

DVCS - Distributed Versions Control System

Cos'è un DVCS?

Come dice la sigla, un DVCS è un sistema di revisione distribuito. Questo significa che tutti i programmatori appartenenti ad un progetto hanno la possibilità di lavorare in locale sul proprio codice e sincronizzarsi in ogni momento con gli altri membri del gruppo. Nascono come soluzione ai problemi che presentano i tradizionali sistemi di revisione centralizzata, come CVS e Subversion. Se i progetti di sviluppo sono troppo grandi e presentano vari branch di lavorazione parallela, i CVS incontrano numerosi intoppi nella gestione dell'intero codice. Ecco perchè si è ricorso a questi nuovi metodi di gestione del codice, sviluppati in modo tale che ogni programmatore lavori sul proprio branch locale e non incontri difficoltà nel *merge* dello stesso al filone principale.

Git,Bazar,Mercurial

Git è una collezione di script shell e di programmi in C scritta originariamente da Linus Torvalds. Lo strumento utilizza un'interfaccia a linea di comando ma è anche incluso un front-end grafico (gitk)

per visualizzare il grafo delle revisioni. Per questo sistema di repository si è data molta importanza alle prestazioni già in fase di progettazione rendendo il tutto più performante possibile. E' l'unico strumento che prevede comandi espliciti per la manutenzione del repository (una specie di garbage collection). Come IDE Java, Git è supportato da Eclipse e IntelliJ IDEA.

Mercurial è scritto in Python con alcune parti in C. Fra gli utilizzatori vi sono i progetti NetBeans, OpenJDK, OpenSolaris e Mozilla. L'interfaccia è tramite comandi shell, che sono pochi, intuitivi e con parametri semplici. Questa semplicità non va a scapito della funzionalità, che è completa per le operazioni fondamentali. Un potenziale limite dello strumento è il fatto che non mette sotto revisione le directory vuote. Anche Mercurial è supportato da Eclipse, NetBeans (integrato in 6.1) e IntelliJ IDEA.

Bazaar è scritto in Python ed ha come principale produttore e sponsor Canonical, l'azienda che produce *Ubuntu*. La principale differenza rispetto agli altri DVCS è che supporta vari modelli di collaborazione (detti workflow), fra cui quello centralizzato e

decentralizzato, con o senza un gatekeeper con ruolo privilegiato. Anche qui l'interfaccia è tramite comandi shell. I comandi di base sono facili ma un po' più complicati se si usano le funzionalità più avanzate. Bazaar è supportato da Eclipse e IntelliJ IDEA.

Il Confronto

Per quanto riguarda le prestazioni o la documentazione tutti questi strumenti hanno raggiunto un ottimo livello. Git ha troppi comandi, troppi parametri e troppa terminologia. Bocciato per ora!

Fra Mercurial e Bazaar la scelta è più difficile. Mercurial ha un 'appeal' immediato dovuta ai comandi eccezionalmente intuitivi. Ma Bazaar ha delle feature in più, prima fra tutte la possibilità di scegliere fra workflow di collaborazione diversi.

Per riassumere, Mercurial è più usabile, mentre Bazaar ha più feature e un più solido design. E' difficile dare un giudizio obiettivo, anche perchè ogni programmatore ha i suoi modus operandi quindi si potrebbe trovare meglio con uno rispetto che con l'altro, quindi l'unico consiglio che si può dare in questi casi è provare e provare!

Description	BZR	GIT	HG
<i>init a new repo</i>	bzr init	git init	hg init
<i>check working tree status</i>	bzr status	git status	hg status
<i>ignore some files</i>	bzr ignore PATTERN	echo "PATTERN" >> .gitignore	echo -e "syntax: glob\n*~" > .hgignore
<i>add / remove / rename files</i>	bzr [add rm mv] FILENAME [DEST]	git [add rm mv] FILENAME [DEST]	hg [add rm mv] FILENAME [DEST]
<i>commit changes</i>	bzr commit	git commit	hg commit
<i>search the history</i>	bzr log [-r REV_START..REV_STOP]	git log REV_START..REV_STOP	hg log [-r REV_START] [-r REV_STOP]
<i>show differences</i>	bzr diff [-r REV_START..REV_STOP]	git diff REV_START..REV_STOP	hg diff [-r REV_START] [-r REV_STOP]