

для QtCreator, CodeBlock або Visual Studio). Цю проблему можна вирішити використовуючи ще одну професійну програмну технологію: автоматизовані IDE-незалежні збирачі (наприклад, CMake).

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Для вдосконалення навчального процесу при навчанні необхідно застосовувати сучасні способи передачі інформації, такі як безкоштовні веб-сервіси для хостингу проектів, наприклад, GitHub [3].

При вивченні нових технологій необхідно демонструвати механізм їх застосування і результат. Викладач може розмістити навчальні приклади на веб-сервісі, а потім у час навчання учні зможуть завантажити їх на комп'ютери і вивчати разом з викладачем замість простого списування з дошки. Перевага таких веб-сервісів в тому, що вони дозволяють налаштувати синхронізацію з комп'ютером таким чином, що зміни, внесені викладачем в програмний код проекту, завантажаться на сервер відразу після збереження, а потім на комп'ютери учнів.

Обмежувальним фактором виступає тільки швидкість з'єднання з глобальною мережею інтернет. У ході перевірки програм, зданих учнями в якості контрольних робіт, викладачеві необхідно визначити, чи були програми написані самостійно або списані повністю або частково. Для цієї мети існують програми в порівнянні файлів, такі як WinMerge [4]. Ці програми порівнюють вміст файлів і папок проектів, а потім візуально відображають відмінності і однакові частини. Використовуючи подібні інструменти, викладач може значно скоротити час на перевірку, збільшивши загальну ефективність. Таким чином можна з високою точністю встановити сумлінність роботи учнів.

ПОСИЛАННЯ

- [1] V.H.Hrytsenko, O.M.Podolyan, *Vykorystannya systemy upravlinnya versiyamy Git dlya orhanizatsiyi komandnoyi roboty nad IT proektom*. Informatsiyні tekhnolohiyi i zasoby navchannya, 2014. T.39. № 1. P.250-262.
- [2] *A short history of Git* : <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>.
- [3] A.V.Shaltunovich, *Organizatsiya sovmestnoy razrabotki veb-prilozheniy v ramkakh sotsial'noy seti github*. Vestnik Nizhnevartovskogo gos.universiteta, 2011. № 3.
- [4] *WinMerge* : <http://winmerge.org/about/?lang=ru>.

Олександра Пархоменко

Аспірантка

Національний університет біоресурсів і природокористування України, м. Київ, Україна

ORCID: 0000-0001-8773-0185

oleksa.parhomenko@gmail.com

ВИКОРИСТАННЯ GITHUB ДЛЯ ФОРМУВАННЯ ПРОФЕСІЙНИХ ТА ОСОБИСТІСНИХ КОМПЕТЕНТНОСТЕЙ У МАЙБУТНІХ ПРОГРАМІСТІВ

Анотація. У статті розглянуто особливості хмарних сервісів та сервісу GitHub, його можливості для застосування у навчанні програмуванню та формуванні особистісних компетентностей у майбутніх програмістів. Виокремлено можливості, переваги та недоліки використання GitHub в навчальних проектах.

Ключові слова: хмарні сервіси, GitHub, особистісні компетентності.

1. ВСТУП

Постановка проблеми. Розвиток і поширення інформаційно-комунікаційних технологій (ІКТ) надає все нові можливості покращення процесу навчання і вирішення найрізноманітніших освітніх завдань. Все більше інформаційних сервісів надається

через мережу Інтернет. Проте в навчанні програмуванню в ЗВО використовується не так багато з них. Проблема полягає як в поширенні знань про можливості веб сервісів, так і в доборі кращих з них, що задовольняли б вимогам сучасної освіти і компетентнісного підходу. Сучасна розробка програмного забезпечення ведеться із застосуванням систем контролю версій, зокрема майже всі сучасні ІТ компанії обираються для цих цілей Git, що зумовлює необхідність використання цього сервісу і в навчальних проектах.

Мета публікації. Метою статті є визначення особливостей і огляд сучасних хмарних сервісів що застосовуються під час розробки програмних продуктів, порівняння та з'ясування можливостей, переваг і недоліків їх використання для планування, організації і супроводу навчальних проектів в сфері підготовки майбутніх програмістів, зокрема використання сервісу GitHub з метою розвитку особистісних і професійних компетенцій майбутніх програмістів.

