Tecnologie Open Source

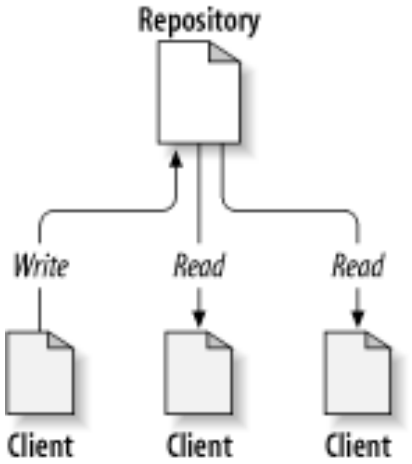
Subversion

Materiale di riferimento

- Version Control with Subversion
  - Rilasciato sotto licenza CC all'indirizzo: http://svnbook.red-bean.com/
- Pragmatic Version Control using Subversion (di Mike Mason)

Version Control System

- Cosa è?



- Multiple revisioni
- Vantaggi rispetto ad altre soluzioni:
  - File server comune
  - Backup periodici
  - Gestione di revisioni e rami

Version Control System

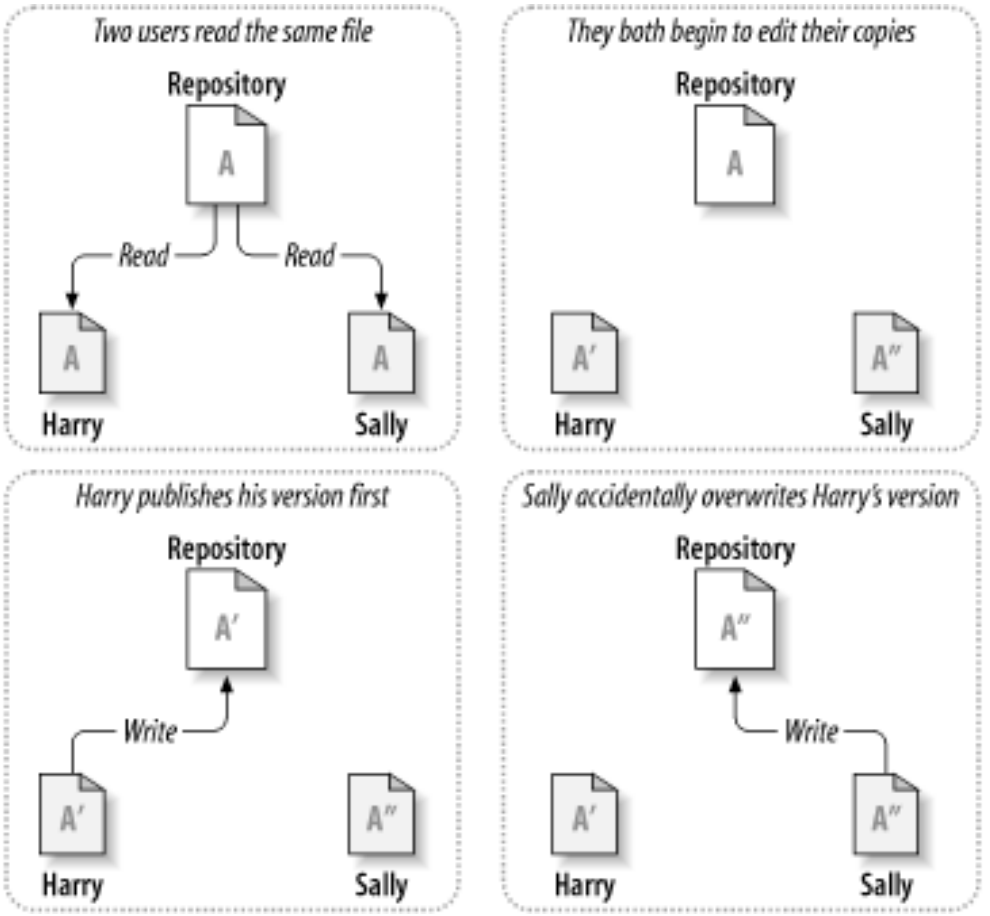
- Termini importanti:
  - Checkout e working copy
  - Export
  - Commit
  - update
  - Revisioni
- Funzionamento delle revisioni:
  - Per repository
  - Collegate al singolo file
  - Tag e branches

Repositories

- Un repository, n progetti
  - Una cartella per ogni progetto
  - Le ramificazioni sono memorizzate come copie di cartelle
  - Possibile perché la copia in subversion è molto efficiente e **mantiene la storia**
- Un repository non contiene una copia diretta dei progetti ospitati
- Accessibile via molteplici interfacce: http, https, svn, svn+ssh, file

