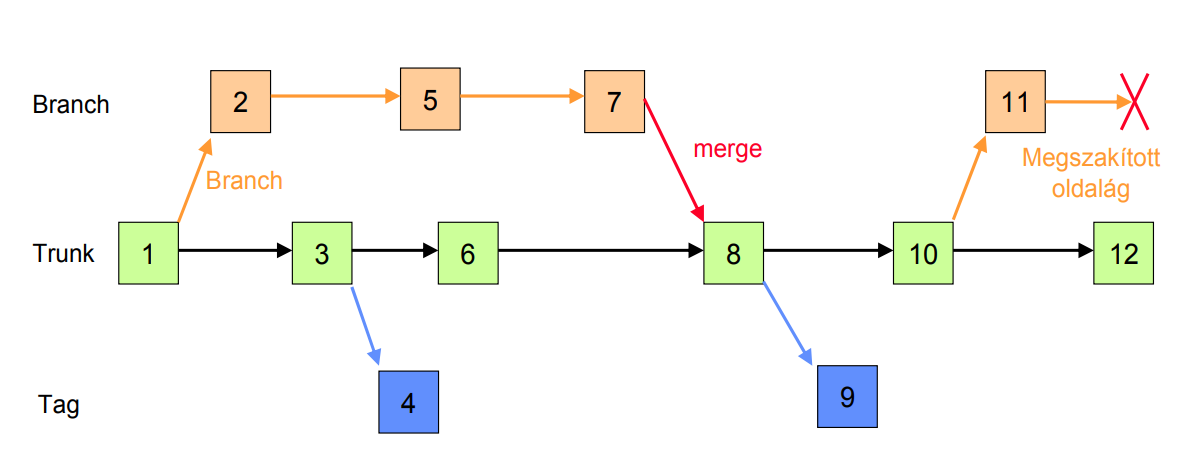
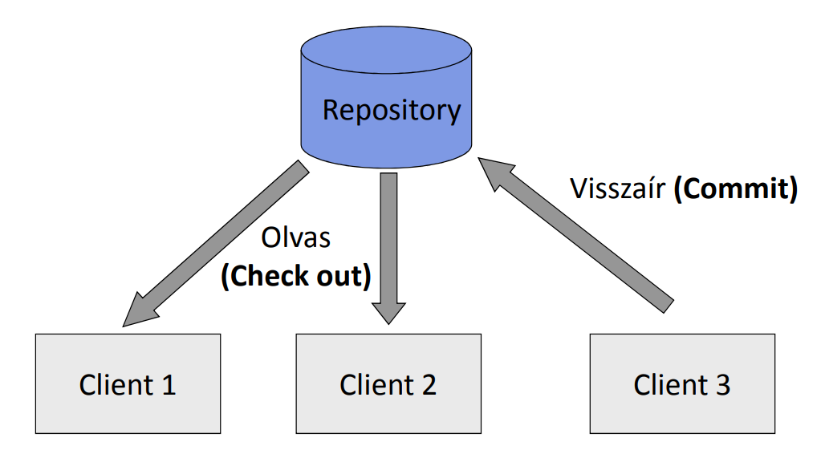
**Konfiguráció-menedzsment**

* Célja a termékek integritásának megőrzése a fejlesztési folyamat során, segítségével reprodukálható és követhető a rendszer változása
* A fejlesztett termékek változtatásait, verzióit egyértelműen kell tudni azonosítani
* A folyamat, tesztesetek változhatnak, ezeket verzió követni kell
* Fontos az, hogy a változások utólag is nyomon követhetőek legyenek
* A termék és a tesztelési elemek kapcsolatát egyértelműen kell megadni
* A legtöbb esetben a konfiguráció-menedzsmentről a projekt gondoskodik, a tesztelés csak felhasználja azt
* A konfiguráció-menedzsmentet általában valamilyen szoftver segítségével végezzük, a három legelterjedtebb: Ansible, Chef, és a Puppet
* Az egyik megközelítési módja, hogy a konfigurációt és a szoftver függőségeit programkódként írjuk le, ennek a pozitívumai:
  + tesztelhetőség
  + újrahasznosíthatóság
  + futtatható dokumentáció
  + reprodukálhatóság
* A konfiguráció-menedzsment elősegíti a rendszer felépítésének és konfigurációjának mélyebb megértését, megkönnyíti az operációs rendszerek közötti migrációt
* A fejlesztésre használt eszközök, környezetek egyezzenek meg az éles környezettel
* A rendszer fejlődése során a korábbi rendszer verziókra lefuttatott teszteket a továbbfejlesztett verziókra is le kell futtatni, bizonyítva azt, hogy a továbbfejlesztés nem okozott hibát a korábban már implementált funkciók végrehajtásában.
* Az inkrementális fejlesztési stratégiát követve tehát a rendszer, és annak helyességét ellenőrző tesztesetek számos verziója keletkezik. Az egyes verziókhoz tartozó forráskódok, dokumentációk és tesztesetek nyilvántartására és kezelésére célszerű valamilyen verziókezelő rendszert használni
* Példa egy tipikus konfigurációs életciklusra:



* A verziókövetés valamilyen formában is, de elengedhetetlen egy jó szoftver megalkotásához
* Centralizált verziókövető rendszerek működési alapjai



* Elosztott verziókövető rendszer például a GIT
  + Előnyei:
    - mindenkinek megvan a saját játszótere
    - offline is működik
    - gyors
    - kevés menedzsment igény
    - könnyű branchelni
  + Hátrányai:
    - Backup-ra továbbra is szükség van
    - nincs olyan, hogy jelenlegi verzió
    - nem léteznek revízió számok
* Verziómenedzsmentre több stratégiai is létezik még:
  + Lock-Modify-Unlock megközelítés
  + Copy-Modify-Merge megközelítés