



УДК 37.091.3:004

PROSPECTS OF USING GIT IN THE EDUCATIONAL AND PROFESSIONAL SPACE OF UKRAINE

ПЕРСПЕКТИВИ ВИКОРИСТАННЯ GIT В ОСВІТНЬО-ПРОФЕСІЙНОМУ ПРОСТОРИ УКРАЇНИ

Pavlyha P.D. / Павлига П.Д.*postgraduate student / аспірант*

ORCID: 0009-0007-7904-1890

*Alfred Nobel University, Dnipro, 18, Sichelavskya Naberezhna, 49000**Університет імені Альфреда Нобеля, Дніпро, Січеславська Набережна, 18, 49000*

Анотація. В роботі розглядається система Git та її застосування в освітньому процесі. Роз'яснено термінологію, сферу застосування, основні проблеми, шляхи їх вирішення в професійній діяльності. Висвітлено різницю між поняттями Git, GitHub та GitLab. Вивчено досвід закордонних університетів щодо використання систем контролю версій під час професійної підготовки майбутніх ІТ фахівців. Розглянуто шляхи введення Git в повсякденне життя студентів та викладачів. Розглянуто можливості, що надає сервіс GitHub Classroom, для організації навчального процесу. Доведено важливість та необхідність використання сучасних систем для вирішення повсякденних завдань. Запропоновано варіант зміни організації навчання студентів для ЗВО України.

Ключові слова: системи контролю версій, освіта в Україні, перспективи розвитку освіти, професійна підготовка, Git

Вступ.

Освіта в галузі інформаційних технологій є однією з найскладніших задач для педагогів. Швидкість розвитку нових технологій вимагає постійно підвищувати свою кваліфікацію, бути в курсі останніх розробок та слідкувати за новітніми тенденціями.

Однією з основних складових колективного процесу розробки програмного забезпечення, в сучасному світі, є система контролю версій: наприклад Subversion (SVN), Revision Control System (RCS), Microsoft Visual SourceSafe (VSS), Mercurial, Vazaar. Система контролю версій Git була розроблена в 2005 році, як безкоштовна заміна системі BitKeeper [1, с. 14]. З того моменту Git розвивався, вдосконалювався і, як результат, в теперішній час став одним з основних інструментів в світі ІТ. Його використання в процесі професійної підготовки в університетах – це потужний поштовх для розвитку майбутніх фахівців та підготовки їх до професійної діяльності.



Основний текст

Дослідженням питання використання систем контролю версій в навчальному процесі в закладах вищої освіти займалось багато вітчизняних та закордонних науковців. Наприклад Сергій Горчинський та Катерина Горчинська в своїй статті «Використання GitHub при вивченні дисципліни «основи веб-технологій» дослідили дидактичні можливості використання платформи для розміщення коду з підтримкою контролю версій та спільної роботи GitHub у навчальному процесі [1]. Джозеф Фелісіано, Маргарет-Енн Сторі та Олексій Загальський зробили тематичне дослідження на тему «Досвід студентів у використанні GitHub на курсах з програмної інженерії» [2], де зібрали відгуки у студентів двох курсів, що використовували GitHub в якості інструменту в навчальному процесі. Зібравши позитивні та негативні відгуки, ними були розроблені рекомендації для команди розробників сервісу GitHub щодо покращення інтерфейсу та забезпеченню необхідного функціоналу для потреб студентів та викладачів.

Термін «система контролю версій» означає програмний інструмент, що дозволяє відслідковувати зміни в коді або документах, зберігаючи історію змін. І хоча в суспільстві існує стала думка, що такі системи потрібні лише робітникам ІТ індустрії, це не є вірним. Як зазначено в книзі «GIT Pro» [3, с. 10] - “Якщо ви графічний або веб-дизайнер і хочете зберегти кожен зображення або макета, система контролю версій (далі СКВ) якраз те, що потрібно. Вона дозволяє повернути вибрані файли до попереднього стану, повернути весь проект до попереднього стану, побачити зміни, побачити, хто останній міняв щось і спровокував проблему, хто вказав на проблему і коли, та багато іншого. Використання СКВ також в цілому означає, що, якщо ви зламали щось або втратили файли, ви просто можете все виправити.”.

