

ОСОБЛИВОСТІ ВИКОРИСТАННЯ СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ ПРИ РОЗРОБЦІ ОСВІТНІХ ПРОЄКТІВ З ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

SPECIFICS OF THE VERSION CONTROL SYSTEM USAGE IN THE DEVELOPMENT OF EDUCATIONAL PROJECTS IN SOFTWARE ENGINEERING

Ця стаття докладно розглядає використання системи контролю версій Git у дистанційному навчанні інженерії програмного забезпечення та висвітлює новаторський підхід у навчанні для покращення практичних навичок студентів при розробці освітніх проєктів.

У цій статті розкриваються такі поняття як дистанційне навчання, освітня проєктна робота, інструменти дистанційного навчання. Описуються основні аспекти використання системи контролю версій як інструмент дистанційного навчання при необхідності групової роботи над проєктом. Розкривається спосіб формування необхідних навичок роботи у команді, за відсутності можливості проводити практичні у аудиторії. Фокус статті зосереджений на прикладі імплементації різних систем контролю версій у навчальний процес. Стаття розглядає конкретні переваги використання системи контролю версій, такі як можливість спільної роботи, відслідковування змін, рецензування коду та інші.

Базуючись на аналізі актуальних затребуваних навичок інженерів програмного забезпечення пропонується підхід до організації проєктного практичної роботи в умовах дистанційного навчання. Демонструються стратегії та методики використання інструментів управління версіями коду для актуалізації навчального процесу та підвищення ефективності спільної роботи студентів над проєктами. Висвітлюються основні можливості та переваги таких систем. Додатково, стаття розглядає можливість інтеграції системи контролю версій у існуючі навчальні програми та плани, а також варіанти оцінювання студентських досягнень з використанням цієї технології. Розглядаються також інші інструменти, які можуть доповнювати систему контролю версій для забезпечення комплексного підходу до навчання програмування та розробки ПЗ.

Детально розглядається процес підготовки викладачів та студентів перед використанням систем контролю версій у навчальному процесі. Надаються рекомендації щодо послідовності необхідних кроків для активного залучення студентів у процес вивчення та застосування систем контролю версій. Крім того, наведені приклади реальних сценаріїв використання систем у навчанні.

Ключові слова: дистанційне навчання, цифровий інструмент, системи контролю версій, Git, практичні навички, інноваційна педагогічна діяльність.

This article takes a closer look at the implementation of Git in software engineering distance learning and highlights an innovative educational approach to improve students' practical skills in the development of projects.

This article covers such concepts as distance learning, project work, and distance learning tools. It describes the main aspects of using a version control system as a distance learning tool when group work on a project is required.

The article reveals a way to develop the necessary teamwork skills in the absence of the possibility of conducting practical classes in the classroom. The article focuses on an example of implementing various version control systems in the educational process. The article discusses the specific advantages of using a version control system, including the ability to collaborate, track the changes, review the code, and others.

Based on the analysis of the current required skills of software engineers, an approach to the organization of project-based practical work in distance learning is proposed. It demonstrates strategies and techniques for using code version control tools to modernize the educational process and increase the efficiency of students' collaboration on projects. In addition, the article discusses the possibilities of integrating branch version control into existing curriculum and syllabi, and the ways in which this technology may be used to evaluate student achievement. It also discusses several other tools that can complement the version control system for a more comprehensive approach to teaching programming and software development.

A comprehensive analysis of the process by which teachers and students are trained to use version control systems in their educational process is presented. It also highlights the main features and advantages for such version control systems. Recommendations are given on the sequence of necessary steps to actively involve students in the process of studying and applying version control systems. In addition, the author provides examples of real-life scenarios of using the systems in education.

Key words: distance learning, digital tool, version control systems, Git, practical skills, innovative pedagogical activity.

УДК 378

DOI <https://doi.org/10.32782/2663-6085/2024/70.1.40>

Грибик Т.Т.,

аспірант кафедри педагогіки та інноваційної освіти
Національного університету
«Львівська політехніка»

Постановка проблеми у загальному вигляді.

