

SVN

ja sorsan kunnossapito

Ongelma

- Missä ja miten säilyttää lähdekoodi niin, että
 - se on kaikkien saatavilla,
 - uusin versio on aina saatavilla,
 - vanhatkin löytyvät.

Ratkaisuja?

Jaettu hakemisto

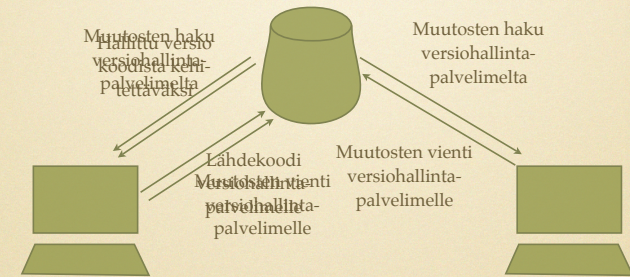
- Idea: yhteinen hakemisto, joka sisältää kaiken lähdekoodin
- Kaikki koodi kaikkien saatavilla
- Entäs kun kaksi henkilöä muuttaa samaa koodia yhtäaikaan?
- Mistäs löytyisivät vanhat versiot?

Uudet versiot ilman versiohallintaa



Vanhaa tiedostoa et takaisin saa, ellet sitten tallenna jokaista tiedostoa talteen. Tällainen toiminta olisi kyllä hankalaa ja tarpeetonta.

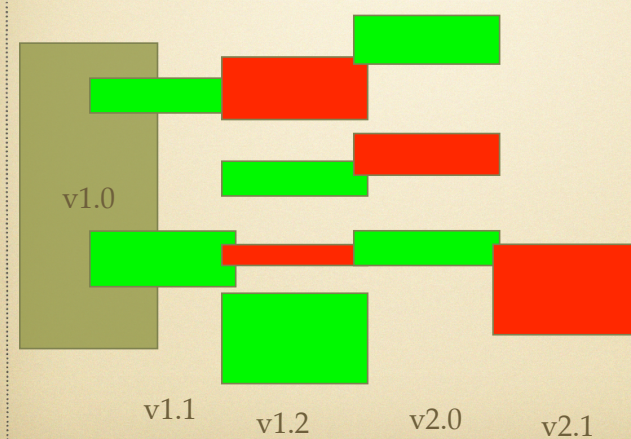
Versiohallinta



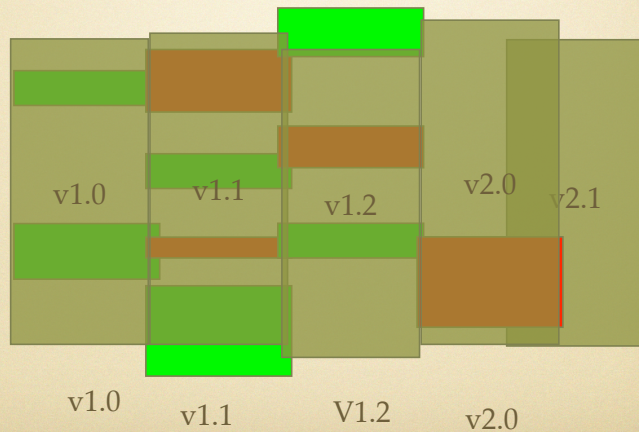
Versioiden tulisi säilyä



Säilötään vain muutokset



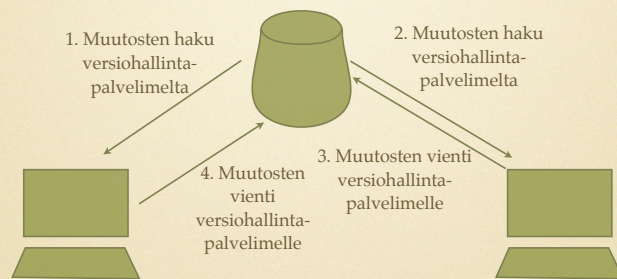
...mutta käänteisesti



Entäs samanaikaiset muutokset?

- Joko kielletään:
 - muokattava tiedosto lukitaan, jolloin vain yksi kehittäjä muokkaa yhtä tiedostoa kerrallaan
- Tai sallitaan:
 - konfliktitilanteita sattuu harvoin: svn merkitsee konfliktit, eikä suostu jatkamaan, ennen kuin ne on korjattu

Konflikti



Jos muutokset nyt kohdistuvat samoihin koodiriveihin, syntyy konfliktitilanne.

Konfliktiesimerkki

Varastossa:

Kissa istuu
ja syö juustoa.

Meidän versio:

Kissa istuu
puussa
ja syö juustoa.

Muiden versio:

Kissa istuu
kuussa
ja syö juustoa.

Konfliktiesimerkki

Kissa istuu

<<<<<<< .mine

puussa

=====

kuussa

>>>>>>> .r3

ja syö juustoa.

Oma, paikallinen versio

Muualta tullut versio

Kuis käytetään?

1. Tallennetaan ensimmäinen versio varastoon
2. Haetaan muutokset / uusin versio varastosta
3. Muokataan
4. Tallennetaan muutokset varastoon
5. Toistetaan kohdasta 2., kunnes sovellus valmis

Mitä käytetään?

