVN má plugin (Subclipse)
- použitie:
  - v perspektíve **CVS Repository Exploring** si pridáme úložisko. Vymámime od správcu
    - názov servera
    - cestu
    - meno a heslo

Praktické použitie

- nový projekt potrebujeme vytiahnuť z úložiska
- CVS Repository Perspective
- nájdeme si projekt, pravý klik naň
- **Checkout Project**
- projekt sa vytiahne z úložiska a nakonfiguruje v Eclipse

Príklad v Eclipse

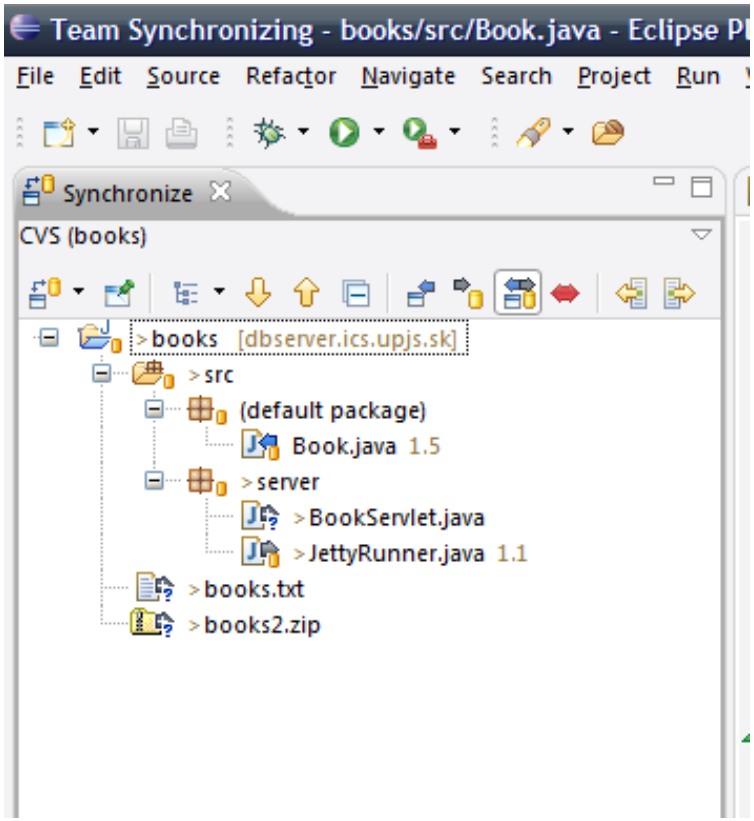


- projekt má v zátvorke názov servera
- každý súbor má číslo revízie
  - Book.java: 1.4 (štyri zmeny)
  - ConsoleUI: 1.1
- ikonka súboru má žltý valček – je pod správou CVS
  - budú sledované jeho zmeny
- otáznik – súbor nie je pod správou
  - príklad: BookServlet.java

