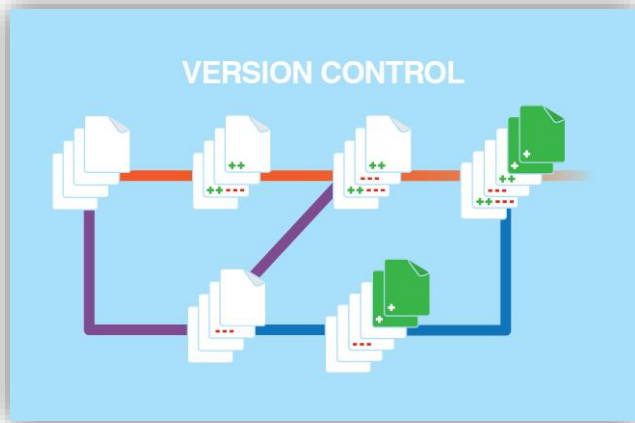


Pràctica 4.1: Teoria dels sistemes de control de versions (SCV)

Tot seguit es plantegen un sèrie de preguntes per comprendre que són els sistemes de control de versions. Amb l'ajuda dels enllaços de la pàgina següent respon les preguntes següents:

1) Què és un sistema de control de versions (SCV)



És un sistema que ens permet organitzar les versions del nostre programa.

D'aquesta manera disposem de seguiment del nostre software.

2) Anomena 4 raons per utilitzar un SCV en els vostres projectes

- Escalabilitat de les aplicacions: a mida que creix el nostre software podem afegir-l'hi noves funcionalitats.
- Reusabilitat: podem partir d'una versió antiga i millorar-la.
- Integritat: som capaços de protegir les nostres versions.
- Productivitat: ens permet compartir o fer còpies del nostre projecte amb molta facilitat.

3) Descriu les diferents etapes del cicle de vida del software

El cicle de vida del software consta del següents procediments:

Definició d'objectius: defineix la finalitat del projecte.

Anàlisis del requisits i la seva viabilitat: recopilar, examinar, formula requisits del client i examinar qualsevol restricció que es pugui explicar.

Disseny general: requisits generals de l'arquitectura de l'aplicació.

Disseny a detall: definició precisa de cada subconjunt de l'aplicació.

Programació: utilitzar un llenguatge de programació per la creació de funcions que s'han definit a l'etapa de disseny.

Proves: es fan proves individuals de cada subconjunt de l'aplicació per garantir que s'han implementat correctament.

Integració: garantir que els mòduls s'integren amb l'aplicació. Aquest procés està acuradament documentat.

Proba beta: es garanteix que el software compleix amb les especificacions originals.

Documentació: documentem informació necessària pels usuaris del software i desenvolupaments futurs.

Implementació: moment on l'aplicació surt.

Manteniment: Suport correcte i actualitzacions que afavoreixen l'aplicació (manteniment continu).

4) Un SCV es caracteritza per un seguit de funcions comunes en la majoria de les solucions existents. Descriu les funcions següents d'un SCV:

Repositori: es un sistema d'informació on s'emmagatzemen dades o fitxers i històrics de versions.

Línia base (baseline): és el conjunt de versions realitzades.

Obrir una branca (branch): es crear una branca a partir d'una versió anterior.

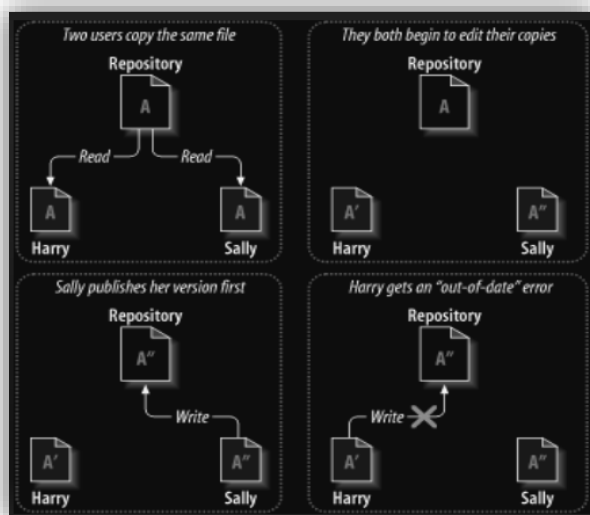
Checkout: permet obtenir una versió local del repositori.

Checkin o commit: serveix per crear confirmacions a l'hora de pujar la teva versió.

Bloquejar (locking): serveix per bloquejar un fitxer, evitant la modificació per altres usuaris.

Congelar: bloqueja la possibilitat de nous canvis a una nova versió.

5) Què entenem per conflicte en un sistema de control de versions? Podries descriure un conflicte.



Un exemple molt clar, seria quan dos usuaris treballem sobre la mateixa versió.

