Manual QA

Основи Git

План уроку

- Архітектури систем контролю версій
- Створення репозиторію
- Стан файлів
- Основні команди
- Перший коміт
- Практика

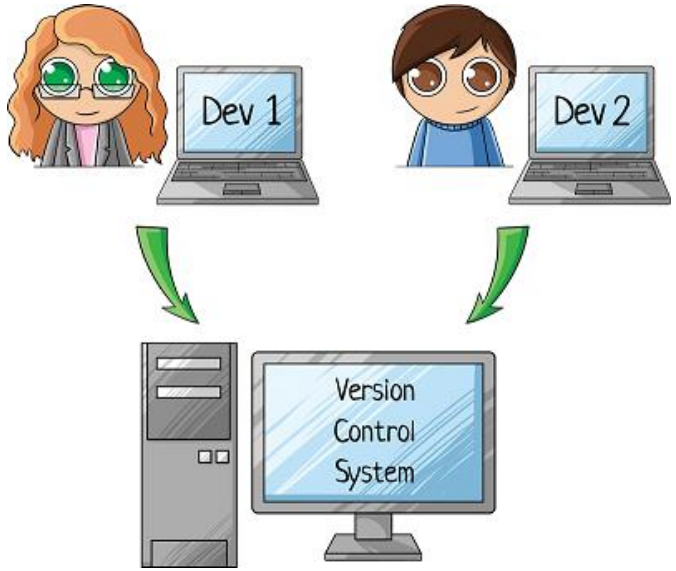
Про системи контролю версій

Manual QA

Схема розробки ПЗ

Системи контролю версій (VCS)

git

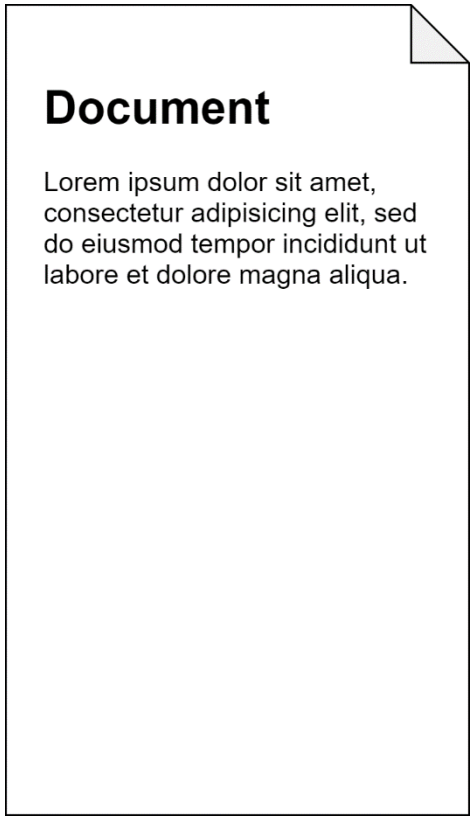


SUBVERSION

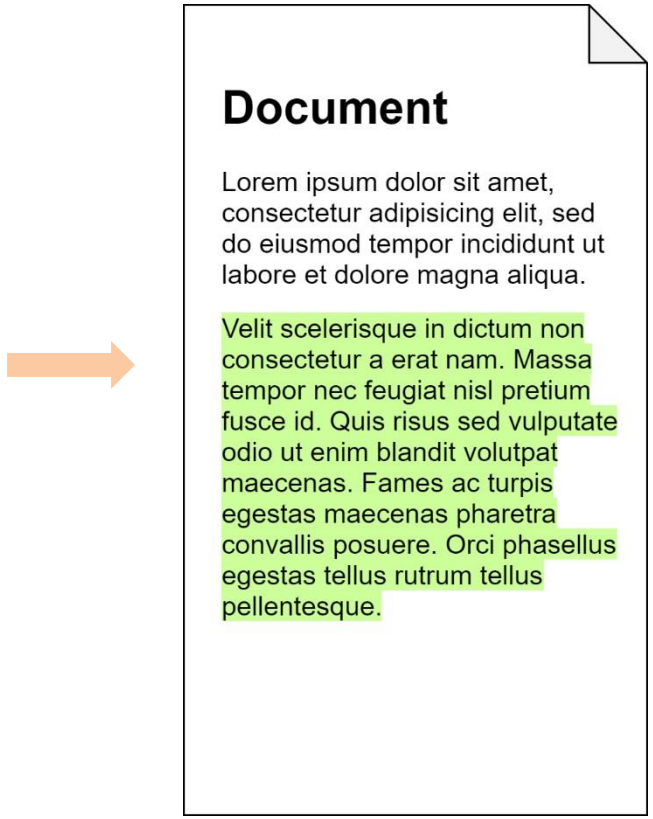
Історія та архітектури систем контролю версій

