

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Кафедра: ІТТС

ЛАБОРАТОРНА РОБОТА №1
КЕРУВАННЯ ВЕРСІЯМИ ЗА ДОПОМОГОЮ GIT

Виконав:
ст. гр. СІ-20-1
Боднарук О.А.
Перевірів:
Штаєр Л. О.

Івано-Франківськ
2023

Мета роботи: вивчити та закріпити на практиці можливості системи керування версіями, одержати практичний досвід у використанні Git та GitHub.

Теоретичні відомості: Система керування версіями (англ. source code management, SCM) — програмний інструмент для керування версіями одиниці інформації: вихідного коду програми, скрипту, веб-сторінки, веб-сайту, 3D моделі, текстового документу тощо. Система керування версіями — це потужний інструмент, який дозволяє одночасно, без перешкод один одному, проводити роботу над груповими проектами. Системи керування версіями зазвичай використовуються при розробці програмного забезпечення для відстеження, документування та контролю над поступовими змінами в електронних документах: у коді додатків, кресленнях, електронних моделях та інших документах, над змінами яких одночасно працюють декілька людей. Кожна версія позначається унікальною цифрою чи літерою, зміни документу занотовуються. Зазвичай також зберігається автор зробленої зміни та її час. Інструменти для контролю версій входять до складу багатьох інтегрованих середовищ розробки. Системи керування версіями існують двох основних типів: з централізованим сховищем та розподіленим. Централізовані системи контролю версій. Централізована система контролю версій (клієнт-серверна) — система, дані в якій зберігаються в єдиному «серверному» сховищі. Весь обмін файлами відбувається з використанням центрального сервера. Є 9 можливостей створення та роботи з локальними репозиторіями (робочими копіями). Переваги: — загальна нумерація версій; — дані знаходяться на одному сервері; — можлива реалізація функції блокування файлів; — можливість керування доступом до файлів. Недоліки: — необхідність мережевого з'єднання для оновлення робочої копії чи збереження змін. До таких систем відносять Subversion, Team Foundation Server. Розподілені системи контролю версій. Розподілена система контролю версій (англ. Distributed Version Control System, DVCS) — система, яка використовує замість моделі клієнт-сервер, розподілену модель зберігання файлів. Така система не потребує сервера, адже всі файли знаходяться на кожному з комп'ютерів. Переваги: — кожний з розробників працює зі своїм власним репозиторієм; — рішення щодо злиття віток приймається керівником проекту; — немає потреби в мережевому з'єднанні. Недоліки: — немає можливості контролю доступу до файлів; — відсутня загальна нумерація версії файлу; — значно більша кількість

необхідного дискового простору; – немає можливості блокування файлів. До таких систем відносять Git, Mercurial, SVK, Monotone, Codeville, BitKeeper. Система контролю дозволяє зберігати попередні версії файлів та завантажувати їх за потребою. Вона зберігає повну інформацію про версію кожного з файлів, а також повну структуру проекту на всіх стадіях розробки. Місце зберігання даних файлів називають репозиторієм. В середині кожного з репозиторіїв можуть бути створені паралельні лінії розробки — гілки. Гілки, зазвичай, використовують для зберігання експериментальних, незавершених (alpha, beta) та повністю 10 робочих версій проекту (final). Більшість систем контролю версії дозволяють кожному з об'єктів присвоювати теги, за допомогою яких можна формувати нові гілки та репозиторії. Використання системи контролю версії є необхідним для роботи над великими проектами, над якими одночасно працює велика кількість розробників. Системи контролю версії надають ряд додаткових можливостей: – можливість створення різних варіантів одного документу; – документування всіх змін (коли і ким було змінено/додано, хто який рядок змінив); – реалізує функцію контролю доступу користувачів до файлів. Є можливість його обмеження; – дозволяє створювати документацію проекту з поетапним записом змін в залежності від версії; – дозволяє давати пояснення до змін та документувати їх. Словник основних термінів-сленгів: – транк (trunk) — основна гілка коду; – бранч (branch) — відгалуження; – чекін (Check in (submit, commit)) — відправлення коду в репозиторій; – чекаут (Check out) — одержання зміни з репозиторію; – конфлікти — виникають, коли кілька людей правлять один і той же код, конфлікти можна вирішувати; – патч — шматок з записаними змінами, які можна застосувати до сховища з кодом.