- RCS – vanha sotaratsu
- CVS – luotettava työhevonen
- SVN, eli Subversion – työhevonen
- GIT – trendikäs poni
- darcs, mercurial, M\$, Rational... – konit, aasit, muulit, pegasukset, troijalaiset ja yksisarviset (riippuen puhujasta)

Subversion, SVN

- Yksi tämän hetken suosituimmista versionhallintaohjelmistoista.
- Syntyi, kun CVS ei riittänyt kaikille.
- <http://subversion.tigris.org/>
- Kaikki oleellinen ohjeistus löytyy kirjasta Version Control with Subversion:
<http://svnbook.red-bean.com/en/1.4/>

Alustus

- Alustus (vain kerran):
`svnadmin create /polku/luotavaan/varastoon`
- ATK-tuki luo projektia varten teille omat varastot, mutta kannattaa testailla ja opetella käyttöä omalla tietokoneella sijaitsevall varastolla.

Osoite

- Varaston osoite:
 - `file:///polku/luotavaan/varastoon`
 - `file:///X:/polku/m$dos/käyttiksissä`
 - `svn+ssh:///svn.cc.jyu.fi/srv/svn/testi`

Ensimmäinen tallennus

- Aja hakemistossa, jossa sorsa sijaitsee komento:
`svn import -m "Ensimmäinen versio" sovellus file:///varaston/polku/projektin_nimi`
- Aina tallennuksen yhteydessä tulee kirjoittaa mukaan lyhyt, selkeä kuvaus tallennuksen syystä (mitä ja miksi muutettu, viittaus buginumeroon, ...). Ei jätetä kuvausta tyhjäksi!

Paikallisen kopion hakeminen

- Poistu projektihakemistosta
- Muuta projektihakemiston nimi, esim. proj → proj.vanha
- Aja komento:
`svn checkout file:///varaston/polku/projektinimi`
- Voit myös lyhentää checkout → `co`

Varaston rakenne

- projektin-nimi/
trunk/ (pääkehityshaara)
tags/ (jäädytetty koodi)
branches/ (muut kehityshaarat)
- SVN ei pakota rakennetta, mutta ylläolevan mukainen rakenne on muodostunut yleiseksi käytännöksi.

Muutosten haku

- Aja projektihakemistossa komento:
svn update
- HUOMAA! Update on ajettava *aina* ennen kuin muokkaat tiedostoja tai viet niitä varastoon.

Eka tallennus

- Aja komento:
svn commit -m 'Muutoksen kuvaus'
- Muista järkevä kommentti!
- Ei tyhjää kommenttia!
- commit lyhenee ci

Uudet tiedostot

- Ensin:
svn add -m 'viesti' tiedostot
- Sitten:
svn ci -m 'viesti'

Tehdyn tarkastelua

- Lähteen tila:
svn status
- Erot kehitysversion ja varaston uusimman välillä:
svn diff tiedosto.cxx
- Loki:
svn log

Konfliktin ratkaiseminen

- Konfliktissa olevasta tiedostosta syntyy kolme uutta tiedostoa: *tiedoston_nimi.mine*, *tiedoston_nimi.rVANHA* ja *tiedoston_nimi.rUUSI*.
- *.mine* sisältää työkopiassa muokatun version
- *.rVANHA* sisältää version ennen konfliktoivia muokkauksia (VANHA - vanha versionumero)
- *.rUUSI* sisältää juuri varastosta saadun version (UUSI - uusi versionumero)

Lähde: Tuomaisen Maunun kalvot

Konfliktin ratkaiseminen

- Muuta varsinainen tiedosto kuntoon
- Aja: `svn resolved tiedoston-nimi`
- Aja: `svn commit`
- Joissain tapauksissa konfliktin *oikea* ratkaisu on vanhan tai uuden version kopiointi työkopion päälle, mutta usein täytyy tehdä käsityötä ja sovittaa konfliktoivat muutokset yhteen

Lähde: Tuomaisen Maunun kalvot

Versionumerot

- Joka kerta, kun varaston tila muuttuu (esimerkiksi commitin seurauksena), sen versionumero kasvaa
- Kaikki varaston tilan muutokset tallentuvat ja mihin tahansa vanhaan versioon voidaan palata
- checkout-komento voidaan parametrizoida esimerkiksi tietyn ajankohdan mukaan, jolloin päästään historiatietoihin käsiksi

Lähde: Tuomaisen Maunun kalvot

Apuva!

- svn help
- svn help komento
- Ohjaajat, naapurit ja RedBook

Linkkejä

- <http://subclipse.tigris.org/> – Subversion-liitännäinen Eclipseen
- <http://subversion.netbeans.org/> – Subversion Netbeansin kylkeen
- <http://www.emacswiki.org/cgi-bin/wiki/SubVersion> – Subversion ja Emacs
- http://subversion.tigris.org/tools_contrib.html – Lisää työkaluja

...ja vielä

- <http://www.mit.jyu.fi/palvelut/sovellusprojektit/luennot/subversion.pdf> – Luento Subversionin käytöstä Sovellusprojektien sivuilta.
- Käytetty apuna tätäkin esitystä päivitettäessä.
- Aijoo: thk:n trac/svn saattaa kysellä salasanaa ensimmäisellä kerralla seitsemänkin kertaa! Koettakaa jaksaa naputella :)

Jotta salaus toimisi

- Avainpari on luotava: ssh-keygen
- Julkinen avain on kopioitava koodivarastokoneeseen:
scp oma.avain mie@cvcs.cc.jyu.fi:.ssh/authorized_keys
- toki onnistuu ilmankin, mutta on ikävä kirjoitella jatkuvasti salasanaa
- SSH:n ssh-agent toinen vaihtoehto
- ...ja VPN