Версійність

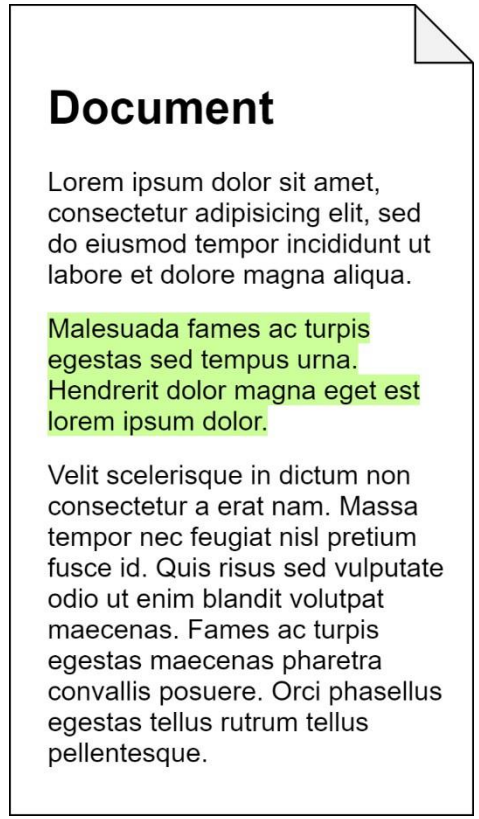
Версия 1



Версия 2

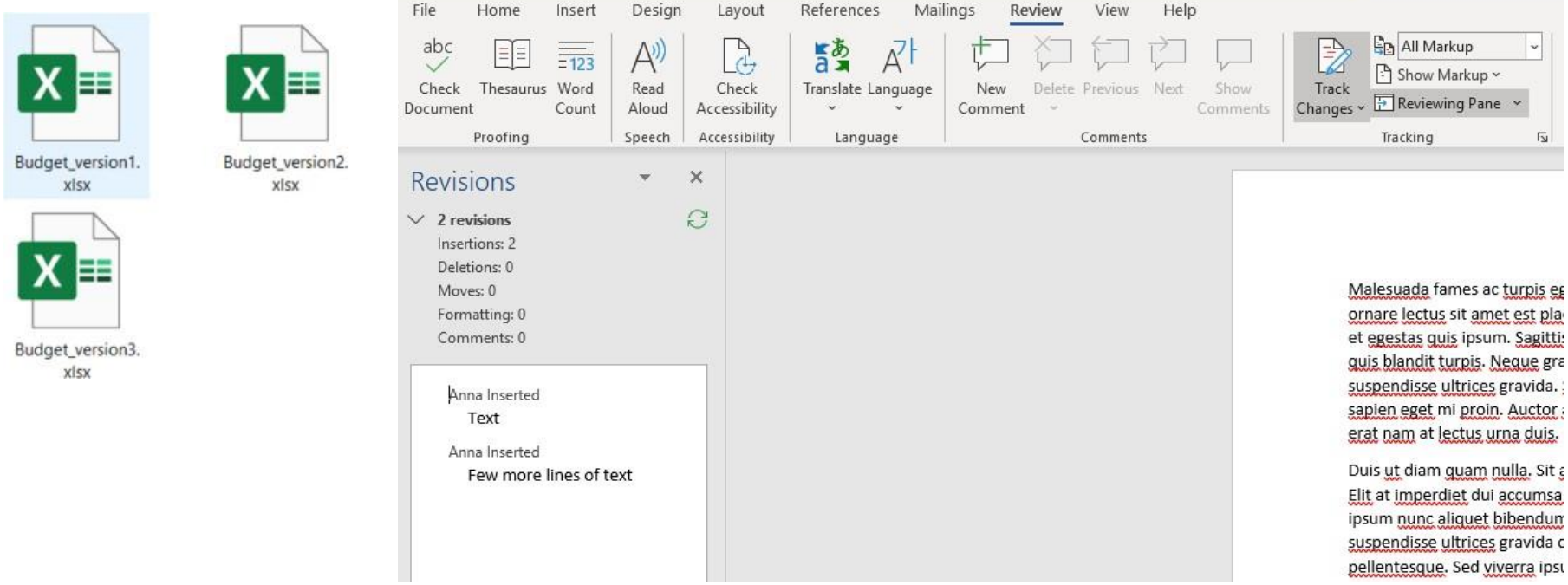


Версия 3

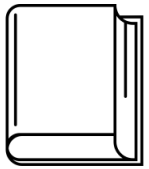


Manual QA

Приклад: Версійність у MS Office



Варіант 1: Локальна система



Локальна копія

- Версія 1
- Версія 2
- …

Які переваги такого підходу?

А які складнощі?