Locking: il problema

- Il problema



Locking: la soluzione lock-modify-unlock

- Lock attivato sui singoli file
- Problemi:
  - Amministrativi
  - Serializzazione non necessaria
  - Falsa sicurezza
- Subversion supporta questa soluzione ma preferisce il locking ottimistico

Locking: la soluzione copy-modify-merge

- Locking ottimistico:
  - Si possono inserire nel repository solo file non out of date
  - Al momento dell'aggiornamento subversion prova a fondere le modifiche remote o locali
    - Fusione
    - Conflitto

Revisioni e file

- Subversion memorizza per ogni file
  - La revisione del repository su cui è basato
  - Un timestamp di quando è stato aggiornato da repository l'ultima volta
  - I file originali prelevati dal repository
- Un file può essere
  - Inalterato localmente e aggiornato
  - Alterato localmente e aggiornato
  - Inalterato localmente e non aggiornato
  - Alterato localmente e non aggiornato

Struttura di un repository

- Conf: configurazione del repository
  - Svnserve.conf
  - Passwd
  - Authz
- db: le cartelle del repository
- Hooks: gli script agganciabili agli eventi
- README.txt
- format

Ciclo fondamentale

- checkout
- Ciclo base
  - Aggiornamento
  - Commit
  - Update (su file inalterati o alterati)
- Updates frequenti
- Push e pull sono separati

Revisioni miste

- Subversion supporta le revisioni miste
- Come si creano?
- Svn log “non funziona”
- Come ottenere le revisioni dei file del nostro progetto
  - Con svn info
  - Con svn status
- Limitazione: delete, commit di metadata e merge possibili solo su file o cartelle aggiornate

Valori simbolici di revisione

- HEAD: l'ultima revisione
- BASE: la revisione corrente di un file in un workspace
- COMMITTED: la revisione precedente a BASE in cui il file è stato modificato
- PREV: COMMITTED-1
- Revisioni per data: {2012-01-03},{15:30}

Popolazione del repository

- Inserire i file nel repository con import
  - Esempio
  - Cosa succede se il path esiste già
  - La cartella usata per l'import non viene posta automaticamente sotto version control
    - svn-commit.*.tmp e svn commit -F

Creazione di un repository

- Specifica delle modalità di accesso al repository
- Repository del tipo file://....
  - Svnadmin create file://...
- Creazione di un repository svn://...
  - Uso di svnserve
    - -d
    - -r
  - Configurazione di un repository

Manipolazione del repository manualmente

- Svn add path
  - --non-recursive
- rm
  - Svn rm path1 path2 ...
  - Svn rm url1 url2 ...
- mkdir
  - Svn mkdir path1 path2
  - Svn mkdir url1 url2

Manipolazione del repository manualmente

- Mv
  - Svn mv path1 path2
  - Svn mv url1 url2
- Cp
  - Svn cp path1 path2
  - Svn cp url1 url2
- Prova su strada

Manipolazione del repository manualmente 2

- cat
  - Svn cat path/url
  - Svn cat -r rev path/url
- ls
  - Svn ls path/url
- Info
  - Svn info path/url
- Prova su strada

changelist

- Cosa sono le changelist
  - Applicate alla working copy, non al repository!
- Aggiungere un file in una changelist con svn changelist file
- Rimuovere una changelist da un file
- Restringersi a una changelist con –changelist
- Prova su strada

export

- Svn export: utile per creare archivi “puliti”
  - Svn export [-r rev] url[@pegrev] [path]
  - Svn export [-r rev] path[@pegrev] [path]
- Digressione: operative revisions e pegrevisions
  - Operative revision <= peg revision
  - Operative revision > peg revision
- Valori di default per le peg revisions

Controllo delle proprie modifiche

- Svn status [-u] [-v] [path]
  - Flag: A,D,M,C,R,X,I,?,!,L
- Svn log [path]
  - -r r1
  - -r r1:r2 [default: BASE:1]
  - -v
  - --stop-on-copy

Usiamo diff

- Svn diff
  - Confronta le modifiche locali apportate dall'ultimo update
- Svn diff file.txt
  - Come prima ma si limita a file.txt
- Svn diff -r 5:6 [file.txt]
  - Mostra le differenze tra la revisione 5 e 6 [eventualmente del file]
- Svn diff url1 url2: mostra le differenze tra due url