У сучасному світі програмне забезпечення є невідмінною складовою практично будь-якого сфери життя. Інженерія програмного забезпечення (ІПЗ) відіграє важливу роль у розробці та підтримці програмних продуктів, що задовольняють потреби користувачів. Однак, навчання ІПЗ вимагає практичного досвіду, включаючи участь у реальних

проєктах. У своїй професійній діяльності інженери програмного забезпечення, інженери-розробники та інші фахівці зі сфери технічних наук часто працюють над проєктами у команді. Ефективне керування проєктом допомагає інженерам забезпечити високу якість результатів своєї роботи. Навички ефективного керування проєктами допоможуть студентам стати більш

привабливими кандидатами для роботодавців та підготують їх до майбутньої професійної діяльності. Однак, для успішної спільної роботи та навчання в цьому контексті необхідним є відповідний інструментарій. Використання систем контролю версій (СКВ) є важливим елементом цього процесу, оскільки воно забезпечує ефективне керування змінами у кодї та сприяє спільній роботї над проектом. Система контролю версій виявляється незамінною у такому середовищі, надаючи студентам та викладачам ефективні засоби спільної роботи та навчання. Перехід на віртуальну форму навчання вносить ряд викликів, серед яких важливим є ефективне управління версіями програмного коду під час спільної роботи студентів над проектами. Управління версіями коду вимагає систематичного відстеження змін, зберігання історії редагувань, а також забезпечення можливості одночасної роботи декількох учасників над одним проектом.

В контексті дистанційного навчання інженерії програмного забезпечення виникають питання щодо ефективності та зручності використання СКВ студентами та викладачами.

Перш за все, існує проблема недостатньої освіти стосовно використання СКВ серед студентів. Багато з них можуть бути недостатньо обізнані з цими системами або навіть незнайомі з їх існуванням, що ускладнює спільну роботу над проектами. Друга проблема стосується складнощів у спільній роботї над кодом в онлайн-середовищі. Сучасні системи навчання

часто не надають зручних інструментів для спільного редагування коду та відстеження змін. Це може призвести до конфліктів у кодї, втрати даних або навіть зупинки робочого процесу.

Крім того, існує проблема забезпечення ефективної комунікації між учасниками проекту. У віртуальному середовищі студенти можуть мати обмежений доступ до можливостей спілкування та обміну інформацією, що може ускладнити вирішення проблем та координацію дій. Отже, перед викладачами і студентами постає завдання розробити та запровадити ефективні стратегії та інструменти управління версіями програмного коду в контексті дистанційного навчання інженерії програмного забезпечення. Розв'язання цих проблем може покращити якість навчання, збільшити продуктивність роботи над проектами та підготувати студентів до викликів реального ринку роботи у сфері програмування.

Аналіз останніх досліджень та публікацій. Науковці, які досліджують використання систем контролю версій у навчанні студентів інженерів, зазвичай звертають увагу на кілька ключових аспектів. Вони вивчають, як використання систем контролю версій може стимулювати зацікавленість студентів у процесі навчання через можливість спільної роботи та внесення власного внеску у проекти.

Наприклад у науковій роботї Г.Вульфа [6] висвітлюються переваги використання Git для студентів та викладачів, зокрема, зручність спільної роботи, відстеження змін та можливість створення версій курсового матеріалу. Стаття також висвітлює виклики, пов'язані з впровадженням Git як основи навчального середовища, а також можливості подальшого вдосконалення цього підходу до навчання. Автор відзначає, що Git, як розподілена система контролю версій, є відкритою та гнучкою платформою, яка дозволяє широкий спектр можливостей для організації навчального процесу.

Ґрунтовно описується поняття систем управління версіями та їх значення в навчальному процесі для студентів у статті Гладкої Л.І. [3]. Стаття проводить порівняльний аналіз різних систем управління версіями, включаючи Git, Subversion, Mercurial та інші, зазначаючи їхні переваги та недоліки.