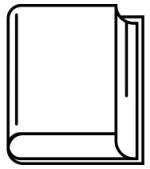
Варіант 1: Локальна система

Переваги

- Простота
- Легкий старт

Недоліки

- Висока ціна помилки
- Важко працювати у кооперації

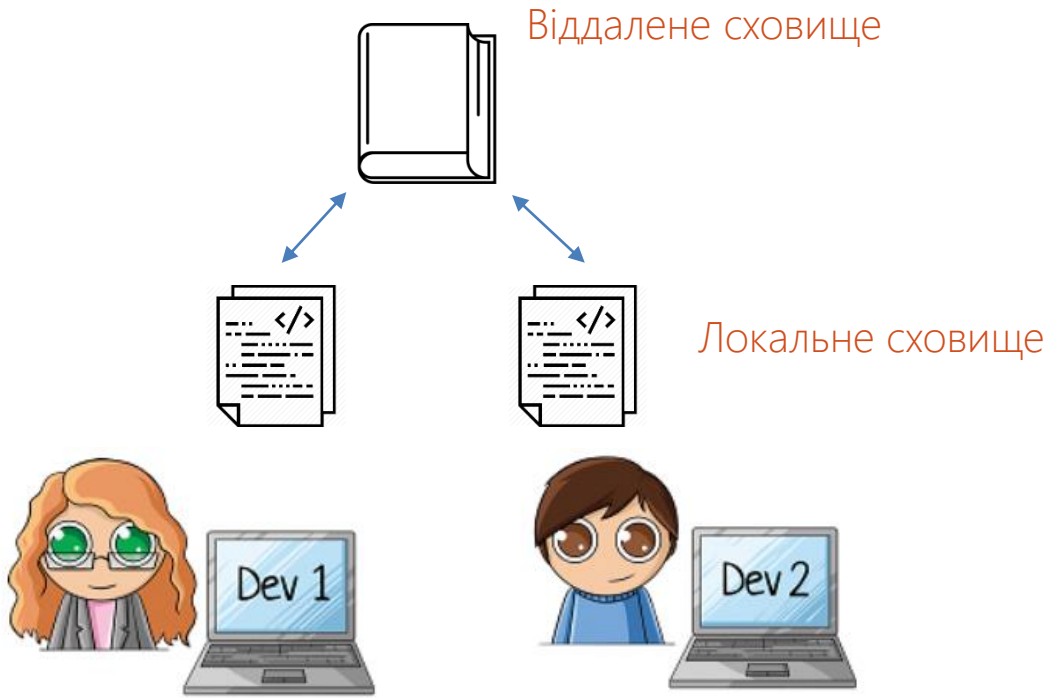


Локальна копія

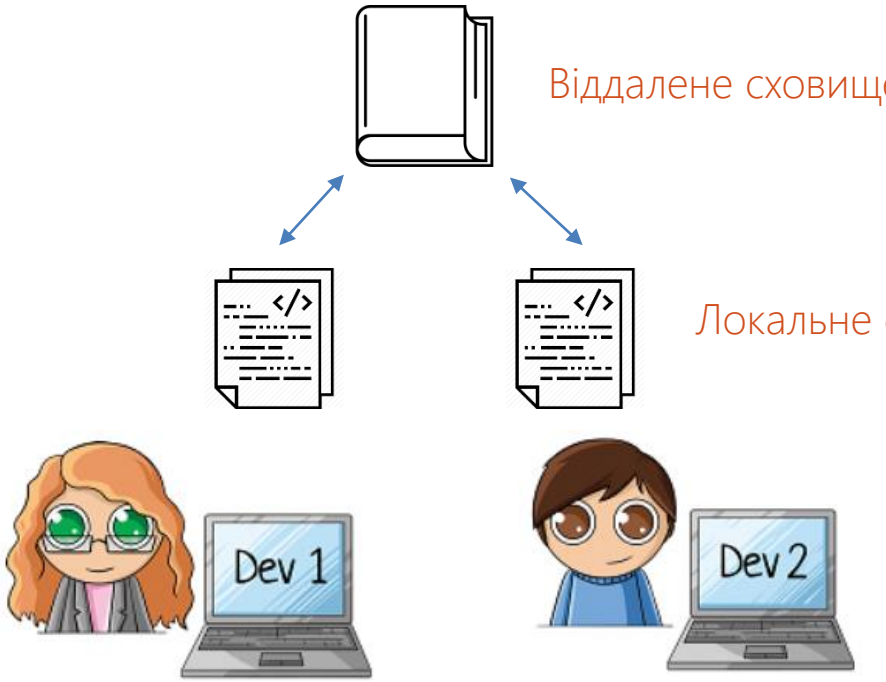
- Версія 1
- Версія 2
- ...

Що можна зробити?

Варіант 2: Віддалений сервер



Варіант 2: Централізована система контролю версій



Віддалене сховище

Містить усі файли та всі версії

Локальне сховище

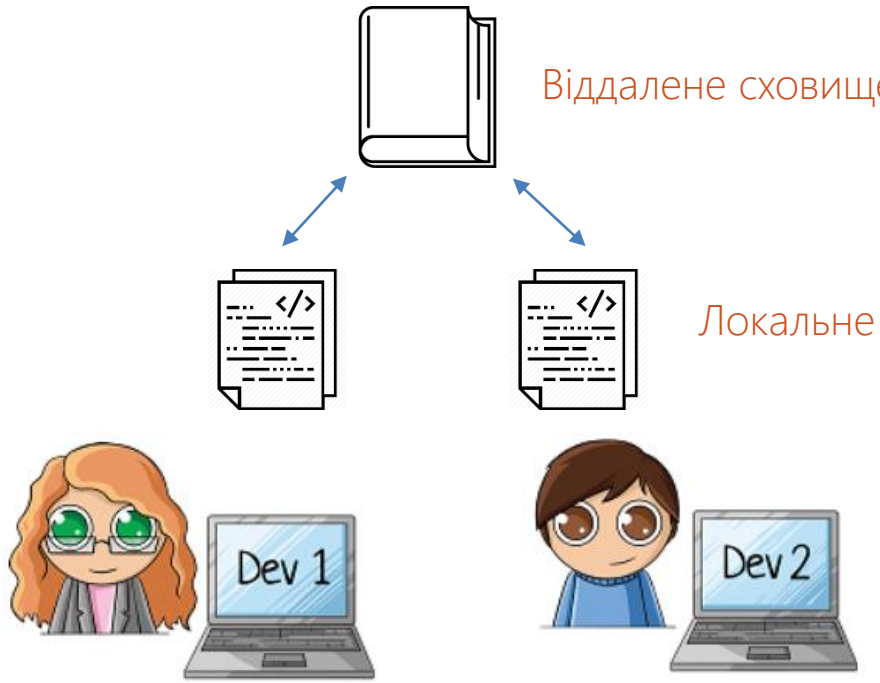
Окремий файл

Які переваги такого підходу?

А які складнощі?