Synchronizácia s úložiskom

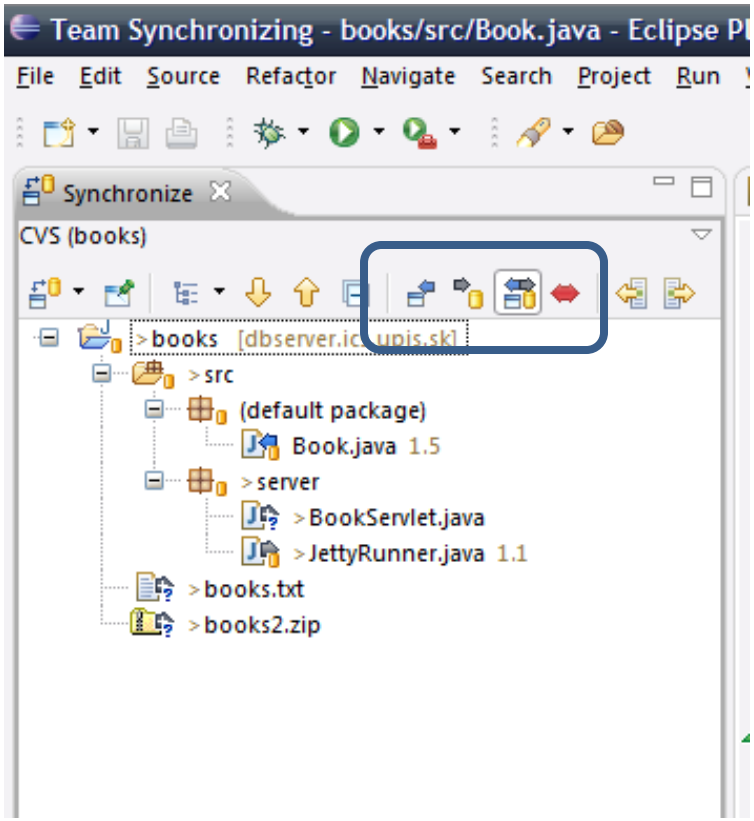
- čas od času je vhodné synchronizovať lokálnu kópiu s úložiskom
  - štandardne 1x za deň
  - v prípade komitu určite
  - ak je veľa zmien, nutné sledovať aj častejšie
- pravý klik na projekt, menu Team | Synchronize with Repository
- otvorí sa perspektíva Team Synchronizing

Príklad v Eclipse



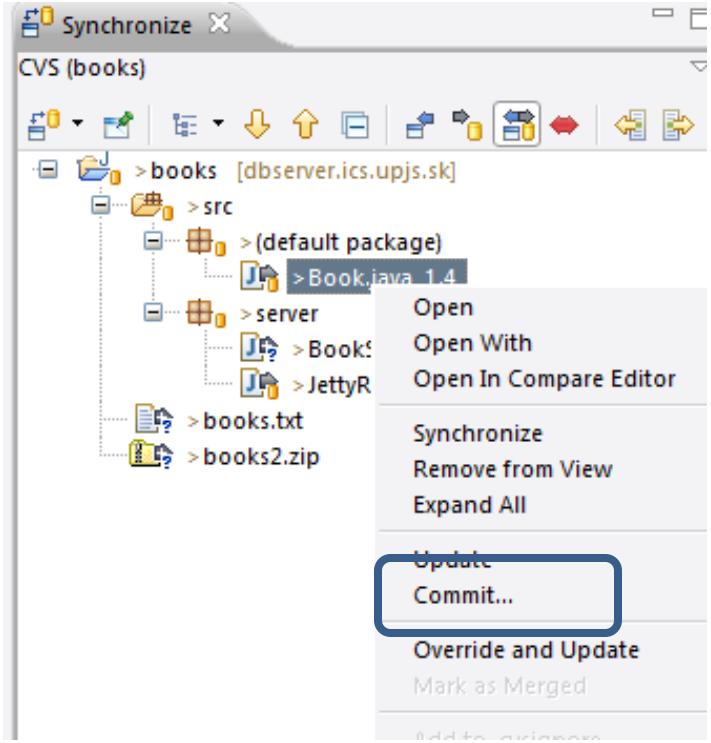
- obsahuje prehľad zmenených súborov
- ,,vľavo" = my, ,,vpravo" = server
- zmenený Book.java treba commitnúť na server (šípka vpravo)
- BookServlet
  - nie je pod správou (otáznik)
  - nie je na serveri (šípka vpravo, plusko)

