Unità di apprendimento 1

Documentazione del software

Unità di apprendimento 1
Lezione 6

Il controllo delle versioni dei documenti e del software

In questa lezione impareremo...

- cosa sono i sistemi di versionamento
- VCS distribuiti e centralizzati
- Git: il sistema di controllo di riferimento

Premessa

- Ogni programmatore, soprattutto dopo che ha incidentalmente perso un proprio lavoro almeno una volta, inizia a perfezionare e ad attuare *tecniche proprie di salvataggio dei codici sorgenti*
- Spesso è necessario effettuare il salvataggio non solo a scopo di backup, ma anche per storicizzare la versione distribuita
  - con l'accumularsi delle release può diventare molto complesso mantenerle

Premessa

- Per indicare le release/versioni si potrebbe usare un sistema di numerazione in cui abbiamo
  - major level (release): indica una modifica sostanziale nel prodotto (ad es. nuova veste grafica, passaggio a una nuova piattaforma)
  - minor level (versione): si migliorano le funzionalità esistenti oppure si aggiungono nuove funzionalità
  - patch level (correzione): si rimuovono errori/malfunzionamenti

- Ma non ci risolve il problema di dover mantenere una serie di release e di tenere traccia delle modifiche apportate

I sistemi di versionamento

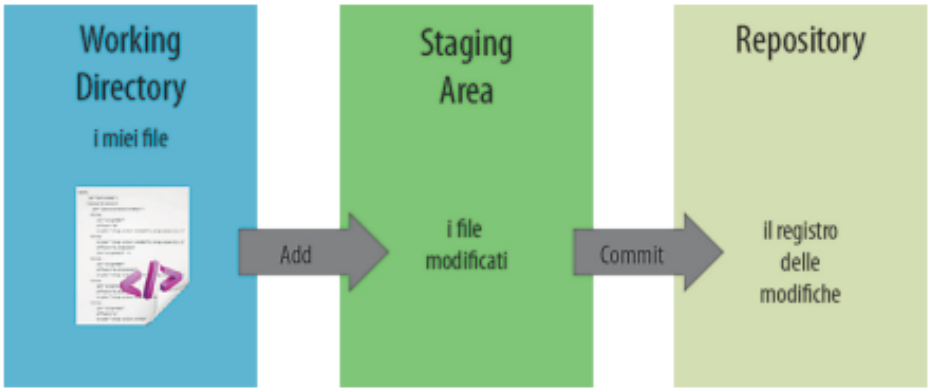
- Un sistema di controllo di versione (VCS, Version Control System) consente invece di gestire le modifiche apportate ai file, anche da più persone
- Scopo di un VCS è di realizzare una corretta gestione delle modifiche garantendo le seguenti caratteristiche:
  - reversibilità
  - concorrenza
  - annotazione

I sistemi di versionamento

- La reversibilità è la capacità di un VCS di poter sempre tornare indietro in un qualsiasi punto della storia del codice
- La concorrenza è la prerogativa che permette a più persone di apportare modifiche allo stesso progetto, facilitando il processo di integrazione di pezzi di codice sviluppati da due o più sviluppatori
- L'annotazione è la funzione che consente di aggiungere note alle modifiche apportate

Terminologia

- Working directory: indica la cartella in cui lavoriamo al nostro progetto, cioè quella che contiene i file nel nostro computer
- Staging area: indica l'insieme degli oggetti (non solo file sorgenti!) che lo sviluppatore ha "lavorato" e che considera in qualche modo "finiti" e meritevoli di essere salvati nel VCS
- Repository: è l'archivio VCS dove gli oggetti sono memorizzati; è analogo a una directory e può essere sia locale sia sul server