Què passa aquí? En el moment que un usuari puja la nova versió l'altre passaria a treballar sobre un fitxer des actualitzat, aleshores provocaria conflictes quan el segon vulgui pujar el seu.

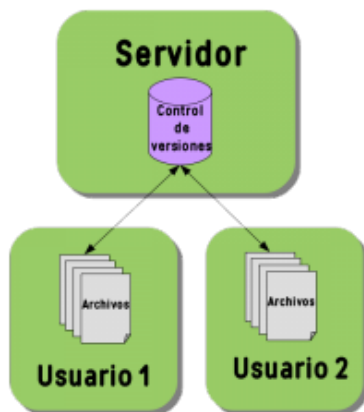
Ens aquests cassos es el propi usuari qui ha de solucionar-ho.

6) Què fa la funció *merge* en un sistema de control de versions?

La funció **merge** permet unificar branques del nostre control de versions, habitualment ajuntarem una branca amb la branca principal.

7) Descriu els SCV centralitzats i descentralitzats i identifica'n les principals avantatges i inconvenients. Quin ús donaries a cada tipus de SCV?

Modelo centralizado



Centralitzat

En aquests existeix un sol repositori on es mantenen totes les versions. Tot canvi s'aplicarà sobre el repositori central i els usuaris treballen amb el mateix repositori.

Avantatges:

- Més control sobre les versions
- Més fàcil d'organitzar respecte als distribuïts.
- Sempre tens l'última versió.

Inconvenients:

- Poca flexibilitat a l'hora de fer branques.
- Depenem de connexió per treballar amb el servidor.
- Si es perd el repositori central, es perd tot.

Aquest tipus de model funciona bé si treballem dins d'un equip petit, on tots els membres tenen la capacitat de comunicar-se ràpidament de manera que no hi hagi dos programadors treballant sobre el mateix codi simultàniament.

Distribuït

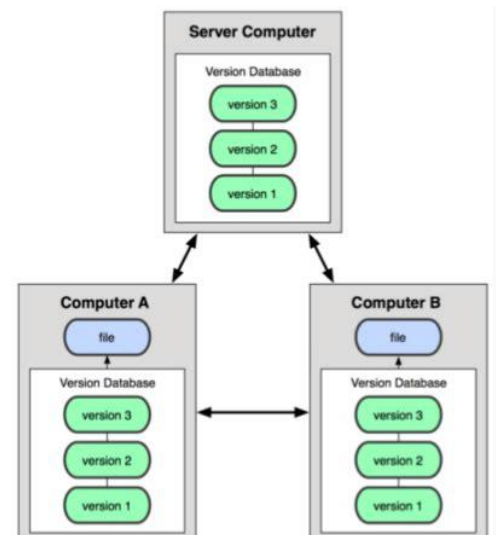
Utilitzant una xarxa P2P, no tenim un únic repositori, sinó que cada usuari té el seu propi. Si tenim una versió finalitzada la podem enviar a la resta d'usuaris.

Avantatges:

- Cada usuari pot crear les versions.
- Cada usuari té les seves pròpies còpies.
- No depenem d'un servidor.
- Millor rendiment
- Fusionar branques és més fàcil

Inconvenients:

- S'han d'aprendre més comandaments.
- Menys control sobre les versions, ja que cada usuari pot tenir les seves.



Aquest tipus funciona bé quan treballem amb projectes grans, en aquests cassos el nostre equip no consta d'uns quants programadors sinó que en tenim a tot arreu. No podríem amb un sol servidor i tenir-ne més implicaria encarir costos, llavors es millor utilitzar un sistema distribuïts. Que siguin el mateixos usuaris el qui tinguin les versions.

8) Indica les solucions existents actualment de SCV i digues quins són centralitzats i quins són distribuïts

Sistemes centralitzats:

- Current Version System (CVS)
- Subversion, basat en CVS
- Tortoise, és una plataforma de Subversion per Windows
- Perforce

Sistemes distribuïts:

- Git
- Mercurial
- Dropbox
- Bazaar
- Darcs

Enllaços interessants sobre sistemes de control de versions:

<http://betterexplained.com/articles/a-visual-guide-to-version-control/>

<http://blog.juliopari.com/buenas-practicas-de-gestion-de-versiones-con-subversion/>

<http://svnbook.red-bean.com/index.es.html>

<http://svnbook.red-bean.com/en/1.6/svn.basic.version-control-basics.html>

https://en.wikipedia.org/wiki/Software_release_life_cycle

<http://www.richappsconsulting.com/blog/blog-detail/svn-vs-git-who-will-be-the-future-of-revision-control/>

<http://www.slideshare.net/thegeekinside/control-de-versiones-distribuido>

<http://codeando.blogspot.com/2007/10/control-de-versiones-distribuido.html>

<https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>

http://talks.castagnetto.org/s5/20080201_IntroSCV/index.html