Існує 3 види систем контролю версій:

- 1) *Локальний*. Досі доволі поширений варіант, що передбачає збереження кожної версії в окремому файлі на локальному комп'ютері.
- 2) *Централізований*. Схожий за основною ідеєю на локальний, тільки



замість локального пристрою, використовується серверне обладнання, і доступ до сховища надається за спеціальними протоколами через локальну мережу або Інтернет. Наприклад Subversion (SVN).

3) *Децентралізований*. На відміну від централізованого варіанту, кожен з клієнтів отримує не лише запитувану версію файлів, а повністю всі файли репозиторію. В такому випадку, в разі виходу з ладу серверного обладнання, чи втрати даних з носія інформації, будь-яка копія на клієнтському пристрої може бути використана для відновлення серверу. Наприклад Git.

Очевидно, що децентралізовані системи мають найбільшу популярність серед розробників програмного забезпечення і тому використання в освітньому процесі саме Git є найбільш ефективним.

Також розглядаючи систему контролю версій Git, потрібно відрізнити Git, GitLab та GitHub. Git – це базова технологія, за її допомогою можна налаштувати повноцінну систему контролю версій, використовуючи допоміжні утиліти, на кшталт gitolite, в якості серверного сервісу. Для забезпечення потреб розробників, цього достатньо, але такий варіант вимагає налаштування та підтримання роботи системним адміністратором.

В цей час GitLab являє собою програмний комплекс, що має ядром сервер для системи контролю версій, і на додачу до цього надає веб-інтерфейс для налаштування, керування та різного роду аналітики. Це безкоштовне програмне забезпечення, що може бути встановлене на серверне обладнання чи персональний комп'ютер і для повноцінного функціонування потребує доступу до локальної мережі (в разі якщо сервер та всі учасники проекту знаходяться в одній локальній чи віртуальній мережі) чи до Інтернет. Також для GitLab необхідне обслуговування і налаштування, та, завдяки графічному веб інтерфейсу, це не потребує навичок в роботі з командним рядком, тому може бути виконано без допомоги системного адміністратора.

З точки зору користувача, GitHub відрізняється від GitLab тим, що це хмарний сервіс, що підпадає під категорію SaaS (software as service, програмне



забезпечення як сервіс), він не має опції розгортання серверу на обладнанні користувачів і доступний лише через офіційний портал, що розташований в мережі Інтернет.

Розглянемо системи контролю версій в ракурсі їх використання в навчально-освітньому процесі в закладах вищої освіти, на прикладі Гарвардського університету, одного з провідних закладів освіти в США. Якщо переглянути сторінку курсу CS107 / AC207: «Systems Development for Computational Science» - «Розробка систем для комп'ютерних наук» [4], можна зазначити наступні основні сфери та напрямки використання Git:

- Надання доступу студентам до навчального матеріалу.
- Організація процесу здачі лабораторних завдань.
- Створення платформ для виконання групових завдань.
- Забезпечення контролю, аналізу та стійкості до загублення інформації, будь то курсові, магістерські роботи, дисертації чи тексти наукових статей.
- Організаційні питання навчального процесу. Наприклад висвітлення розкладу занять, консультацій, екзаменів та заліків студентам та викладачам.
- Швидка передача файлів та комунікація між викладачами та студентами. Один раз налаштувавши репозиторій на локальному комп'ютері, процес обміну файлами займає 1-2 кліка. Наприклад, студент має запитання щодо тексту програми. Він робить завантаження на сервер свого варіанту коду з питаннями у коментарях. Викладач періодично запитує сервер на предмет наявності завантажень від студентів, і один рухом може отримати запитання від студента на своєму пристрої (це можна робити навіть зі смартфона в будь-якому місці та в зручний час) розглянувши питання, він може відповісти новою версією файлу, або просто коментарем. Студент отримав відповідь зможе швидко побачити, які зміни запропоновані, та як викладач пропонує вирішити його питання.