Найбільше уваги можливим викликам при використанні систем контролю версій приділяється у науковій роботї В.Ізомоттена та М. Кочеза [5]. Стаття ідентифікує основні складнощі та плутанини, з якими стикаються студенти під час навчання системі контролю версій Git. Автори аналізують основні причини, які призводять до плутанини та непорозумінь під час вивчення Git, такі як складність концепцій, відмінності від інших інструментів та неочевидність деяких функцій. Автори обговорюють різні методи та підходи, які можуть допомогти студентам подолати складнощі вивчення Git, такі як використання демонстрацій, практичні завдання та індивідуальне керівництво.

Мета статті полягає у розробці методики використання інструментів управління версіями коду для вдосконалення навичок командної роботи студентів в умовах дистанційного навчання. Розроблення авторської методики ставить за ціль створення можливості розвинути такі необхідні професійні навички студентів інженерів, як навички співпраці та комунікації, командної роботи та ефективного керування проектами і версіями коду. Зокрема основний акцент робиться на тому, що така методика повинна забезпечити майбутнім фахівцям достатній рівень практичних навичок, що відповідають стандартам галузі інженерії програмного забезпечення.

Виклад основного матеріалу. Сучасна освіта в галузі інформаційних технологій ставить перед собою завдання не лише в передачі теоретичних знань, але й формування практичних навичок, які є важливими для успішної кар'єри майбутніх фахівців. Освітня парадигма ставить перед викладачами завдання підготувати майбутніх фахівців, які здатні ефективно працювати у галузі програмного забезпечення, яка перебуває у постійному розвитку.

Інтеграція систем контролю версій (СКВ) у навчальний процес стає ключовим чинником для досягнення цієї мети. Оскільки це дозволяє

студентам отримати цінний досвід управління проектами, спільної роботи та розробки програмного забезпечення.

Використання систем контролю версій (СКВ) у дистанційному навчанні має численні переваги, які сприяють покращенню якості навчання та розвитку навичок студентів. Зокрема варто відзначити такі основні переваги:

- колаборативна робота: СКВ дозволяють студентам спільно працювати над проектами, навіть якщо вони розташовані в різних місцях. Кожен студент може вносити зміни до коду та спільно працювати над його розвитком. Це сприяє збільшенню продуктивності та співпраці між учасниками проекту.

- історія змін: СКВ автоматично відстежують історію змін, що дозволяє студентам та викладачам відслідковувати еволюцію проекту та виконані зміни. Це допомагає виявляти та виправляти помилки, а також аналізувати навички програмування.

- автоматичне злиття коду: СКВ автоматично зливають зміни, внесені кількома учасниками, що запобігає конфліктам у коді. Це зменшує ризик втрати даних та полегшує спільну роботу над проектами.

- відокремлені гілки: СКВ дозволяють створювати відокремлені гілки для виконання різних завдань чи реалізації різних функцій. Це сприяє організації роботи та забезпечує можливість проведення експериментів без впливу на основний код проекту.

- відслідковування прогресу: Викладачі можуть відслідковувати прогрес студентів, аналізуючи їхні внески у репозиторій проекту. Це дозволяє вчасно виявляти проблеми та надавати додаткову підтримку, а також оцінювати індивідуальний внесок кожного студента.

Перед тим як розпочати роботу над проектами, важливо навчити студентів використовувати основні функції СКВ, такі як створення “комітів”, створення гілок та злиття змін (рис.1). Для цього можна провести спеціальний урок або вебінар, в якому показати на прикладах як працює система контролю версій.



Рис. 1. Схематичне зображення гілок та їх злиття в системі контролю версій

Важливо створити чітку структуру репозиторію для проектів. Це допоможе студентам легко знаходити необхідний код та ресурси. Наприклад, можна створити окремі теки для документації, тестових скриптів та інших компонентів проекту.

Потрібно провести розподіл ролей та відповідальності серед студентів у команді. Наприклад, один студент може бути відповідальний за управління гілками та злиттям змін, інший – за написання тестів, і так далі. Це допоможе забезпечити ефективну роботу над проектом.

Заохочення студентів до регулярних “комітів” коду та огляду змін своїх колег, дозволить швидко виявляти та виправляти помилки, а також взаємно навчатися від один одного.