Príklad v Eclipse



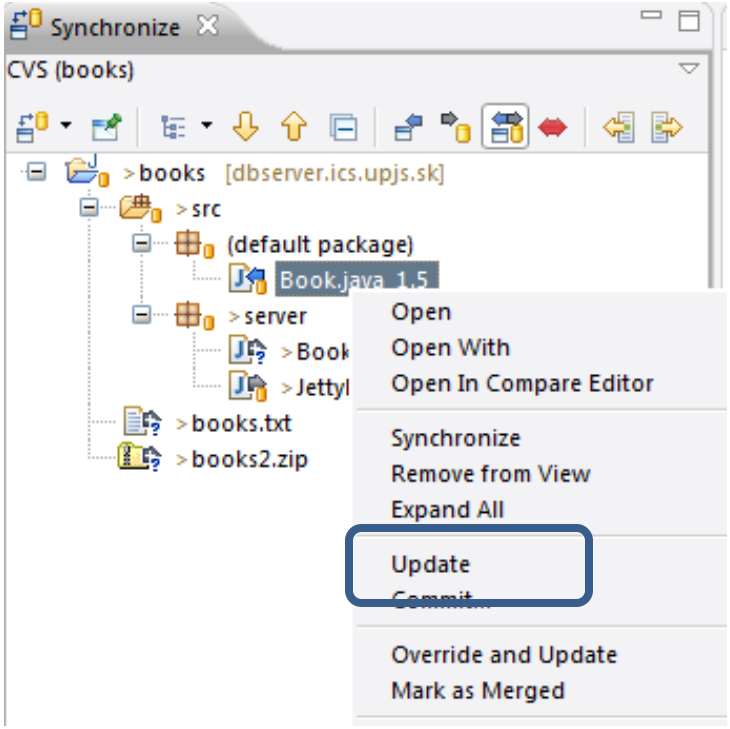
- štyri druhy pohľadov
- len prichádzajúce súbory
- len odchádzajúce
- oba predošlé
- konflikty (červená dvojšípka) – rozoberieme neskôr

Commitovanie v Eclipse



- pravý klik na súbor / adresár
- Commit... odoslanie na server
- v ďalšom dialógu vyplníme komentár o zmenách
  - komentár jednou vetou slovne zhŕňa zmeny
  - **"blablabla"** nie je komentár
  - **"pridanie nového textfieldu"** je komentár
- finishom sa dáta odošlú