Google Drive

Варіант 2: Централізована система контролю версій



Переваги

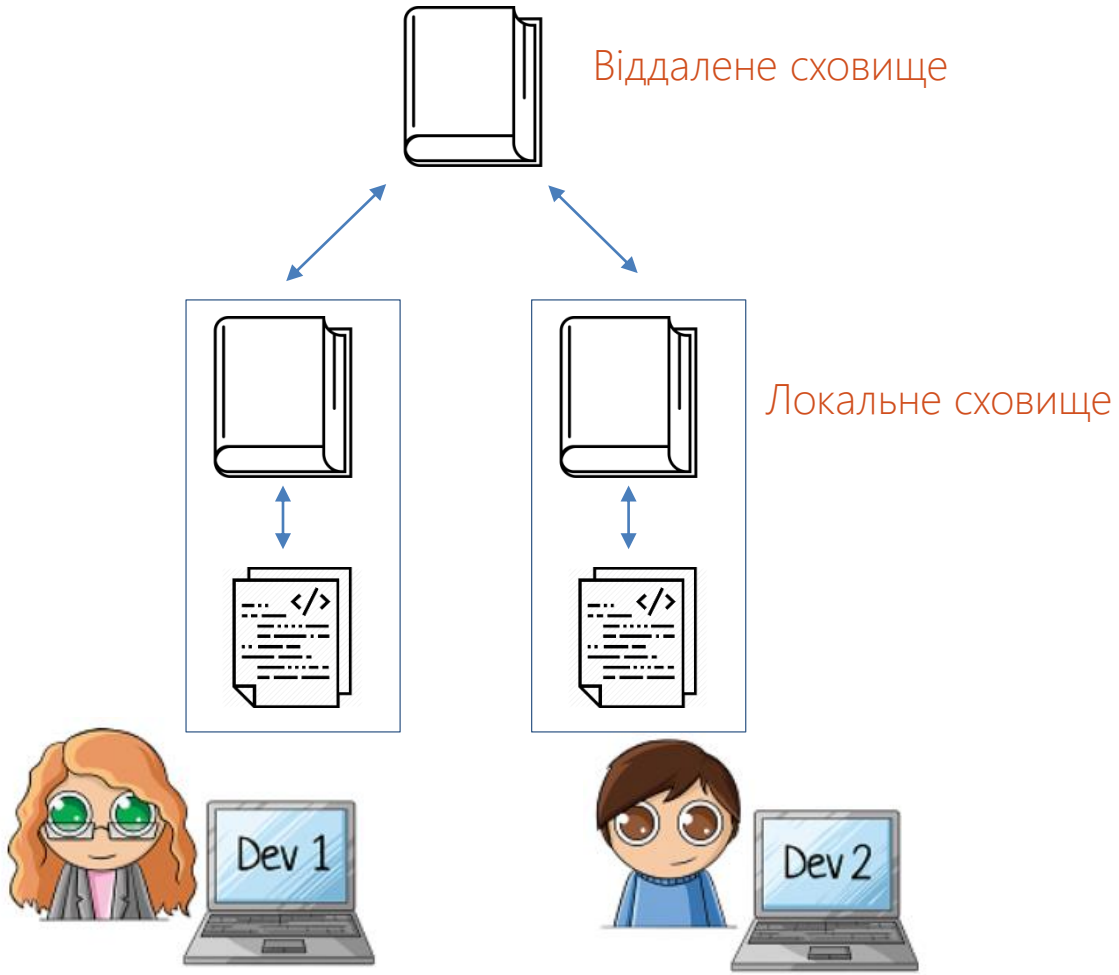
- Проста схема роботи
- Одне джерело правди

Недоліки

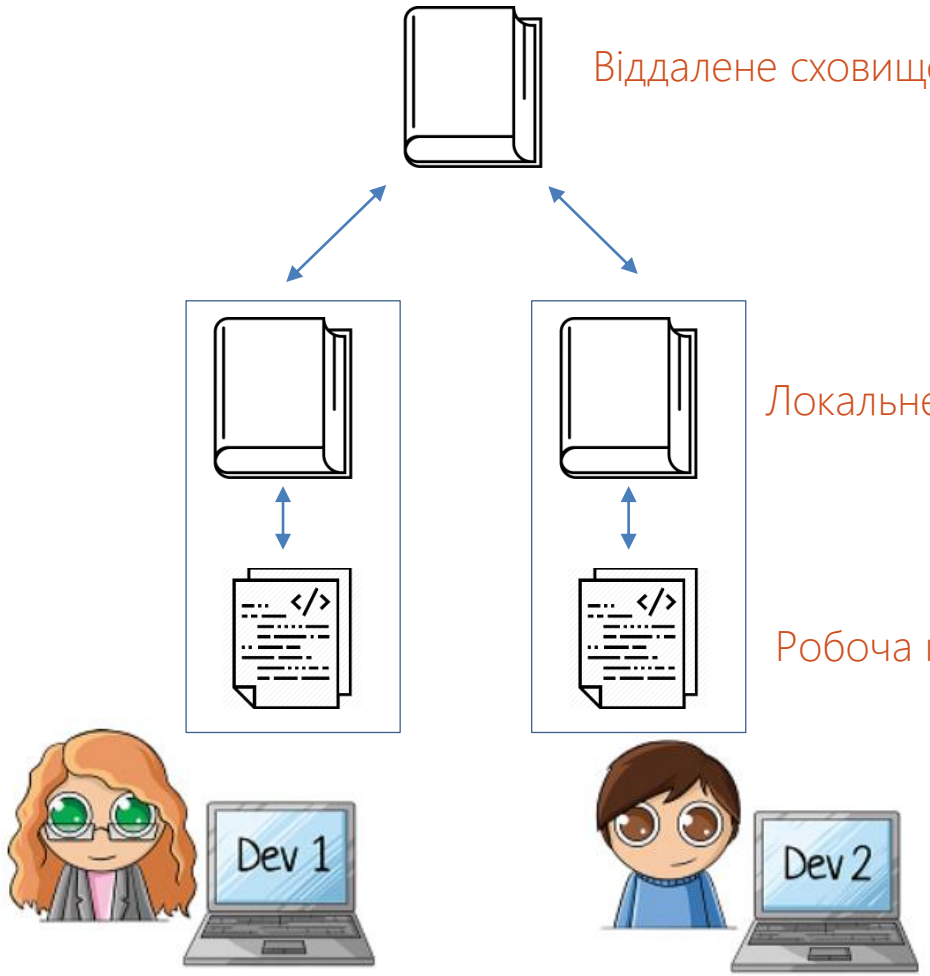
- Одна точка відмови - сервер

Що можна зробити?