2. ТЕОРЕТИЧНІ ОСНОВИ

У загальноєвропейських рекомендаціях з освіти зазначено, що пріоритетною освітньою метою є підготовка студентів до життя в демократичному суспільстві шляхом запровадження компетентнісного підходу та методів викладання навчальних дисциплін, які стимулюють незалежність думки, судження, спонукають до відповідальної компетентної діяльності. У зв'язку з цим, в навчальному процесі на всіх рівнях освіти все більше застосовуються проектні технології під час вивчення навчальних дисциплін та окремо на міжпредметній основі. Крім того, активно впроваджуються соціальні проекти, проводяться спортивні, ігрові, культурно-масові заходи, які реалізують як виховні, так і навчальні цілі.[1] Як зазначає В. Круглик, здебільшого ІТ-випускники й роботодавці оцінюють загальний рівень професійної підготовки інженерів-програмістів в Україні як середній або вищий за середній. Роботодавці також наголошують на дефіциті високопрофесійних програмістів. [2] З огляду на це необхідним є вдосконалення змісту і форм навчання майбутніх програмістів.

3. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Широкий спектр хмарних технологій розширює можливості роботи для викладачів та студентів. Існує велика кількість найрізноманітніших сервісів, що можна використати в освітньому процесі (сервіси зберігання, поширення і спільної роботи з даними, соціальні мережі), специфічних програмних веб сервісів (онлайн доступ до графічних редакторів, відео-редакторів, створення інтелектуальних карт), та ряд особливих сервісів спеціально створених для освітніх цілей (електронні навчальні середовища, платформи масових онлайн курсів, навчальні онлайн-ігри). Веб сервіси можуть бути використані на всіх етапах навчального процесу: від створення навчальних планів до мотивації, контролю успішності, і організації зворотного зв'язку.

Для проектів в сфері ІТ особливу увагу слід приділити таким сервісам, як системи контролю версій [3]. До них відносяться централізовані системи контролю версій (Subversion, Concurrent Versions System, Team Foundation Server) та розподілені системи контролю версії (Git, Mercurial, Bazaar, Monotone, Codeville).

Найрозповсюдженою в сучасній розробці є система Git — розподілена система керування версіями файлів та спільної роботи над ними, без якої наразі не обходиться переважна більшість ІТ розробок в усьому світі.[4] Наразі це найзручніший інструмент спільної та індивідуальної роботи з програмним кодом, який до того ж використовують для спільної роботи над документами. Аналогів Git з її системою збереження версій файлів зліпками, а не послідовностями змін, наразі не існує.

Репозиторій Git призначений для зберігання текстової інформації та історії її змін. Git дозволяє: зберігати програмний код, запам'ятовувати історію змін до коду (та

дозволяти у будь-який момент побачити хто саме зробив зміни, коли зробив зміни); повернутись до будь-якої версії коду у будь-який момент; автоматично об'єднувати зміни різних версій, станів та розробників; розробляти проект командою розробників, одночасно працюючи над одними і тими ж модулями і навіть строками коду; готувати код до релізів на продакшн системи; вести статистику змін до коду; ділитись своїм програмним кодом зі спеціалістами з усього світу. Для роботи з Git достатньо командного рядка і безкоштовної реєстрації в системі. Проте існують графічні оболонки які спрощують, візуалізують роботу з кодом та надають більше можливостей для керування проектами та спілкування. GitHub — один з найбільших веб-сервісів для спільної розробки програмного забезпечення, що базується на системі керування версіями Git. Крім розміщення коду тут, учасники можуть спілкуватись, коментувати редагування один одного, а також слідкувати за новинами знайомих. Для проектів є особисті сторінки, невеликі Вікі та система відстеження помилок. Прямо на сайті можна дивитись файли проектів з підсвічуванням синтаксису для більшості мов програмування, що робить GitHub зручним не тільки для зберігання коду, а й безпосередньо для його написання.

Для роботи з Git необхідно розібратись з основними поняттями, та принципами роботи репозиторію. Для початку роботи треба авторизувати користувача, та підключитись до конкретного репозиторію (в GitHub у кожного користувача одразу є персональний репозиторій, і робота проходить в ньому). Текст або програмний код зберігається в конкретному репозиторії у вигляді файлів, кожна його зміна підтверджується командою commit, в результаті чого з'являється новий зліпок (комміт) стану системи. Система завжди зберігає всі комміти, і між ними можна переходити, відновлюючи таким чином потрібну версію коду. Послідовність коммітів в Git зберігається як гілка (branch). При цьому в будь-який момент можна створити нову гілку у вигляді розгалуження від конкретного комміту. Нова гілка коммітів може бути в будь-який момент злита з основною, від якої вона була розгалужена. Таким чином декілька розробників можуть працювати над одним файлом з кодом. Якщо кожен з розробників змінював код в різних місцях, то в результаті злиття код в основній гілці автоматично буде оновлено з урахуванням усіх змін, що їх вносили розробники. Проте, якщо при злитті можливі конфлікти версій, в такому випадку необхідно буде втручання людини, яка буде відповідальна за рішення які зміни лишити. Якщо команда працює злагоджено і кожен відповідальний за свій фрагмент програми, то проблем з версіями не буде. Таким чином робота з Git вимагає командної злагодженості і сприяє відповідальності перед іншими членами команди, при цьому полегшуючи збирання фрагментів коду до купи та забезпечуючи надійність.