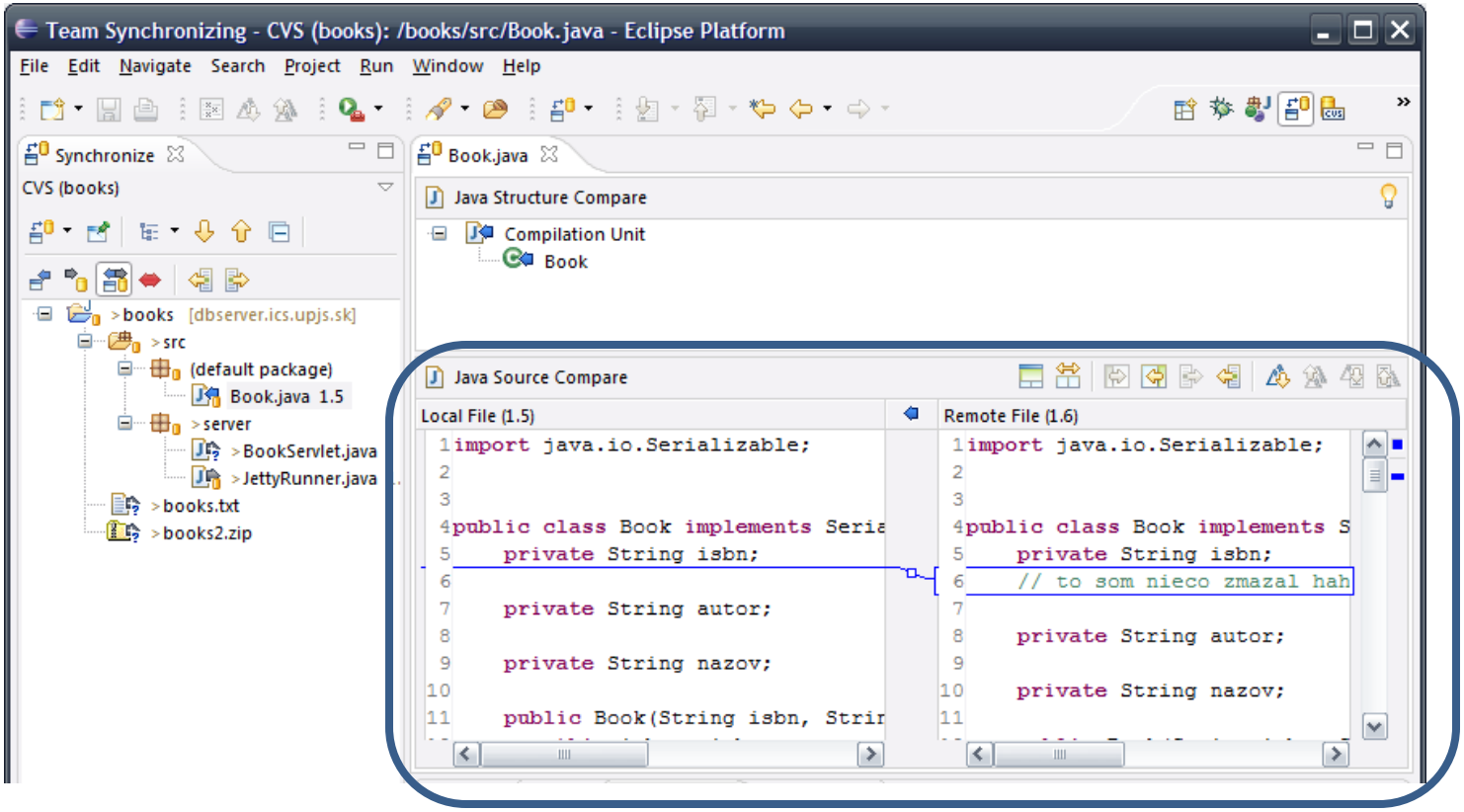
Update v Eclipse



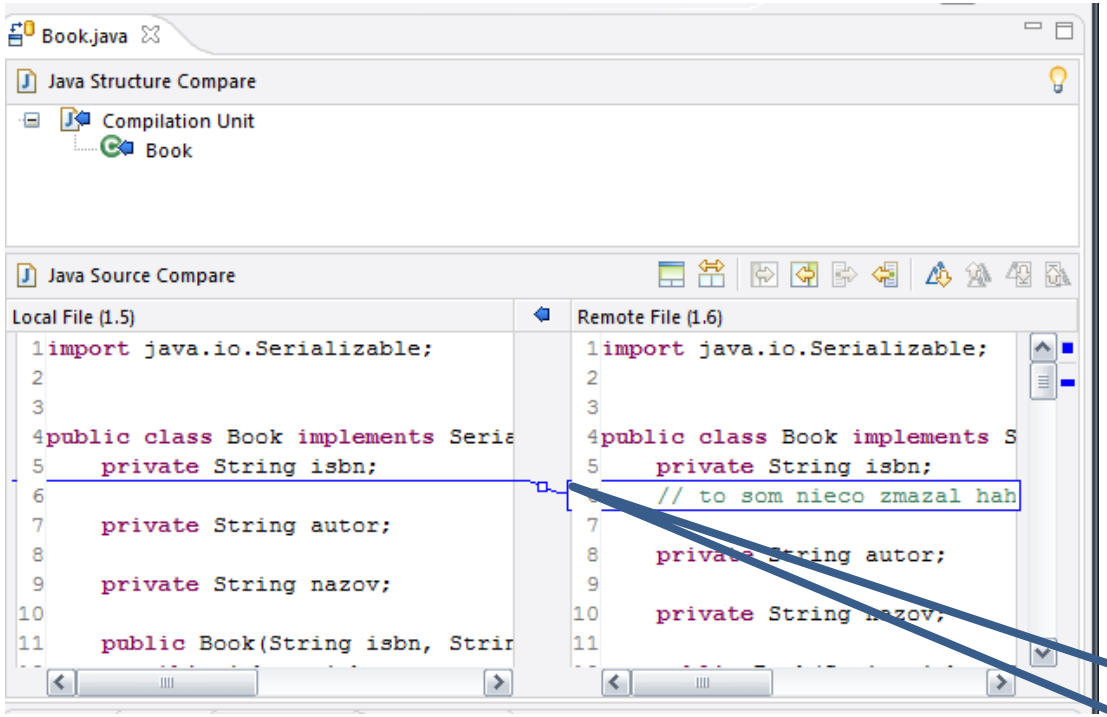
- pravý klik na súbor / adresár
- Update... získanie verzie zo servera
- stiahne sa aktuálna verzia
- v lokálnej kópii nesmú byť zmeny (inak nastane *konflikt*)