Для вирішення таких задач, сервіс GitHub має спеціальний розділ – Classroom. На офіційному сайті [5] можна знайти детальну інформацію, та безкоштовно створити власний курс. Під час створення курсу є можливість з'єднати його з вже існуючим Гугл класом, завдяки чому автоматично спрацює механізм надання доступу до курсу всім учасникам Гугл класу. Інтерфейс створення нового курсу GitHub показано на рисунку 1.



Рисунок 1 – Інтерфейс створення курсу GitHub

Джерело: [5]

Більша частина функціоналу доступна без додаткових налаштувань, та, якщо представник університету пройде перевірку на належність до навчального закладу, то відкриються додаткові можливості, наприклад - Codespaces Education. Це спеціальна функція, що надає віртуальну середу розробки для студентів, це означає, що студентам та викладачам не потрібно витратити час на налаштування інструментів розробки. Але і без неї, GitHub Classroom має достатньо функціоналу для організації навчального процесу.

Створивши курс, можна додати завдання до нього. Інтерфейс форми нового завдання (Рисунок 2) максимально простий, але має необхідні для викладача налаштування.



Let's set up the basics for your assignment.

Assignment title *

Custom repository prefix *

This will prefix each GitHub repository that is created for this assignment. May only contain alphanumeric characters, underscores or hyphens.

Deadline

(Optional) If left blank, there will be no deadline. Date format: 31.12.2022, 23:59

This is a cutoff date
If selected, the student will lose write access to their repository after the date is reached.

Individual or group assignment

Рисунок 2 – Інтерфейс створення завдання курсу

Джерело: [5]

Назва завдання, кінцевий строк задачі та вид завдання (індивідуальний чи груповий) – ось основні параметри. Також можна вказати:

- префікс для назв репозиторіїв, що будуть створені для цього завдання студентами. Дуже зручна функція, що забезпечить комфортне середовище для викладача та студента, і замість збереження на комп'ютері файлів на кшталт "Іванов лабораторна 1.doc", "Лаб. раб. перша Мартинюк.doc" "л.р.1_Петров.doc", "Lab.N_1_Pavliga.doc", буде структурований список:

- laboratorna_1_Ivanov
- laboratorna_1_Martynyuk
- laboratorna_1_Petrov
- laboratorna_1_Pavliga

і такий порядок забезпечується автоматично, без потреби контролю з боку викладача.



- Cutoff date – Дата закінчення. Надає можливість відключення доступу студентам до репозиторіїв завдань після настання кінцевого терміну здачі завдання.

- Далі можна обрати репозиторій, що буде слугувати шаблоном для студентів при початку роботи над завданням. В ньому можна розмістити методичні вказівки, теоретичний матеріал, фрагменти програмного коду тощо.

- Також потрібно вказати налаштування доступу до репозиторіїв із завданнями, що будуть здаватись студентами. Існує два варіанта:

1. Кожен із студентів має доступ до репозиторіїв інших одногрупників.
2. Кожне здане завдання відображається лише у викладача та у студента, що його здав.

- Ще одна з дуже корисних функцій – це авто тестування. Мається на увазі автоматична перевірка програм після здачі завдань студентами. Викладач може налаштувати входні параметри та результати, що мають бути прораховані програмою, що здав студент. Таким чином перевірка завдання відбувається за кілька секунд і студент миттєво отримує оцінку. Це не є обов'язковою функцією і може використовуватись за потреби.

Тож, використовуючи такий функціонал, можемо рекомендувати використовувати GitHub Classroom, задля організації навчального процесу на найвищому рівні, забезпечити більш комфортне середовище для студентів, зменшити навантаження на викладачів, мати прозорий процес оцінювання знань та комунікації викладачів зі студентами. Такі зміни можуть бути впроваджені з мінімальними витратами часу та без додаткових витрат, що робить їх максимально корисними та привабливими для використання в процесі професійної підготовки майбутніх фахівців в сфері інформаційних технологій. Для успішного використання цього сервісу, потрібне лише розробити методичні вказівки для студентів по створенню облікових записів на офіційному порталі GitHub та основних функцій системи контролю версій – створення нового репозиторію, клонування репозиторію з віддаленого серверу, перевірка статусу віддаленого репозиторію, завантаження останніх змін з віддаленого



репозиторію, збереження змін в локальному репозиторії, надсилання нової версії локального репозиторію на віддалений сервер. В цілому базові функції можливо надати як вступну частину курсу з програмування на будь-якій мові з навантаженням 2-4 академічні години.