GitHub має всі ознаки соціальної мережі: тут кожен користувач має власний профіль і наповнює його контентом на свій розсуд, може підписуватись на оновлення інших користувачів, коментувати їх розробки, та займатись спільною роботою над проектами. Відкритість кодує робить його доступним для будь-кого, забезпечуючи зворотній зв'язок зі спільнотою розробників. Таким чином можна як слідкувати за розробками провідних програмних продуктів з відкритим кодом, вивчати як вони створені, так і отримувати допомогу інших користувачів. Особливої мотивації додає об'єднання незнайомих розробників з усього світу в роботі над відкритими проектами, де кожен працює добровільно, з власної цікавості і має можливість набути досвіду спільної розробки за межами навчальних закладів.

Отже, використання хмарного сервісу контролю версій GitHub у навчанні програмуванню дозволяє отримувати електронний інформаційний супровід проекту, спрощує спільну роботу з інформацією, надає можливість онлайн комунікації, поширення результатів проекту на широкий загаль, спрощує оформлення і планування

діяльності, підвищує мотивацію учасників. До недоліків можна віднести два основні моменти: необхідність доступу до мережі Інтернет і низька конфіденційність розміщеної в мережі інформації при безкоштовній підписці. Необхідність комп'ютера або іншого мобільного пристрою з можливістю доступу до мережі Інтернет наразі не є недоліком, адже майже вся студентська і шкільна аудиторія забезпечена такими пристроями, що є нормою в сучасному світі.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Сучасне програмування є колективним, і корисність окремого програміста тісно пов'язана з його корисністю для всієї команди, тому важливо забезпечити відповідну підготовку майбутніх програмістів. Суттєвим є внесок у формування компетентностей інженерів-програмістів, що можна віднести до особистісних, зокрема: соціально-поведінкові, комунікативні, загальнонаукові, управлінські та дослідницькі. Всі ці аспекти проявляються в командній роботі, для якої в навчальній діяльності варто використовувати системи контролю версій та спільної роботи над кодом, зокрема найбільш розповсюджену систему Git з інтерфейсом GitHub, що надає можливість не тільки спільної роботи над кодом, а й соціалізації розробки проектів, обміну досвідом з розробниками з усього світу. Подальшого розгляду потребує імплементація сучасних програмних засобів для командної розробки в навчальний процес ЗВО.

ПОСИЛАННЯ

- [1] О. Зосименко. «Періодизація становлення та розвитку проектного навчання», Педагогічні науки : зб. наук. праць СумДПУ ім. А.С. Макаренка, ч.2, с. 238-245, 2006.
- [2] В. Круглик, «Аналіз сучасного стану професійної підготовки інженерів-програмістів в Україні», Науковий вісник Мелітопольського державного педагогічного університету, Серія: педагогіка, № 1 (18), с.138-145, 2017.
- [3] M. Cochez, V. Isomottonen, V. Tirronen and J. Itkonen, «The Use of Distributed Version Control Systems in Advanced Programming Courses», ICTERI, pp.221-235, 2013
- [4] А. Симончук, «GitHub-аккаунты украинских ITшников: зачем нужны, как раскручивали и советы новичкам», 2017. [Електронний ресурс] Режим доступу: <https://dou.ua/lenta/articles/quiz-github/> Дата звернення: 28.04.2019.

Ірина Якимчук

Викладач, завідувач відділення програмування

Відокремлений підрозділ «Рівненський коледж НУБіП України», м. Рівне, Україна

irina.o.yakymchuk@gmail.com

МЕДІАКОМПЕТЕНТНІСТЬ ВИКЛАДАЧІВ І ЗДОБУВАЧІВ ОСВІТИ ЯК ОБОВ'ЯЗКОВА СКЛАДОВА ОСВІТНЬОГО ПРОЦЕСУ В КОЛЕДЖІ

Анотація. Широкий простір Інтернету дозволяє сучасній людині отримувати необхідну інформацію не виходячи з дому. Але обмежувати поповнення інформаційного запасу тільки засобами Інтернет-простору не раціонально. Для розширення такого запасу інформаційне суспільство використовує книги, періодику, телебачення, пошту, телефон тощо – медіа (книги, преса, радіо, кіно, телебачення, Інтернет).

На взаємодію з різноманітними медіа припадає все вагоміша частка у бюджеті вільного часу громадян України, чим зумовлюється значний вплив медіа на всі верстви населення, передусім на дітей і молодь.

Вимоги реформування освіти потребують від майбутніх фахівців правильної організації власного інформаційного простору шляхом накопичення знань, навичок та вмінь, які дають