Terminologia

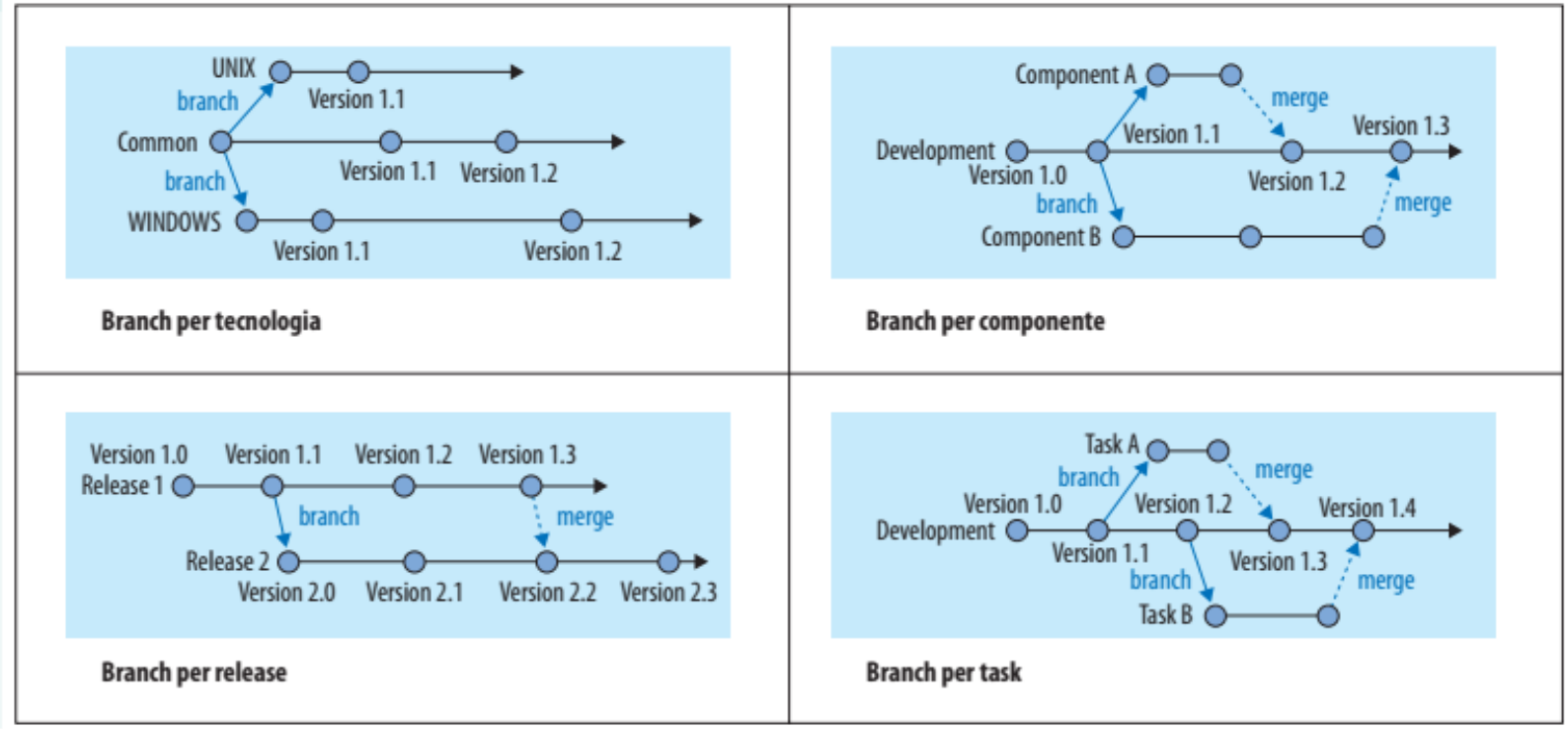
- Un repository è come un database che
  - archivia le modifiche apportate al codice nel tempo
  - tenendo traccia della cronologia del progetto
  - e di chi le ha apportate nel caso di condivisione con altri utenti
- All'atto pratico si tratta di una cartella all'interno di un progetto

Terminologia

- Add: è l'operazione che registra quali file della working directory dovranno essere versionati
  - i file devono essere stati modificati
  - vengono posti nella staging area, un file che mantiene tali informazioni
- Commit: è l'operazione che effettua il salvataggio di una versione, quest'ultima conterrà la lista dei file che erano stati posti nella staging area e delle relative modifiche apportate
  - può essere visto come una fotografia del progetto che ha immortalato tutte le modifiche che erano presenti nell'area di staging
  - per poterla recuperare in caso di bisogno è necessario darle un nome

Terminologia

- branch: è l'operazione che crea una diramazione (ramo) del percorso di sviluppo, è fondamentale nel caso di sviluppi paralleli di un progetto