Annulliamo le nostre modifiche

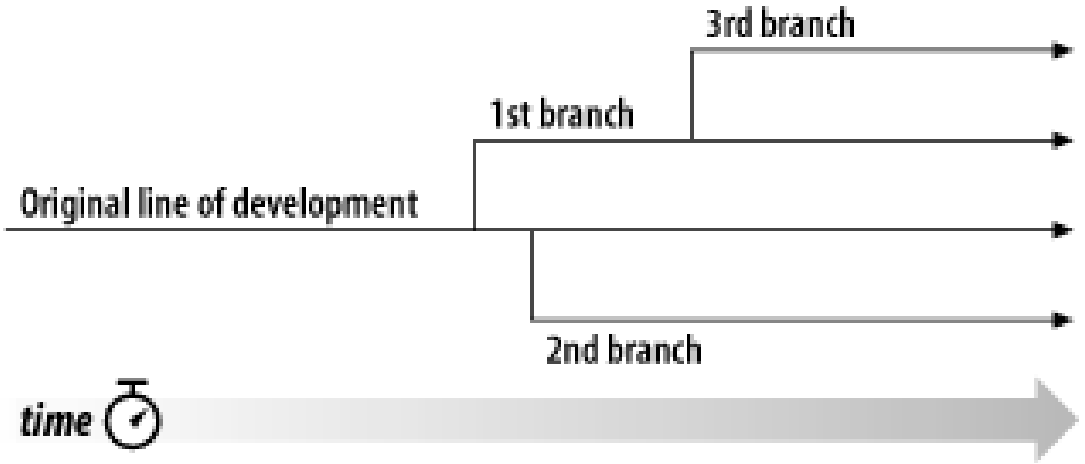
- Svn revert path
  - --recursive
  - Funziona anche con l'aggiunta e cancellazione di file
- Svn revert e svn update -r BASE
- Digressione: diff, status, revert e accesso alla rete

Conflitti

- Conflitti:
  - Conflitti su file
  - Conflitti d'albero
  - Conflitti di proprietà
- Svn update in caso di conflitti
- Prova su strada
- Rimuovere manualmente lo stato di conflitto da un file
- Risolvere conflitti d'albero: esempio

branches

- Le branches sono ramificazioni di un progetto che servono per:
  - Release stabili
  - Bug fix complessi
  - Sperimentazioni
  - altro



tags

- I tag sono nomi simbolici per specifiche selezioni di revisioni dei file del progetto
- Servono per rappresentare:
  - Release (es. REL-1.0.1,REL-2.0.3)
  - Inizio fine di un bug fix complesso
  - Inizio fine di una sperimentazione
  - ...

Branches e tags in subversion

- Branches e tags sono semplici copie di cartelle
- Tecnicamente identiche, sono distinte dal loro uso:
  - Le branches vengono modificate nel tempo
  - Le tag no
- Struttura di un repository:
  - Trunk
  - Branches
  - tags

Esempi

- Come creare una ramificazione
- Come creare un tag
- Esempi:
  - Una parte del team smette di lavorare temporaneamente su trunk e lavora per la stabilizzazione della release stabile
  - Dopo un po' viene rilasciata la release 1.0.0
  - Si vogliono trovare le modifiche fatte alla branch stabile

Esempi

- Un bug fix trovato nella branch stabile deve venire applicato anche a trunk
  - Svn merge ^/trunk con merge tracking
  - Svn merge [-r r1:r2] source
  - Svn merge source1 source2
- Si vuole rimuovere un commit sbagliato
- Si vuole ripristinare un file cancellato per errore (e committato)

Switch

- Svn switch
- Switched items

Settare i metadata

- Aggiungere, togliere, visualizzare i metadata
  - Svn propset [-F]
  - Svn propget
  - Svn propedit
  - Svn proplist
  - Svn propdel

Metadata speciali

- Svn:mime-type e la gestione dei conflitti
- Svn:eol-style
  - Il problema del fine linea
  - Valori possibili
    - Native
    - CRLF
    - LF
    - CR
- Svn:ignore e l'opzione --recursive

Inserimento automatico di proprietà

- Il file di .subversion/config
  - editor-cmd
  - diff-cmd
- enable-auto-props=yes
- Sezione [autoprops]: *.cpp= p1=v1;p2=v2;...
- Esempio: attiviamo svn:eol-style su tutti i file cpp

Locking esplicito

- Svn lock
  - -m
- Svn unlock
  - --force
- Lock stealing
- La proprietà svn:needs-lock
- Prova su strada

Backup del repository

- Svnadmin hotcopy
- Svnadmin dump
- Svnadmin load
- svndumpfilter
- Migrare un progetto a un nuovo repository