Porovnávanie súborov

- dvojkliknutím na súbor môžeme vidieť *diff*
- zoznam rozdielov v súboroch



Porovnávanie súborov



- vľavo: lokálny súbor, vpravo: vzdialený
- zvýraznené zmeny v riadkoch
- kliknutím na gombík môžeme kopírovať jednotlivé zmeny sprava doľava

Zhrnutie operácií

- Príklad: stiahneme zo servera revíziu 1.4
- vykonáme zmeny u seba
- **commit**neme:
  - na serveri je verzia 1.5
  - u nás je verzia 1.5
- ideme sa vyspať
- v noci niekto komitne novú verziu
  - na serveri je verzia 1.6
  - u nás je verzia 1.5
- ráno **update**neme => u nás je verzia 1.6

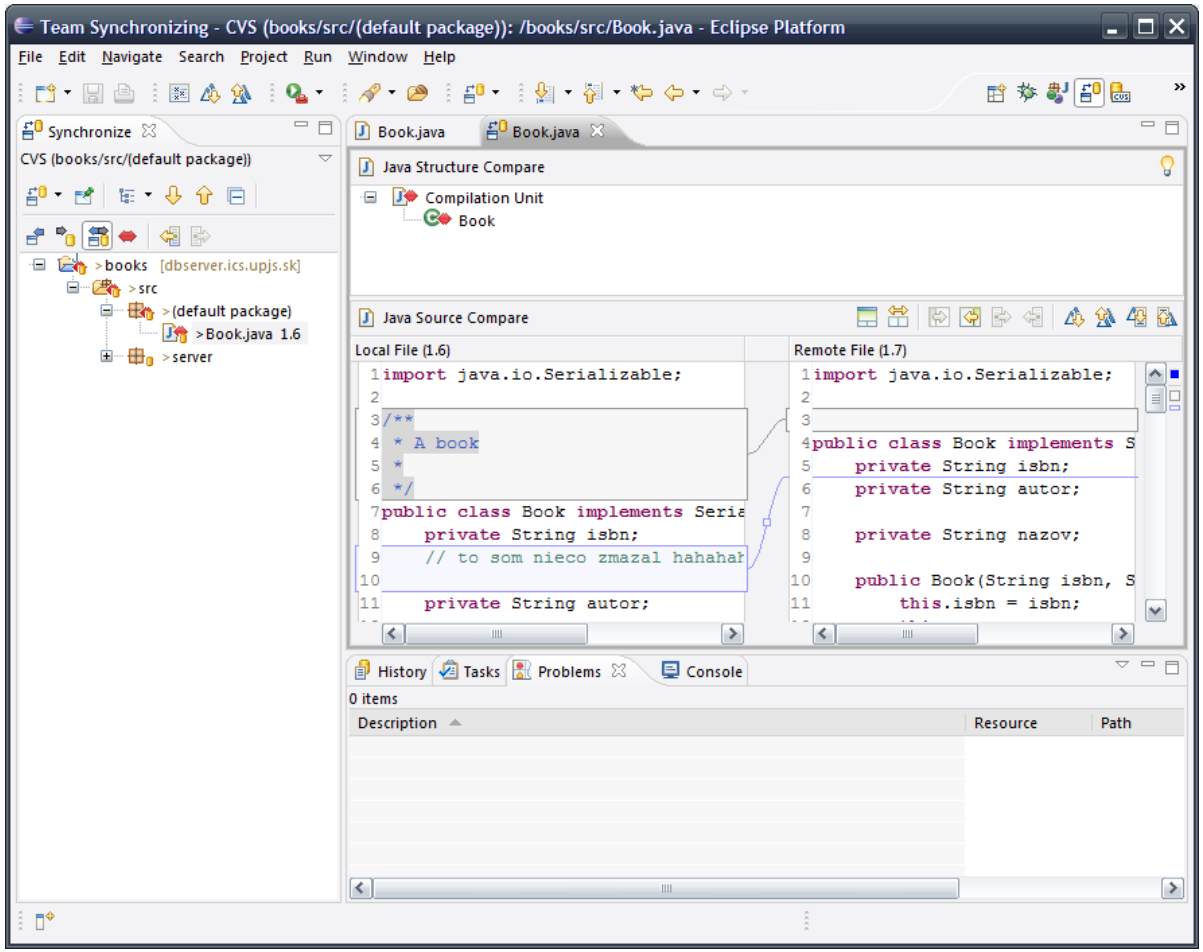
Konflikty

- čo keď viacerí pracujú nad tým istým súborom?
- niektoré systémy podporujú **uzamykanie**
  - stiahnem si súbor
  - uzamknem ho, aby mi ho niekto neprepisoval
  - po dokončení prác komitnem a tým ho odomknem
- CVS/SVN však toto neriešia
- je to jednoduchšie na prácu, ale môžu nastať sporné situácie

Commit a konflikt

- Príklad: stiahneme zo servera revíziu 1.4
- vykonáme zmeny u seba
- **commit**neme:
  - na serveri je verzia 1.5
  - u nás je verzia 1.5
- kým obedujeme, niekto komitne novú verziu
  - na serveri je verzia 1.6
  - u nás je verzia 1.5
