HomeWork Day 2

Siete il release manager di un progetto, descrivete il git flow che tutto il team dovrà seguire.

Spiegate le naming convention che volete adottare, le modalità di commit e integrazione dei vari branch.

In caso di hotfix, descrivete la modalità con cui volete procedere per fixare eventuali bug nel miglior modo possibile e come integrare tali fix.

Descrivete inoltre tutte le operazioni che non devono essere eseguite per preservare l’integrità del progetto.

Come esercitazione:

Staccare un branch dal develop del vostro repository (CognomeNome)

Nelle cartella HomeWork pushare il vostro homeWork2.

Spiegare cosa è una Pull Request e fare la richiesta da GitHub indicando “uiip32” come reviewer.

**Svolgimento**

Il **GIT flow** è un flusso di sviluppo, concepito per sfruttare al meglio le potenzialità del software di versionamento GIT. E’ finalizzato a conservare una storia implementativa pulita e leggibile, dove un nuovo rilascio comunica a tutti gli utilizzatori che vi è una nuova versione del prodotto. Esso faciliterà chi abbia intenzione di utilizzare il nostro software e consentirà di diversificare e concentrare gli sforzi del team che dovrà sviluppare il software.

Il GIT flow prevede la presenza di due elementi principali, il branch master e il branch develop.

Il **branch master**, usato per i rilasci ufficiali, è l’immagine di tutto ciò che è rilasciato in produzione.

Il **branch develop**, utilizzato per lo sviluppo del software, è utile come ramo per le integrazioni per le features.

Oltre al branch master e a quello di develop, abbiamo i branch di funzionalità, o meglio noti come **“feature branch”**, che aggiungono appunto nuove funzionalità al progetto di base. Nel modello GIT flow, questi branch devono sempre essere derivati dal branch di develop, e mai dal branch master.

Ogni features, dovrebbe avere un suo branch, chè può essere pusciato nel repository centrale per esigenze di backup o di collaborazione. Quando lo sviluppo della features è completa, viene mergiata nel branch di develop. Quando vi è un numero abbastanza alto di features o si è in prossimità di una data di scadenza, si stacca un ramo di release dal branch di develop. In questo modo inizia la fase di rilascio della nuova release.

Quando questa release è pronta per essere deploiata, il branch di release viene immesso nel master e taggata con un numero di versione. E’ buona norma anche mergiarlo dentro il develop, che può trovarsi in uno stato più avanzato rispetto a quando il ramo di release era stato staccato.

Al completamento di una merge sul master, essa verrà contrassegnata con un opportuno tag, che ne indichi il versionament.

L’utilizzo del GIT flow non assicura la nascita di problemi molto gravi, che devono essere risolti con urgenza. Straordinariamente, solo in questo caso, è possibile derivare direttamente dal branch master un cosiddetto branch di manutenzione, o meglio noto come branch di Hotfix, che cercherà di risolvere l’errore che si è verificato. Bisogna aggiungere, che se alcune funzionalità sono direttamente connesse alla feature che si vuole correggere, per precauzione bisognerà sospendere il lavoro su queste funzionalità.

Per distinguere questi branch, essi vengono nominati con un nome significativo e consistente, di solito vi è la presenza del prefisso hotfix. Naturalmente questo branch dovrà essere riportato sul master ma anche sul branch di develop, per mantenere una situazione allineata tra i due branch. Anche in questo caso verrà aggiornato un tag sul master.

Una pull request è una procedura usata per informare le persone facenti parte del team di sviluppo, di aver completato una nuova funzionalità. In questa procedura, i membri del team possono discutere delle modifiche apportate e nel coso modificarle.