Висновки.

Під час дослідження були вивчені основні види систем контролю версій – локальні, централізовані, децентралізовані. Виявлено, що третій варіант – децентралізовані системи мають найбільше розповсюдження в професійному світі розробників програмного забезпечення сьогодення. Було роз'яснено різницю між Git – базова технологія контролю версій, GitLab – програмне забезпечення, що дозволяє розгорнути сервер на сторонньому обладнанні та GitHub – хмарний сервіс, що надає доступ до свого функціоналу через офіційний веб-портал. Наведено приклад, як в Гарвардському університеті, США використовується сервіс GitHub, для організації навчального процесу з дисципліни «Розробка систем для комп'ютерних наук». Проаналізовано функціонал спеціального розділу сервісу GitHub – Classroom, що надає дуже гнучкий та широкий функціонал для створення освітнього середовища для навчальних дисциплін – починаючи з комунікації між студентами та викладачами та закінчуючи автоматичною перевіркою виконаних завдань за допомогою підготовлених тестів.

В результаті були сформовані рекомендації щодо застосування сервісу GitHub Classroom під час професійної підготовки фахівців з ІТ в закладах вищої освіти України.

Література:

1. Горчинський С. ВИКОРИСТАННЯ GITHUB ПРИ ВИВЧЕННІ ДИСЦИПЛІНИ «ОСНОВИ ВЕБ-ТЕХНОЛОГІЙ» [Електронний ресурс] / Сергій Горчинський, Катерина Горчинська // Professional Pedagogics. – 2022. – Т. 1, № 24. – С. 195–202. – Режим доступу: <https://doi.org/10.32835/2707-3092.2022.24.195-202> (дата звернення: 21.09.2024). – Назва з екрана.



2. Feliciano J. Student experiences using GitHub in software engineering courses: a case study [Electronic resource] / Joseph Feliciano, Margaret-Anne Storey, Alexey Zagalsky // 38th International Conference on Software Engineering Companion (ICSE '16): Proceedings, NY. – New York, 2016. – P. 422–431. – Mode of access: <https://doi.org/10.1145/2889160.2889195> (date of access: 21.09.2024). – Title from screen.

3. Chacon S. Pro GIT. Everything you need to know about git [Electronic resource] / Scott Chacon, Ben Straub. – [S. l. : s. n.]. – Mode of access: <https://git-scm.com/book/en/v2> (date of access: 21.09.2024). – Title from screen.

4. CS107 / AC207: Systems Development for Computational Science [Electronic resource] // GitHub. – Mode of access: <https://harvard-iacs.github.io/2021-CS107/> (date of access: 21.09.2024). – Title from screen.

5. GitHub Classroom [Electronic resource] // GitHub Classroom. – Mode of access: <https://classroom.github.com/> (date of access: 21.09.2024). – Title from screen.

Abstract. *The paper discusses the Git system and its application in the educational process. The terminology, scope, main problems, and ways to solve them in professional activities are explained. The difference between the concepts of Git, GitHub and GitLab is highlighted. The experience of foreign universities in the use of version control systems in the professional training of future IT professionals is studied. The ways of introducing Git into the everyday life of students and teachers are considered. The opportunities provided by the GitHub Classroom service for organising the educational process are considered. The importance and necessity of using modern systems for solving everyday tasks is proved. The options for changing the content of disciplines for Ukrainian higher education institutions are proposed.*

Key words: *version control systems, education in Ukraine, prospects for the development of education, professional training, Git*

Науковий керівник: Осадчий В.В., д. пед. н., проф.

Стаття надіслана: 23.09.2024 г.

© Павлига П.Д.