Source control systems

Идеята на Version control system-ите е да се проследяват промените по някакви файлове (например от даден проект). Не е задължително файловете да съдържат сорс код. Може да се проследява историята за промените по файловете

Source control system = Version control system – позволява ни да работим върху версиите на даден софтуер, който разработваме. Ако решим, че сме направили някаква грешка по нашия проект, можем да се върнем към по-стара версия, от преди месец например.

Software development cycle:

* Analysis – решаваме какви технологии ще са ни нужни, какъв програмен език ще използваме, какви структури от данни и т.н.
* Models – правим дизайна на софтуера – например ако правим интернет сайт правим страниците, който са ни нужни; или ако е някакво приложение пишем класовете, които ще има
* Implementation – писането на сорс код
* Build – build-ваме приложението, изчистваме грешките, които се появяват
* Testing – тестване на приложението
* Release – изкарване на final продукта

Version control система:

* File version control – поддържа различни версии на файловете/ сорс кода
* Merge and differences search – merge-ване на различни версии на един и същ файл
* Branching – чрез branching-а може да правим различни версии на един и същ продукт
* Console and GUI clients – има конзолна среда и GUI клиенти за работа с VCS системата

VCS системите пазят change log (history) на промените направени през времето

Repository – сървъра на който се съхранява проекта

Revision, Version – всеки път когато правим някаква промяна по проекта, това е нова версия

Clone (git), Check-out (SVN) – вземаме копие от даден проект намиращ се на отдалечен сървър и сваляме това копие локално на нашата машина

Change – промяна на даден локален файл

Change set/Change list – свъкупност от промени, които ще бъдат commit-нати по едно и също време

Commit/Check-in – submit-ва промените, които сме направили на нашата локална машина, в отдалеченото repository, където се намира проекта.Автоматично се създава нова версия.В този случай може да възникнат конфликти

Conflict – едновременна промяна на файловете в repository-то от няколко души

Update (SVN)/Pull (Git) – сваляме най-новите update-нати данни на нашата локална машина от сървъра

Undo changes/Revert (Git) – връща даден файл, който сме променили, към последната версия, към която е бил update-нат

Merge – когато се получава конфликт, можем да слеем/merge-нем двете промени

Branch – разделяне на проект на различни потоци за да може да се работи паралелно по този проект

Versioning models

* Lock-Modify-Unlock – файлът, по който се работи се lock-ва от user-а и когато работата свърши, файлът се unlock-ва
* Copy-Modify-Merge – user-ите правят паралелни промени на работните копия. Получават се конфликти когато двама или повече user-a искат да модифицират една и същ файл. Тук няма version control
* Distributed version control – всеки user си има свое собствено локално repository и работи по него. Когато стане време локалното repository се push-ва към отдалеченото repository

Branching – allows splitting the development line into separate branches.

* Suitable for development of new features in a new version of the product