

Versionshantering

Eric Elfving

Institutionen för datavetenskap (IDA)

Har du någon gång använt "spara som..."?

- Versionshantering är bra till både för små och stora projekt
- Ger en ökad säkerhet:
 - Skydd mot datakorruption (genom dubblett av data)
 - Skydd mot dåliga ändringar i kod (går att gå tillbaka i historiken)
- Gör det lättare att sammarbeta:
 - Alla kan jobba var för sig och sedan sammanför systemet arbetet

Vanliga operationer

- Check out: Skapa en lokal kopia av reposet
- Commit: spara mina ändringar
- Update (sync): hämta eventuella ändringar
- Revert: ångra ändringar och gå tillbaka till föregående version
- Add: Lägg till en lokal fil i repos
- Delete: ta bort en fil från repos

Server – klient

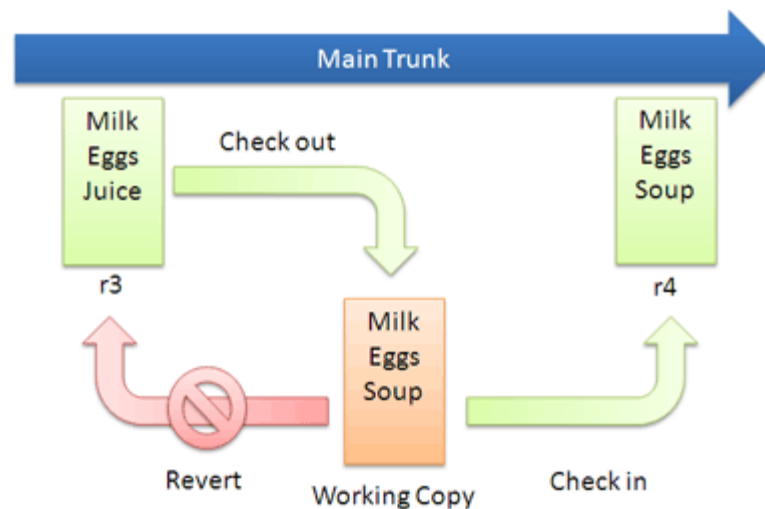
- Servern håller koll på all information
- Klienterna hämtar filer från servern, gör lokala ändringar och skickar dessa ändringar till servern
- Servern håller koll så att det inte blir krockar
- Kan bli väldigt problematiskt om servern dör
- Bara de som har tillåtelse att skriva får göra ändringar
- Alla som har tillåtelse att skriva får göra ändringar

Distribuerade system (DRCS)

- Arbetar med en peer-peer modell
- Alla som tar ut en kopia får göra vilka ändringar de vill
- Ändringarna delas med andra
- Vissa förutbestämda individer bestämmer vad som ska finnas i huvudgrenen av projektet
- Fungerar generellt bättre till större projekt (t.ex. linuxkärnan)

Subversion

- Aktivt projekt som drivs av Apache
- Är av typen server-client
- Alla commits är fullt atomära
- Endast ändringar skickas (åt båda hållen) vilket betyder att tidsåtgången beror på antal ändringar och inte kodstorlek
- Versionshanterar filer, länkar, mappar och namnbyten effektivt



Subversion

- Kan sättas upp på flera sätt men lättast är en lokal katalog:
`svnadmin create "path till mappen"`
- Åtkomst sker med kommandot `svn`. En bra start är import
`svn import "mapp" file://"path till repos"`
 - Importerar "mapp" rekursivt till reposet
- När man satt upp och importerat ett repos kan andra skapa en egen kopia med `co`:
`svn co file://"path"`

Subversion

Vanlig arbetsordning

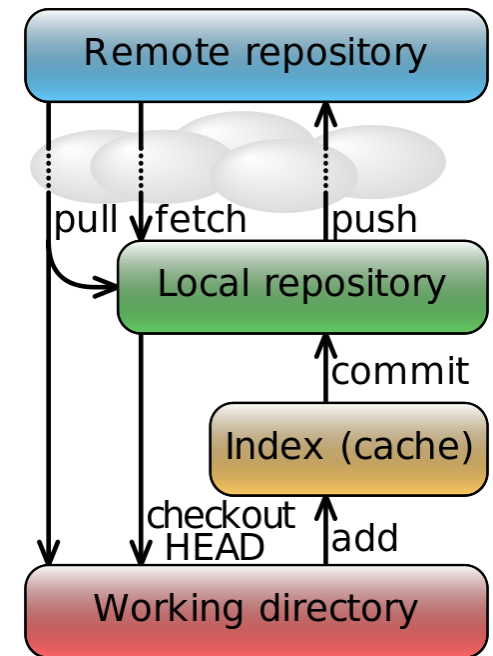
- Man inleder ofta sitt arbete med en sync
`svn sync`
- Därefter börjar man koda
- När man är klar gör man en commit
`svn ci -m "meddelande"`
- Om det inte fanns krockar (t.ex. någon annan ändrade nyss på samma fil) läggs ändringarna in som en ny version

Subversion

- Väldigt lätt att få hjälp:
 - `svn help [kommando]`
 - Google: välkänt och välanvänt

Git

- Skapades av Linus Torvalds för att hantera linuxkärnan
- Ett distribuerat system
- Varje lokal version har all ändringshistorik
- Andras ändringar kan importeras som andra grenar



Git

- Skapa ett nytt projekt med init i den mapp som ska vara root för projektet

`git init`

- Lägg till filer och mappar med add

`git add *.py`

`git commit`

- Andra kopierar ditt projekt med clone

`git clone [url]`

GitHub

- Man kan installera en egen server för att hantera kod
- GitHub låter dig skapa projekt gratis (om open-source)
 - Man får både ett Git-arkiv samt massor av bra verktyg för att hantera sitt projekt såsom wiki, statistik och webbaserad kodgranskning

Git

- Läs mer!
 - Man-sidorna
 - <http://git-scm.com/documentation>
 - <https://github.com/>



Linköpings universitet

expanding reality

www.liu.se