Варіант 3: Розподілена система



Варіант 3: Розподілена система контролю версій



Віддалене сховище

Містить усі файли та всі версії

Локальне сховище

Повна копія віддаленого сховища

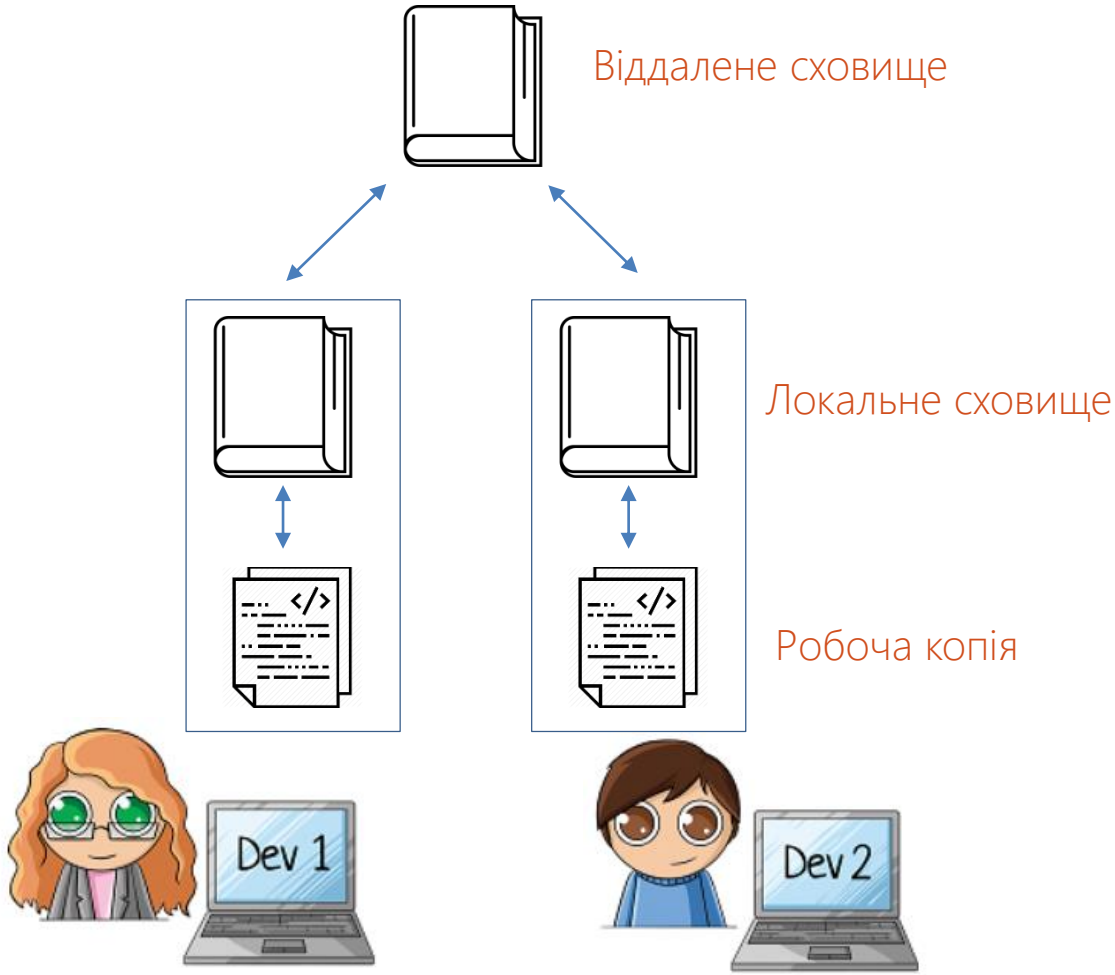
Робоча копія

Окремий файл

Google Drive

+ desktop app

Варіант 3: Розподілена система контролю версій



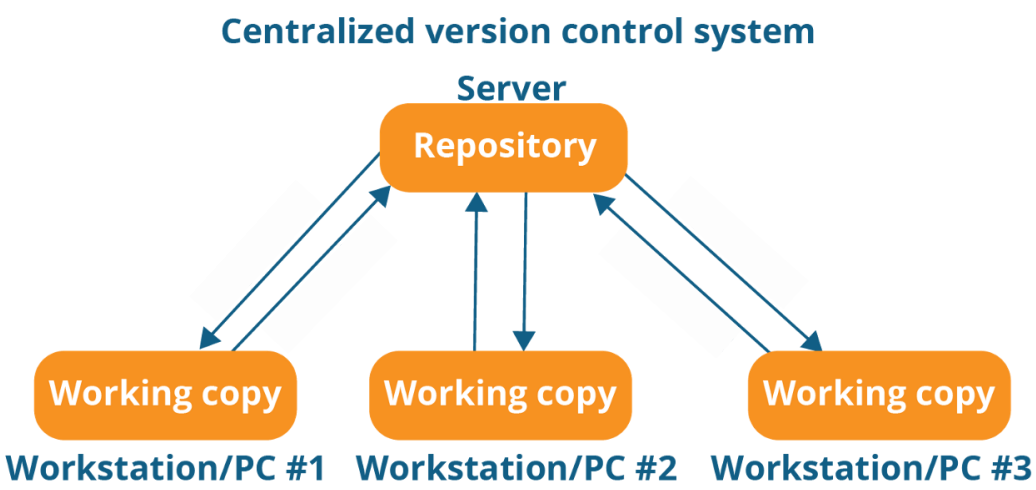
Переваги

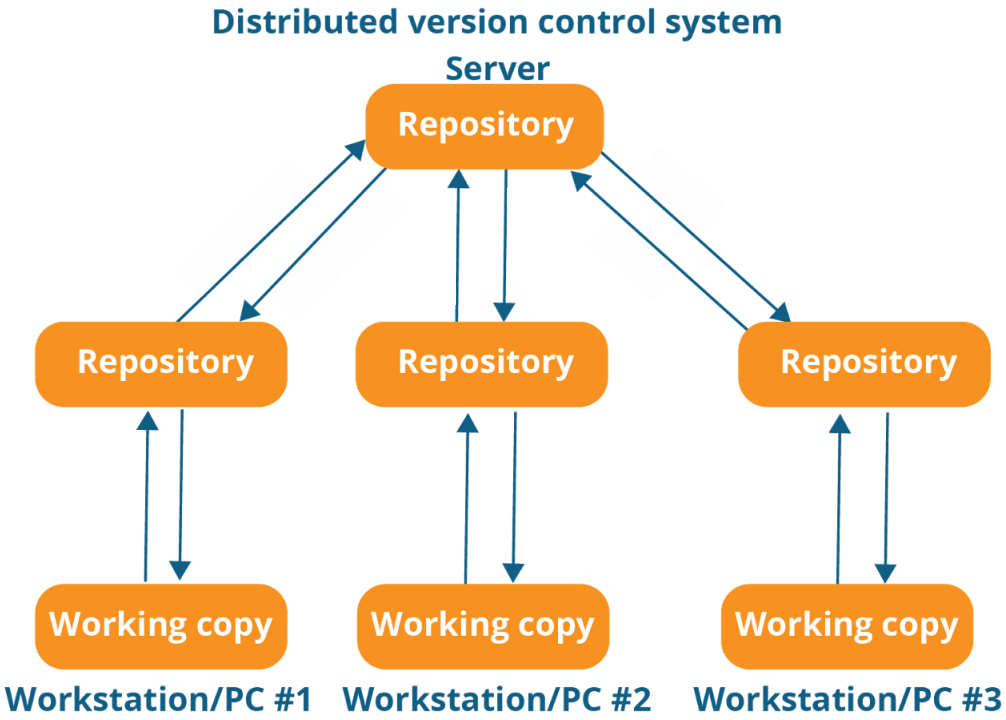
- Повна автономність кожного розробника
- Велика стійкість до втрати даних

