

ПЕРЕВАГИ ЗАСТОСУВАННЯ PERFORCE ЯК СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ У РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

Проаналізовано переваги застосування Perforce у розробці програмного забезпечення. Описано ключові особливості та відмінності з відомими рішеннями систем контролю версій.

Ключові слова: програмне забезпечення, Perforce, система контролю версій.

Abstract

The advantages of using Perforce in software development are analyzed. Key features and differences from known version control system solutions are described.

Keywords: software, Perforce, version control system.

Вступ

Системи контролю версій – це програмні інструменти, які допомагають розробникам відслідковувати зміни у вихідному коді та цифрових ресурсах, забезпечуючи збереження різних версій проекту [1]. Вони дозволяють відновити попередні стани проекту, порівнювати зміни та координувати роботу команд. Такі системи є надзвичайно важливими у сучасній розробці програмного забезпечення, як у великих командних проектах, де одночасно працюють багато розробників, так і у індивідуальній розробці. Використання систем контролю версій дозволяє мінімізувати конфлікти, такі як втрата даних, зберігати історію змін та підвищувати ефективність розробки.

Метою застосування систем контролю версій є забезпечення стабільного та безпечного середовища для збереження проекту, управління змінами та ефективної співпраці між розробниками.

Основна частина

Perforce (Helix Core) – це високопродуктивна система контролю версій, призначена для управління вихідним кодом та цифровими активами [2]. Вона пропонує централізовану архітектуру з акцентом на швидкість обробки великих файлів, безпеку та масштабованість. Perforce часто використовується у великих проектах, де потрібно зберігати та контролювати велику кількість файлів одночасно. Це робить його популярним вибором для команд, що працюють з великими ігровими рушіями, такими як Unreal Engine.

Perforce є офіційно підтримуваною системою контролю версій для Unreal Engine. Інтеграція з рушієм дозволяє розробникам зберігати як вихідний код, так і великі мультимедійні активи (текстури, моделі, звуки тощо).

Переваги використання Perforce з Unreal Engine:

1. Централізоване сховище дає змогу всій команді працювати з єдиною версією проекту, спрощуючи управління активами та запобігаючи конфліктам змін.
2. Масштабованість – це підтримка великих файлів без втрати продуктивності, що є критичним для проектів з високою якістю графіки та багатими медіаресурсами.
3. Кращий контроль доступу можливість детального налаштування прав доступу до окремих файлів або папок.
4. Підтримка функцій Unreal Engine, як-от автоматичне блокування файлів (file locking), що захищає від конфліктів при редагуванні однакових ресурсів різними розробниками.
5. Централізоване зберігання забезпечує захист від втрати даних і надає можливість швидкого відновлення проекту в разі виникнення конфліктів.

Git – розподілена система контролю версій, яка широко використовується для управління вихідним кодом. Проте, при роботі з Unreal Engine виникають певні проблеми:

1. Git не оптимізований для зберігання великих мультимедійних файлів, що призводить до проблем із продуктивністю.
 2. Не завжди зручна для централізованих проєктів з великими командами, оскільки вимагає частого злиття змін (merge) та вирішення конфліктів редагування одного файлу декількома розробниками.
 3. Інтеграція з Git вимагає додаткових налаштувань та компромісів у продуктивності.
 4. Відсутність блокування файлів (file locking) – Git дозволяє одночасно редагувати один файл декількома користувачами, що може спричинити конфлікти у великих проєктах, де потрібне синхронне редагування ресурсів.
 5. Застарілий плагін у середині графічного двигуна Unreal Engine. Плагін від Git все ще знаходиться у бета версії, що сигналізує про нестабільну роботу та потенційні конфлікти у подальшій роботі над проєктом.
- Зважаючи на централізовану архітектуру Perforce, він є менш гнучким для індивідуальної розробки у порівнянні з Git. Для особистих проєктів або невеликих команд Git залишається популярним вибором через його простоту використання, швидке. Однак, якщо проєкт передбачає роботу з великими файлами або потребує інтеграції з Unreal Engine, Perforce може бути корисним навіть у однокористувацькій діяльності [4].

Висновок

На основі проведеного дослідження можна зробити висновок, що Perforce є кращим вибором, тому, що він забезпечує централізоване сховище, можливість блокування файлів та високу масштабованість. Завдяки цим перевагам Perforce дозволяє зменшити ризик конфліктів при спільній роботі та забезпечує швидкий доступ до даних без втрати продуктивності. У порівнянні з Git, який не оптимізований для великих файлів та потребує частого злиття змін, Perforce демонструє значно кращі результати в умовах розробки з Unreal Engine. Тому, вибір Perforce для створення програмного забезпечення з використанням Unreal Engine є доцільним рішенням, що підвищує продуктивність та надійність проєкту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. В. О. Кузьмич, О. В. Коваль, Р. А. Тараненко. Управління версіями програмних засобів проєкту. КПІ ім. Ігоря Сікорського. Київ. 2023. – 9 с.
2. Perforce P4 (Helix Core) URL: <https://www.perforce.com/products/helix-core> (data of access 18.03.2025).
3. Git URL: <https://git-scm.com> (data of access 18.03.2025).
4. Perforce vs Git URL: <https://www.perforce.com/blog/vcs/git-vs-perforce-how-choose-and-when-use-both> (data of access 18.03.2025).
5. Perforce vs Git Understand the Core Differences URL: <https://www.theknowledgeacademy.com/blog/perforce-vs-git> (data of access 18.03.2025).

Колодій Владислав Віталійович – студент групи 1ПІ-24м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: vladkolodiy2014@gmail.com

Науковий керівник: **Ліщинська Людмила Броніславівна** – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Vlad Kolodii – student of group 1PI-24m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vladkolodiy2014@gmail.com

Supervisor: **Lishchynska Lyudmyla Bronislavivna** – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua