

ВИКОРИСТАННЯ СИСТЕМ КОНТРОЛЮ ВЕРСІЙ ЯК НЕВІД'ЄМНОЇ ЧАСТИНИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

Проаналізовано важливість застосування систем контролю версій у розробці програмного забезпечення. Описано ключові особливості та приклади застосування.

Ключові слова: програмне забезпечення, розробка програмного забезпечення, система контролю версій.

Abstract

The importance of using version control systems in software development is analyzed. The key features and examples of application are described.

Keywords: software, software development, version control system

Вступ

Системи контролю версій стали однією з ключових технологій, яка забезпечує збереження історії змін, координацію командної роботи, а також покращення надійності та безпеки розроблюваних проєктів. Їх використання стало стандартом у сучасному програмуванні, де навіть незначні зміни в коді проєкту можуть мати критичне значення для його функціональності. Системи контролю версій дозволяють не лише відслідковувати та документувати процес розробки, а й забезпечують можливість паралельної роботи багатьох розробників над одним проєктом без ризику втрати або пошкодження даних [1]. У сучасному світі, де розробка програмного забезпечення часто відбувається в розподілених командах, значення систем контролю версій стає особливо очевидним.

Основна частина

Системи контролю версій надають можливість зберігати повну історію змін програмного коду, починаючи від створення першого файлу й до кожної правки, яка вноситься в проєкт [2]. Це означає, що всі внесені зміни залишаються задокументованими, що є надзвичайно важливим при роботі над проєктами. Можливість повернення до попередніх версій файлів дозволяє розробникам не лише знаходити джерела помилок, а й випробовувати різні рішення, порівнюючи їх ефективність без ризику порушити роботу основної версії програми. Наприклад, якщо новий алгоритм виявляється менш ефективним, ніж попередній, його можна легко видалити та відновити попередню версію коду. Завдяки даній функції розробникам вдаються більш креативні рішення, адже для впровадження нових не знайомих раніше функцій, не потрібно ставити під ризик стабільність усього проєкту.

Особливої уваги заслуговує роль систем контролю версій у командній розробці програмного забезпечення. Сучасні програмні продукти рідко створюються однією особою, натомість над ними працюють цілі команди, часто розподілені по різних країнах та часових поясах. Системи контролю версій дозволяють кожному учаснику проєкту вносити зміни незалежно від інших, а потім об'єднувати ці зміни в єдину кодову базу. Відгалуження, які також відомі як «гілки», дозволяють працювати над різними функціональностями паралельно, не створюючи конфліктів у головній версії проєкту. Кожен розробник має можливість створити свою гілку проєкту, що забезпечує гнучкість та дозволяє значно прискорити процес розробки [3].

Крім того, системи контролю версій допомагають покращити надійність програмного забезпечення завдяки можливості створення резервних копій та маркерів випусків: v.01, v.02. і тд. Створення контрольних точок або релізів дозволяє чітко зафіксувати стабільні версії програми, які можуть бути використані для тестування або розгортання. Якщо нововведення виявляються нестабільними або небажаними, завжди можливо повернутися до попередньої перевіреної версії. У

випадках коли програмне забезпечення повинно оновлюватися та підтримуватися протягом зазначеного часу, ця функція відіграє одну з ключових ролей.

Окрім забезпечення надійності, системи контролю версій сприяють покращенню прозорості проекту. Кожна зміна в кодовій базі документується із зазначенням часу, автора та описом внесених правок. Створена документація дозволяє розробникам легко відслідковувати процес розробки, переглядати зміни, зроблені іншими учасниками команди, та швидко розуміти, які частини проекту були модифіковані та хто відповідальний за визначену частину коду [4]. Така прозорість допомагає організовувати зменшення кількості помилок, оскільки полегшує перевірку коду.

Системи контролю версій зручні не лише в командній розробці, а й в особистих проектах. Вони дозволяють індивідуальним розробникам створювати контрольні точки на різних етапах розробки, експериментувати з новими функціями без ризику пошкодження робочої версії, а також повертатися до попередніх рішень у разі необхідності. Такий підхід робить розробку гнучкішою та менш стресовою, оскільки дозволяє зберігати кожний крок у процесі створення програмного забезпечення. Такий крок слугує підказкою розробнику про теперішній стан проекту, його історію та напрям розробки.

Важливим аспектом використання систем контролю версій є їх інтеграція з іншими інструментами розробки, такими як системи автоматичного тестування, управління завданнями та безперервного розгортання. Інтеграція дозволяє створювати цілісні процеси розробки, в яких зміни автоматично перевіряються на наявність помилок та конфліктів перед злиттям у головну кодову базу, що значно підвищує якість програмного забезпечення та робить процес розробки більш організованим та ефективним.

Висновок

На основі проведеного дослідження, можна зробити висновок, що системи контролю версій є необхідними у теперішній сфері розробки програмного забезпечення, тому що забезпечують структурованість, надійність та ефективність процесу створення програмних продуктів. Їх використання не лише спрощує відстеження змін та виправлення помилок, а й сприяє ефективній документації, що є особливо важливим у проектах. Уміння користуватися системами контролю версій стало обов'язковою навичкою для кожного розробника, адже саме вони дозволяють поєднувати індивідуальні зусилля в єдиний, якісний програмний продукт. Завдяки таким системам, програмування стає не лише процесом написання коду, а й структурованим підходом до створення надійного та функціонального програмного забезпечення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Version control Wiki. URL: https://en.wikipedia.org/wiki/Version_control (data of access 19.03.2025).
2. В. О. Кузьмич, О. В. Коваль, Р. А. Тараненко. Управління версіями програмних засобів проекту. КПІ ім. Ігоря Сікорського. Київ. 2023. – 9 с.
3. What is version control. URL: <https://www.atlassian.com/git/tutorials/what-is-version-control> (data of access 19.03.2025).
4. Version control systems. URL: <https://www.geeksforgeeks.org/version-control-systems> (data of access 19.03.2025).
5. Version control software: An overview. URL: <https://bitbucket.org/product/version-control-software> (data of access 19.03.2025).

Колодій Владислав Віталійович – студент групи ІПІ-24м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: vladkolodiy2014@gmail.com

Романюк Оксана Володимирівна – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: romaniukoksana@gmail.com

Vlad Kolodii – student of group ІПІ-24m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vladkolodiy2014@gmail.com

Oksana Romaniuk – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksana@gmail.com