Недостатки

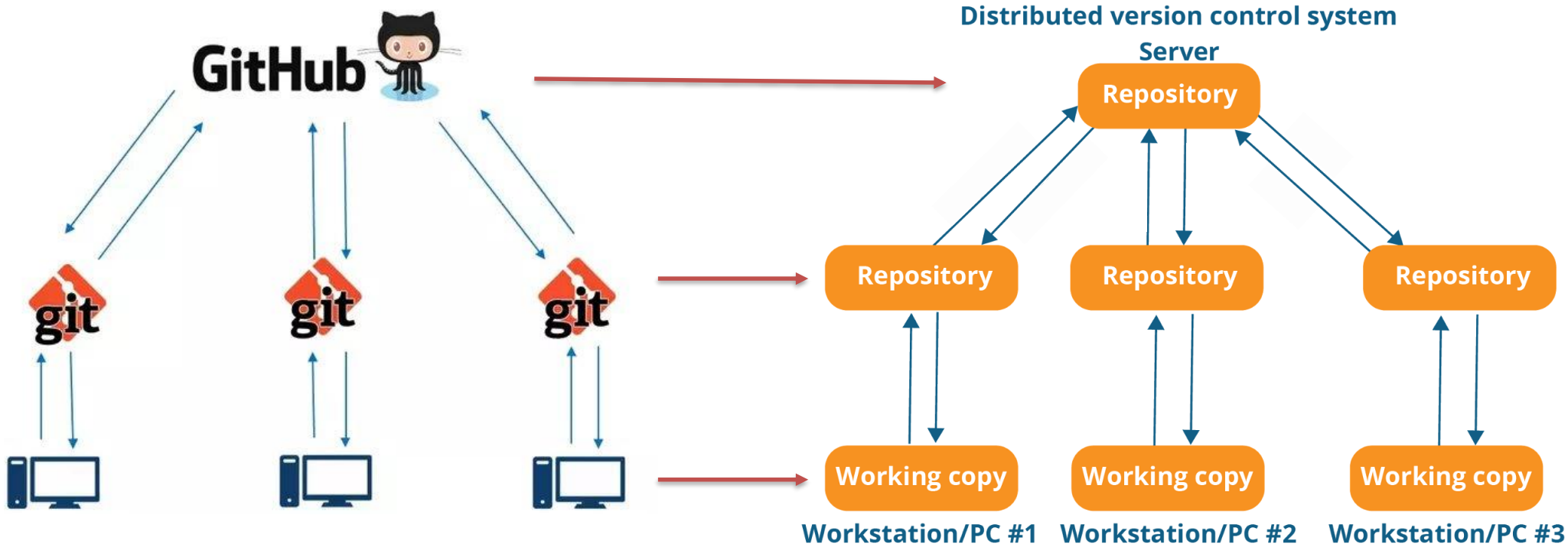
- Найбільш складна схема роботи

Порівняння систем





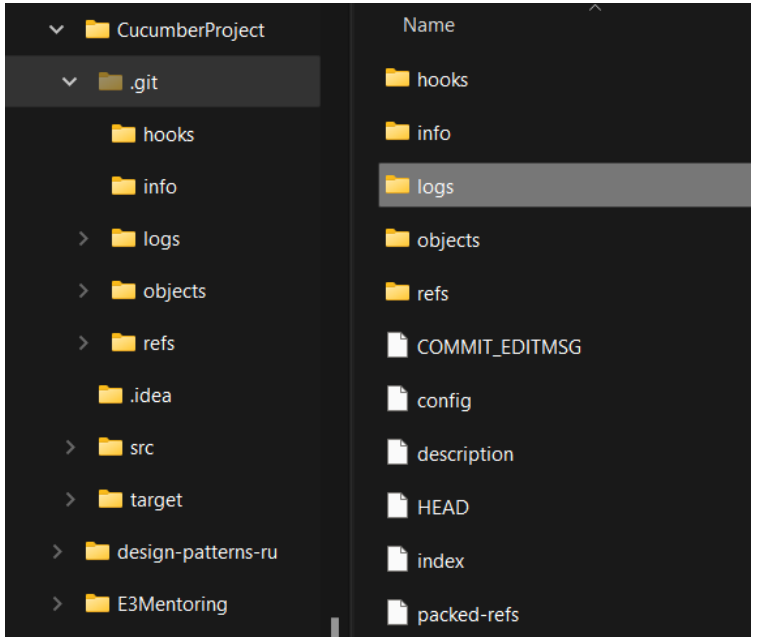
Інструменти



Git

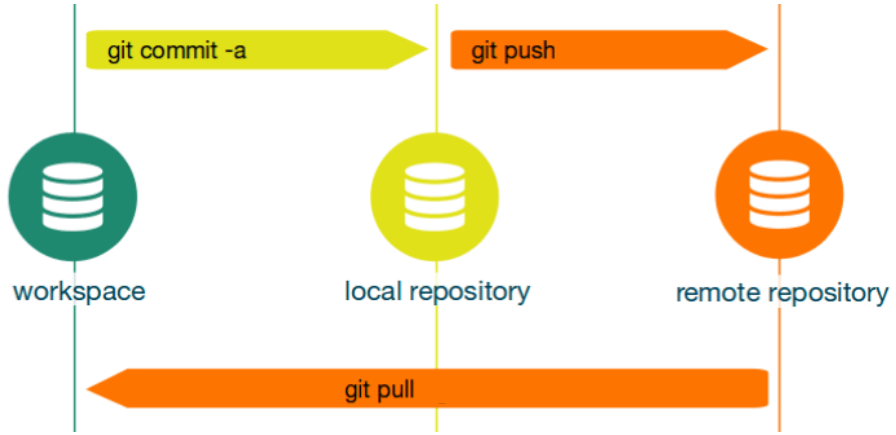
Що таке Git?