Вибір зручних інструментів керування версіями, таких як Git з веб-інтерфейсом, можуть полегшити роботу з репозиторіями та сприяти кращому сприйняттю матеріалу студентами.

Заохочуйте студентів активно співпрацювати та спілкуватися між собою через інструменти СКВ. Це може включати обговорення проблем в коментарях

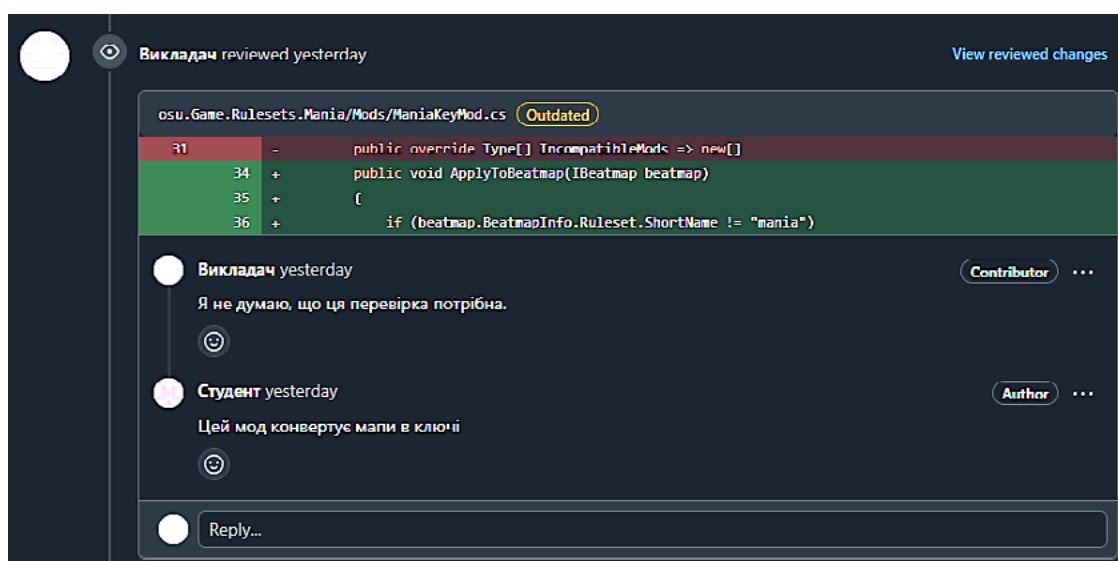


Рис. 2. Приклад контрибуції викладача та студентів

до комітів, використання інструментів для спільної роботи та обміну інформацією. Надайте студентам зворотний зв'язок щодо їхнього внеску у проект та допомагайте їм вирішувати труднощі, з якими вони можуть зіткнутися під час роботи з СКВ.

Всі студенти та викладач повинні мати встановлені git на робочому комп'ютері. Викладач повинен створити репозиторій проекту, в якому будуть працювати студенти. Після чого зробити перший "коміт" з описом проекту та початковий код. У свою чергу студенти, які розробляють проект повинні скопіювати репозиторій на свою локальну машину. Кожен студент який отримав завдання повинен зробити свою окрему гілку, в якій повинен виконувати своє завдання. Після завершення зробити "пул реквест" в репозиторій з проектом.

Інші студенти повинні зробити огляд коду та підтвердити внесення змін або залишити коментарі (рис 2.) для покращення коду. Проте остаточне рішення про злиття гілок повинен приймати викладач.

Висновки. Використання систем контролю версій у навчальних проєктах з інженерії програмного забезпечення є не лише доцільним, але й необхідним. Воно сприяє не тільки покращенню рівня знань студентів у галузі розробки програмного забезпечення, а й розвитку їхніх навичок співпраці, комунікації та якості роботи в команді. Зважаючи на високу потребу у впровадженні іновативних дистанційних методів навчання, особливо таких, що адаптують дистанційне навчальне середовище для здобуття студентами практичних навичок, впровадження СКВ у навчальні програми з ІПЗ має великий потенціал для підготовки якісних та компетентних інженерів програмного забезпечення.

