**7.[Аналіз функцій і ролі системи контролю версіями](http://masters.donntu.org/2019/fknt/gurin/diss/indexu.htm" \l "p2)**

Система управління версіями - це ПЗ для спрощення роботи з документами, які часто змінюються. Система керування версіями може зберігати різні версій одного і того ж документа, і дозволяє переміщатися за різними версіями проекту і навіть до найпершим. Так само, при отриманні будь-якої версії, можна подивитися всі зміни документа, і хто в який час вніс зміни. Дані системи часто застосовуються при розробці ПЗ для зберігання і управління документів з вихідним кодом. Але це не виключає ситуацію, що їх можна використовувати і в інших сферах, в які робота проходить в часто змінюються документах і потрібно відстежувати всі зміни з ними. Так само варто відзначити, що дуже часто СУВ використовуються в САПР, які використовуються в складі системи управління даними про виріб (PDM).[[2](http://masters.donntu.org/2019/fknt/gurin/diss/indexu.htm" \l "ref2)]. Традиційні системи контролів версій має централізовану модель. Система має певне сховище і управління цієї системи відбувається на сервері. Порядок основних дій з системою контролів версій:

1. користувач, який працює з файлом, отримує потрібну версію з сховища. Отримати можна абсолютно будь-яку версію, як і саму останню, так і найпершу.
2. виробляється створення локальної копії на комп'ютері користувача.
3. потім користувач вносить зміни і зберігає файл, який автоматично буде поміщений в сховище, але важливо розуміти, що старий видалений не буде.

Як було сказано вище, файл можуть змінювати два і більше користувача. У такій ситуації може статися те, що один користувач випадково скасувати зміни іншого. Система контролів версії відстежує подібні конфліктні ситуації і здатна надавати їх вирішення. Якщо зміни двох користувачів не впливають один на одного, то система здатна сама об'єднати зміни обох користувачів в один результуючий файл.

У різних системах управління зустрічаються такі можливості, як:

* створення різних варіантів одного файлу, так звані гілки, які містять повну історію зміни документа до розгалуження і історію кожної гілки окремо;
* перегляд повного опису змін версії одного документа, зокрема хто вносив зміни в певні частини рядків документа і коли
* ведення журналу зміни, який містить в собі призначені для користувача пояснення про дані зміни в кожній версії документа
* контроль прав доступу кожного користувача, наприклад, дозволяючи і забороняючи читання, або зміни даних, в залежності запитуваної доступ.

Існують ще розподілений системи управління версіями. Їх головна відмінність в тому, що вони мають розподілену модель, а не традиційну клієнт-серверну. Вони не потребують наявності певного сховища: вся історія зміни і всі версії зберігаються на машині, в локальному сховищі, і тільки при необхідності потрібні частини документів синхронізуються з подібним сховищем на іншому комп'ютері. Це дає перевагу перед централізованими в тому, що кожен користувач працює над своєю локальною копією, а потім вже додає (виробляє push) на централізоване сховище. Але перед цим етапом потрібно завантажити актуальну копію на своїй комп'ютер (зробити pull). Це робиться з тією метою, щоб, якщо є конфлікти в двох версіях документа (того, який на іншому комп'ютері і того, який на локальному комп'ютері), їх дозволити. У розподілених є ряд недоліків:

* збільшене вимоги до обсягу дискової пам'яті, бо на локальній машині будуть зберігається все версії документів;
* немає можливості блокувати або стежити за файлом або групу файлів;
* немає єдиної наскрізної нумерації системи.

Переваги розподілених систем керування версіями:

* позбавляє від необхідності виділяти один з комп'ютерів під сервер;
* під час розробки невеликого проекту не потрібно виділяти загальні ресурси;
* під час розробки великого розподіленого проекту, в якому користувачі можуть працювати довгий час над своєю локальною версією, при цьому без обов'язкового постійної наявності в інтернет.

Системи контролю версій можуть використовуватися, так звані дельта-компресії - це такий підхід, при якому записується тільки зміни документа, а не всього версії. Простий приклад може привести на цифрах, замість запису 2, 4, 6, 7, 9, 5, 15, ми будемо записувати 2, 2, 2, 1, 2, -4, 10 [[3](http://masters.donntu.org/2019/fknt/gurin/diss/indexu.htm#ref3)]. Так як в більшості випадків потрібно остання версія, то на сервері записуємо, замінюючи попередню актуальну модель.