Git-репозиторій – це за своєю суттю невелика файлова система. Вона зберігає всі файли та їх версії. Папка `.git`.

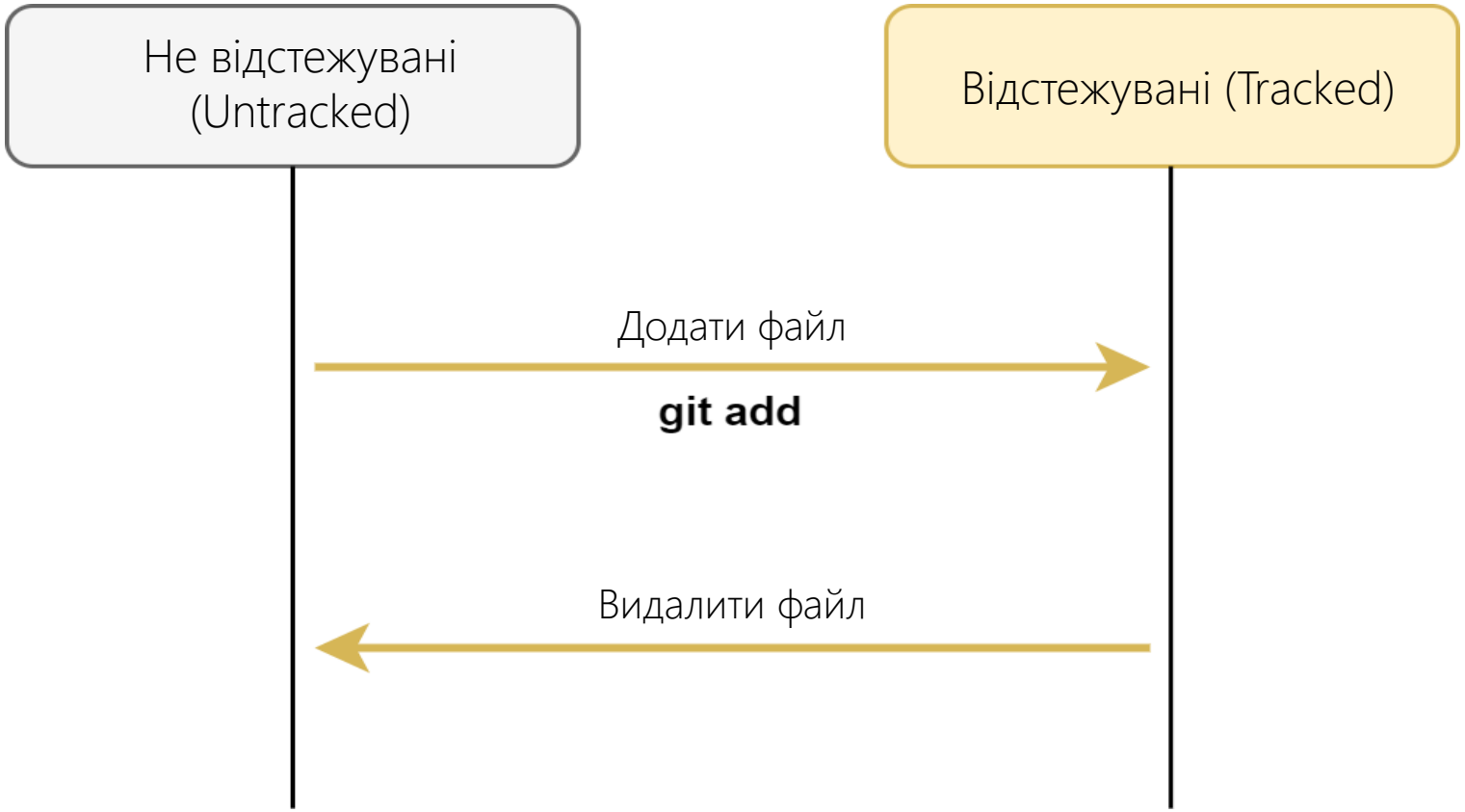


Спрощена структура Git

Workspace – це папка за якою стежить Git.
На цьому рівні розробник змінює код.