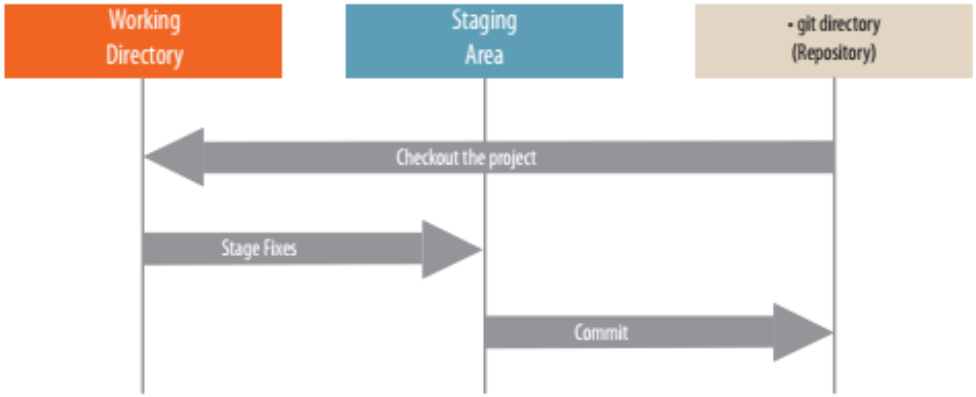


Terminologia

- fork: è l'operazione che serve per creare una copia completa e indipendente di un intero repository da un'altra parte (ad es. su un altro file system o server)
- merge: è l'operazione che permette di riunire versioni differenti in una nuova versione che tiene conto di tutte le modifiche
  - le versioni differenti possono essere dovute a modifiche alternative fatte su versioni differenti o su rami differenti (branch)

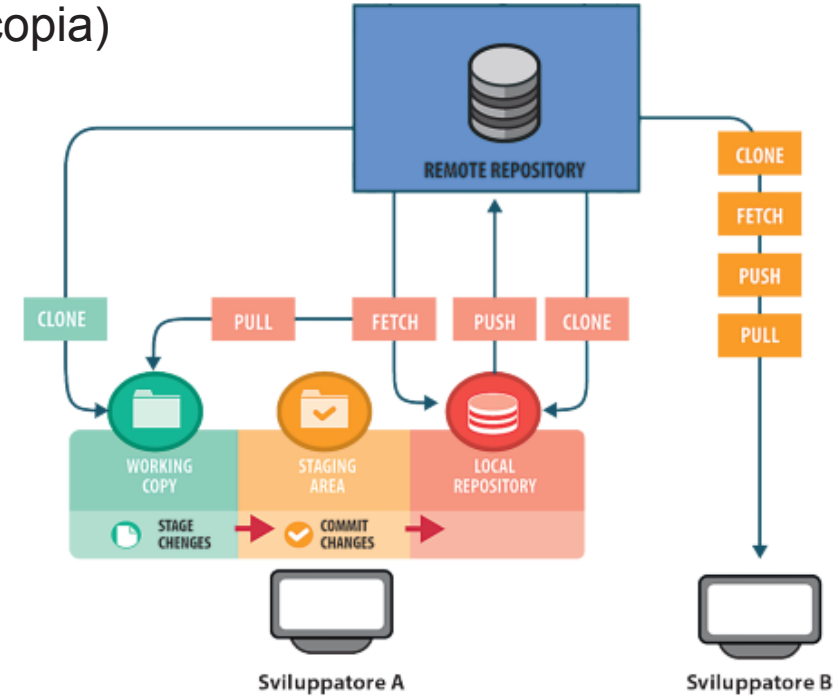
Terminologia

- checkout: è l'operazione che permette di recuperare, tornare alle modifiche inserite in un precedente commit e quindi di riportare il progetto alla versione referenziata da quel commit (sullo stesso ramo in cui ci si trova o su un altro ramo)
  - copia le modifiche fatte in quel commit e salvate sul repository nella propria directory



Terminologia

- push/fetch: servono rispettivamente per
  1) pubblicare i commit dal repository locale ad uno remoto
  2) recuperare gli ultimi commit presenti in un repository remoto in quello locale (sostanzialmente è una copia)



VCS centralizzati e distribuiti

- Esistono due tipologie di sistemi VCS:
  - i sistemi di versionamento centralizzati CVCS, progettati per avere una singola copia completa del repository, ospitata in uno o più server, dove gli sviluppatori salvano le modifiche che hanno apportato
  - i sistemi di versionamento distribuiti DVCS, dove ogni sviluppatore ha una propria copia locale di tutto il repository e dunque può salvare le modifiche ogni volta che vuole

Che cos'è Git

- Git è un sistema controllo di versione distribuito, ovvero ogni sviluppatore ha una copia locale del repository sul proprio computer e può apportare modifiche al codice sorgente in modo indipendente
- Nel momento che un membro del team raggiunge una situazione "significativa" può inviare la sua modifica al repository centrale condividendole con altri membri del team per integrarle nel loro sviluppo del progetto
- Permette di effettuare le operazioni di gestione delle versioni tramite istruzioni a riga di comando
- Permette di salvare lo stato del lavoro e di caricare/ripristinare ognuno degli stati precedenti