- vrátime sa z obeda, upravujeme vlastnú verziu 1.5, ktorú chceme komitnúť
- máme konflikt:
  - commitom chceme našu verziu povýšiť na 1.6
  - lenže na serveri už 1.6 existuje

Commit a konflikt



- Ako sa prejavujú konflikty?
- Dve červené šípky.
- Zvýraznenie v *diffe*

Commit a konflikt

- Ako riešiť konflikty?
- Dohovorom
  - sadneme si s autorom konfliktnej verzie
  - v jednej verzii zlúčime zmeny (**merge**)
  - následne verziu označíme za zlúčenú
    - pravý klik na súbor, **Mark as Merged**
  - a komitneme

Commit a konflikt

- Ako riešiť konflikty?
- Ak vidíme, že verzia na serveri je lepšia ako naša
  - pravý klik na prichádzajúci súbor v Synchronize
  - **Override and Update** – prepíše naše zmeny verziou zo servera

História

- všetky commity sa evidujú
  - pridania
  - úpravy
  - zmazania
- hocikedy si vieme vytiahnuť hocijakú verziu súboru
  - aj zmazaného
- prehľad histórie: pravý klik na súbor, Team, **Show in History**
  - zobrazí históriu commitov s dátumami, autormi a commit komentármi

Ďalšie vlastnosti

- CVS/SVN poskytuje kopu ďalších vlastností
- branche
  - ak sa súčasne pracuje na viacerých verziách projektu
  - opravujeme vetvu 1.X
  - a zároveň vyvíjame verziu 2.X
- anotácie
- značkovanie...
- viď dokumentácia