Стан файлів


Не відстежувані (Untracked)
Відстежувані (Tracked)
Додати файл
git add
Видалити файл

Основні команди Git

- git init – створює репозиторій.
- git add – додає файли в категорію відстежувані/індекс для подальшого коміту.
- git commit -a – використовується для фіксації змін у репозиторії.
- git status – виведення стану файлів.
- git log – використовується для виведення історії комітів.
- git help – виведення довідки.
- git help <command> – виведення доступних опцій використання команди.

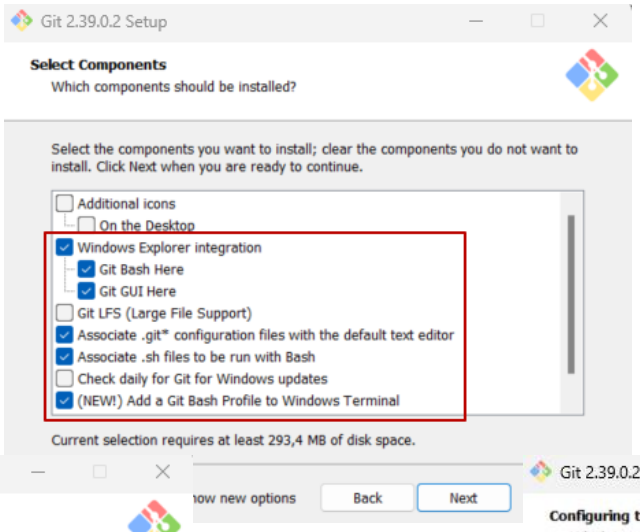
Перший коміт

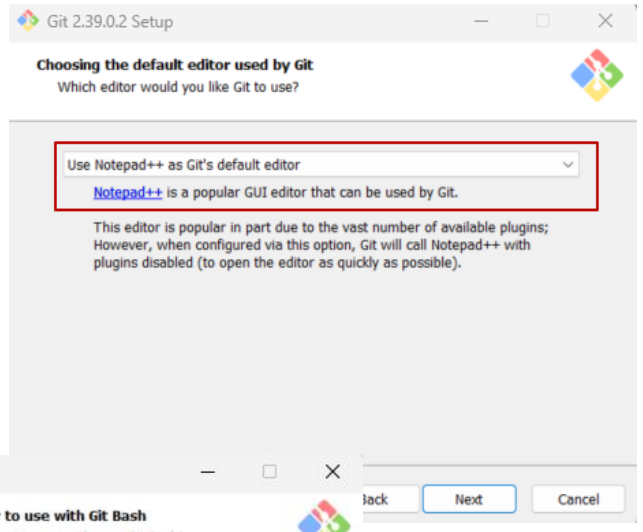
Практика

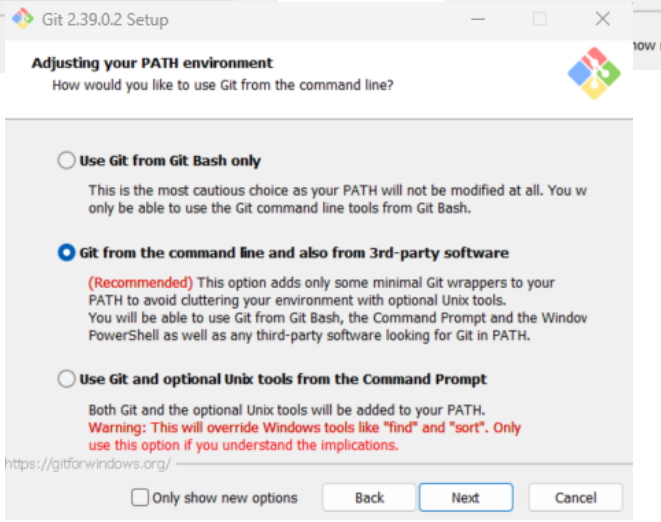
1. Встановіть git https://git-scm.com/downloads

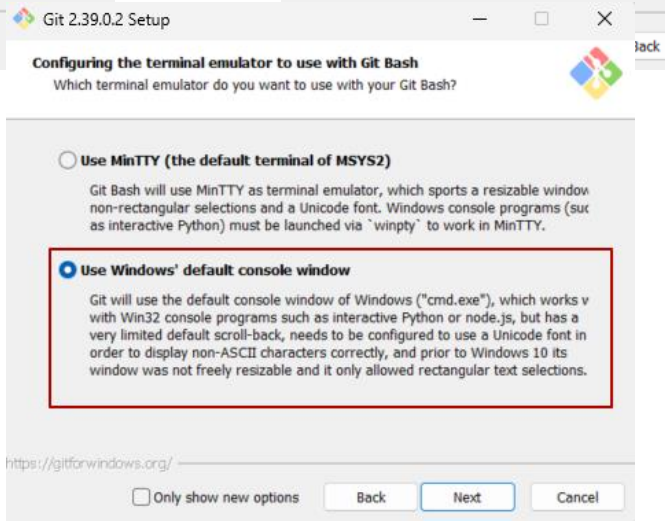
Manual QA

Практика









Налаштування Git

- git config --list --show-origin – переглянути всі свої налаштування

Для того щоби робити комміти вам необхідно вказати ваші дані

Наступні команди вам допоможуть

- git config --global user.name "John Doe"
- git config --global user.email “johndoe@example.com”

Практика

1. Створіть репозиторій (git init)
2. Створіть текстовий файл і напишіть у ньому щось
3. Перевірте статус файлу (git status)
4. Зробіть коміт (git commit –a)

5. Змініть текст файлу
6. Перевірте статус файлу (git status)
7. Зробіть коміт (git commit –a)

Що ми сьогодні вивчали

План уроку

- Архітектури систем контролю версій
- Створення репозиторію
- Стан файлів
- Основні команди
- Перший коміт
- Практика

Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

Дякую за увагу! До нових зустрічей!

TestProvider Pearson CyberBionic systematics Microsoft Partner Windows Azure EBA