Che cos'è Git

- Dunque Git si crea un piccolo file system, che abbiamo detto chiamarsi repository
- A ogni salvataggio (commit) viene generata una cartella con i soli file che sono stati modificati rispetto al salvataggio precedente (con notevole risparmio di spazio)
- Tiene traccia delle modifiche al codice sorgente utilizzando una struttura a grafo basato su istantanee
- Registra l'intero contenuto di ogni file in ogni commit sul computer locale dello sviluppatore
- Ogni commit contiene informazioni come l'autore, la data, l'ora della modifica e una descrizione

Che cos'è Git

- Git consente di creare branch separati per lavorare su funzionalità o correggere bug specifici in modo indipendente e di unire i branch quando sono pronti per essere integrati nel repository principale
- Git offre sia la possibilità di lavorare in locale, sia quella di lavorare in cloud tramite GitHub in modo che da ogni personal computer dell'organizzazione l'utente possa
  - accedere ai propri file
  - condividere file e progetti con altri utenti nella realizzazione di progetti condivisi

GitHub

- Sintetizziamo le operazioni che ogni sviluppatore di un team di progetto effettua:
  1. scarica una copia del progetto dal server GitHub
  2. applica liberamente tutte le modifiche al progetto mentre gli altri programmatori, in parallelo, effettuano le loro modifiche nella loro copia locale
  3. al termine del suo lavoro ogni programmatore aggiorna il progetto sul server VCS
  4. nel frattempo altri programmatori potrebbero richiedere aggiornamenti della loro working copy al repository o generare delle ulteriori versioni

GitHub

- Nel caso in cui due programmatori modifichino lo stesso file
  - se le modifiche si riferiscono a linee di codice diverse, si possono fondere le due versioni
  - se le modifiche riguardano le stesse linee di codice si verifica un conflitto la cui soluzione manuale viene demandata al programmatore
- Questo schema di lavoro prende il nome di modello copy/modify/merge
  Ogni sviluppatore avrà tre copie dei propri file che sono rispettivamente: nella working copy, sul repository locale e sul repository remoto

VERIFICA... le conoscenze

SCELTA MULTIPLA

1. **Quale tra i seguenti livelli non è presente nei sistema di numerazione mediante tre contatori?**
   a. Major level.
   b. Medium level.
   c. Minor level.
   d. Patch level.

2. **Quali caratteristiche deve garantire un sistema VCS?**
   a. Reversibilità, concorrenza, annotazione.
   b. Reversibilità, integrazione, annotazione.
   c. Versatilità, concorrenza, annotazione.
   d. Versatilità, integrazione, annotazione.

3. **Con staging area si intende:**
   a. la cartella locale in cui lavoriamo al nostro progetto.
   b. l'insieme degli oggetti "finiti" e meritevoli di essere aggiunti al repository.
   c. l'insieme degli oggetti "finiti" appena aggiunti al repository.
   d. nessuna delle precedenti.

4. **Git è un sistema controllo di versione** (*due risposte esatte*)**:**
   a. distribuito.
   b. centralizzato.
   c. con struttura a grafo.
   d. con struttura ad albero.

5. **GitHub è un sistema controllo di versione** (*tre risposte esatte*)**:**
   a. distribuito.
   b. centralizzato.
   c. con gestione delle collisioni.
   d. che utilizza il modello di lavoro copy/modify/merge.
   e. che utilizza il modello di lavoro lock/modify/unlock.

6. **La sicurezza del modello copy/modify/merge è anche data dal fatto che ogni sviluppatore avrà più copie dei propri file:**
   a. due copie.
   b. tre copie.
   c. quattro copie.
   d. una copia per ciascun commit.

VERO/FALSO

1. Il major level indica la numerazione di release.
2. Il patch level indica la numerazione di versione.
3. Con working directory indichiamo la cartella in cui lavoriamo al nostro progetto.
4. Il repository è l'archivio presente sul server dove i file sono memorizzati.
5. Commit è l'operazione che permette di effettuare il salvataggio di una versione.
6. Ogni commit necessita di un branch, cioè di un'etichetta che gli viene assegnata.
7. I sistemi di versionamento centralizzati sono anche indicati con CVCS.
8. Git è un sistema di versionamento centralizzato.
9. Git consente di creare branch separati per lavorare su funzionalità o correzioni di bug specifiche in modo indipendente.
10. Git offre sia la possibilità di lavorare in locale siae quella di lavorare in cloud.
11. GitHub consente di creare branch separati per lavorare su funzionalità o correzioni di bug specifiche in modo indipendente.
12. GitHub offre sia la possibilità di lavorare in locale sia quella di lavorare in cloud.