У контексті дистанційного навчання інженерії програмного забезпечення система контролю версій Git є незамінним інструментом для спільної роботи студентів та викладачів. Використання системи контролю версій у дистанційному освітньому процесі представляє собою вагомий інноваційний методологічний прийом, який ефективно вирішує ряд завдань, стоячих перед сучасною освітою.

СКВ сприяє спільній роботі над проєктами. У віртуальному навчальному середовищі студенти та викладачі мають можливість одночасно працювати над матеріалами курсу або проєктами, вносячи зміни та коментуючи їх. Це стимулює колаборацію та взаємодію, відтворюючи атмосферу співпраці, характерну для традиційного навчального процесу.

СКВ дозволяє відстежувати всі зміни в матеріалах курсу. Завдяки цьому студенти та викладачі можуть легко виявляти та виправляти помилки, а також здійснювати контроль над еволюцією курсу. Це робить процес навчання більш прозорим та дозволяє швидко реагувати на потреби студентів.

Версіонування матеріалів курсу – є ключовим для забезпечення якості та стабільності

навчального процесу. Git дозволяє зберігати різні версії курсових матеріалів, що дозволяє студентам та викладачам повертатися до попередніх варіантів та порівнювати їх для поліпшення результатів.

Крім того, Git забезпечує зручний обмін матеріалами та автоматизацію процесів. Завдяки репозиторіям та гілкам, студенти та викладачі можуть ефективно організувати та керувати навчальними ресурсами. Автоматизація певних завдань, таких як злиття змін чи виправлення конфліктів, дозволяє зосередитися на навчанні та творчому процесі.

Використання Git у дистанційному навчанні сприяє інтерактивному підходу до освіти, покращенню комунікації, організації робочого процесу та підвищенню ефективності навчання. Він дозволяє студентам активно залучатися до процесу навчання, набуваючи практичних навичок роботи з сучасними інструментами та підвищуючи їхню конкурентоспроможність на ринку праці.

Отже, використання Git у дистанційному освітньому процесі не лише полегшує спільну роботу та взаємодію між студентами та викладачами, але й сприяє якісному та ефективному навчанню, відповідаючи вимогам сучасної освіти.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Mosiiuk O. Особливості навчання майбутніх учителів інформатики роботі із системою контролю версій Git. *Науковий вісник Ужгородського університету. Серія: «Педагогіка. Соціальна робота»*, 2021. Вип. 2(49), С. 107–110. URL: <https://doi.org/10.24144/2524-0609.2021.49.107-110> (дата звернення: 02.04.2024).
2. Герасименко І. В. Використання технологій дистанційного навчання в підготовці майбутніх бакалаврів комп'ютерних наук : дис. на здобуття наук. ступеня канд. пед. наук : 13.00.10. Черкаси, 2014. 302 с. URL: <https://lib.iitta.gov.ua/705004/1/dis%20Герасименко.pdf>. (дата звернення: 20.03.2024).
3. Гладка Л. І. Огляд систем управління версіями як інструменту командної роботи студентів. *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку* : зб. матеріалів доп. учасн. Всеукр. наук.-практ. Інт.-конф., Черкаси, 2018. С. 186-189.
4. Isomöttönen, V., Cochez, M. Challenges and Confusions in Learning Version Control with Git. *Information and Communication Technologies in Education, Research, and Industrial Applications. ICTERI*, 2014. Vol. 469. P. 178–193. URL: https://doi.org/10.1007/978-3-319-13206-8_9 (дата звернення: 20.03.2024).
5. G. Wolf, "Using the Git Version Control System to replace a Learning Management System," *Revista Iberoamericana de Tecnologías del Aprendizaje. IEEE*, 2024. Vol. 19. P. 24-32.
6. Burns, M. Distance Education for Teacher Training: Modes, Models and Methods. 2023. 427 p. URL: https://www.researchgate.net/publication/373644565_Distance_Education_for_Teacher_Training-Second_Edition. (дата звернення: 20.03.2024).