

оновлення продукту - додавання функціоналу або виправлення помилок відразу для всіх платформ; 4) мобільна версія - більшість кроссплатформених рішень дозволяють генерувати мобільну версію сайту з програми; 5) єдина логіка програми - логіка додатка однаково працюватиме для всіх платформ. Означені переваги істотно впливають на поліпшення праці й швидкості її виконання.

Водночас, до особливостей використання вільних кроссплатформових середовищ програмування слід віднести: 1) при використанні кроссплатформової розробки використовуються спеціальні інструменти, які дозволяють створювати додатки відразу для декількох мобільних операційних систем; 2) більшість пакетних рішень забезпечують багатомовну підтримку; 3) можливість розширення стандартних інструментів за рахунок встановлення розширень; 4) наявність статичних і динамічних бібліотек; 6) автоматичне створення звітів про помилки.

Отже, з результатів цієї науково-пошукової роботи випливає, що вільні кроссплатформові середовища програмування володіють значними перевагами, які полегшують виконання ІТ-проектів, сприяють більш швидкій реалізації ПЗ. Разом з тим мають місце особливості, які теж сприяють на полегшення праці під час розроблення програмних засобів.

ДЖЕРЕЛА

1. Поняття про середовища програмування [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://studopedia.eu/16_3881_ponyattya-pro-seredovishcha-programuvannya.html.

2. Нативная или кроссплатформенная разработка [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <http://wnfx.ru/nativnaya-ili-krossplatformennaya-razrabotka-chto-luchshe/>.

МЕТОДИЧНІ ЗАСАДИ РЕАЛІЗАЦІЇ ПРОЦЕСУ КОМПЛЕКСУВАННЯ ПІД ЧАС МОДУЛЬНОГО ПРОГРАМУВАННЯ

Ткаченко О.С.

(науковий керівник – к.т.н. Доренський О.П.)

Центральноукраїнський національний технічний університет, м. Кропивницький

Важливу роль у реалізації програмних засобів, особливо під час модульного програмування, відіграє процес комплексування (інтегрування). Він часто поєднується або є субпроцесом конструювання ПЗ, тобто детального створення програмного забезпечення за допомогою комбінації кодування, верифікації, модульного й інтеграційного тестування і відлагодження. Конструювання пов'язане з усіма іншими дисциплінами програмної інженерії, найбільш сильно – з проектуванням і

кваліфікованим тестуванням ПЗ. Тож метою цієї праці є аналіз методичних засад реалізації комплексування ПЗ під час модульного програмування.

У SWEBOOK процес конструювання окремо не розглядається, тому однією з ключових діяльностей, що здійснюються в процесі конструювання, є інтеграція окремо сконструйованих операцій (процедур), класів, компонентів і підсистем (модулів). Крім цього до інтеграційних завдань конструювання відносять такі: планування послідовності, в якій інтегруються компоненти; забезпечення підтримки створення проміжних версій програмного забезпечення; завдання "глибини" тестування (зокрема, на основі критеріїв "прийнятності" якості) та інших робіт щодо забезпечення якості інтегруються в подальшому компонент; визначення етапних точок проекту, коли будуть тестуватися проміжні версії конструюється програмної системи.

Методика Unified Process пропонує фазовий підхід до реалізації ПЗ, у ній комплексування є фазою конструювання, яка містить ітеративну розробку системи, що може успішно взаємодіяти з користувачами на бета-оточені. Кожна з фаз складається із ітерацій, які є мініатюрними субпроектами обмеженої тривалості. Результатом ітерації є реліз, що має поліпшення порівняно з попередньою версією ПЗ. В ході роботи над ітерацією команда ІТ-проекту виконує: усунення критичних ризиків перед початком робіт, створення плану ітерації з потрібним ступенем деталізації, плановані роботи, аналіз отриманого релізу, оновлення списку ризиків, оновлення плану проекту відповідно до результатів ітерації [3].

Методологія MSF поєднує в собі ітераційний, фазовий і інтегрований підходи. У ній процес конструювання з'єднаний з процесом конструювання, тобто є його певним ітераційним підпроцесом. Загалом MSF допускає, що мінімальний колектив може складатися з трьох осіб, тобто проект може бути створеним якісно, але при мінімальних затратах на кожний крок процесу розробки ПЗ і на процес інтегрування окремо [4].

Стандартом, у якому був відокремлений процес інтегрування, був ISO/IEC 12207:2008. Процес комплексування ньому є процесом більш низького рівня, ніж процес реалізації ПЗ. Його мета полягає в об'єднанні програмних блоків і компонентів, створення інтегрованих програмних елементів, узгоджених з проектом ПЗ, які демонструють, що функціональні і нефункціональні вимоги до ПЗ задовольняються на повністю укомплектованій або еквівалентній їй операційній платформі.

Разом з тим, до найпопулярніших сервісів для виконання процесу інтегрування ПЗ можна віднести такі: GitLab, який є платформою управління Git-репозиторіями, аналізу коду, відстеження помилок, тестування, деплоя, ведення каналів і вікі-сторінок; GitLab допомагає розробникам вести безперервний процес розгортання для тестування, створення і деплоя коду, стежити за ходом тестів, підвищувати контроль над якістю, фокусування на побудові продукту замість настроювання

інструментів; GitHub – “соціальна мережа для розробників”, учасникам, крім безпосереднього зберігання коду, своїх проектів, можна спілкуватися, коментувати зміни один одного, відстежувати новини знайомих, реалізовано можливість об’єднувати репозиторії та виводити внесок учасника у вигляді дерева; Phabricator є набором веб-інструментів для спільної розробки ПЗ, що включає засоби перевірки коду, сховища, моніторинг змін, трекер ошібок і вики; Beanstalk – сервіс для повноцінного написання, перегляду і розгортання коду, який допомагає контролювати версії, імпортувати або створювати репозиторії Subversion і Git, дозволяє налаштувати обмеження доступу до певних репозиторіїв, видати дозволи тільки для читання або для повного доступу.

Отже, під час науково-пошукової роботи отримано результати, які у сукупності можуть використовуватися для обґрунтування вибору методики виконання ІТ-проекту в частині реалізації комплексування ПЗ.

ДЖЕРЕЛА

1. ISO/IEC 12207 : 2008. Systems and software engineering – Software life cycle processes. – ISO/IEC-IEEE, 2008. – 122 p. – (International Standard)
2. The Guide to the Software Engineering Body of Knowledge (SWEBOOK Guide) – SWEBOOK Pierre Bourque, Robert Dupuis; executive editors, Alain Abran, James W. Moore, eds., 2004.
3. Kendall S. The Unified Process Explained / S. Kendall. –2002. – 185 p.
4. Майкл, Тернер Основы Microsoft Solution Framework / Тернер Майкл. – М.: Питер, 2013. – 550 с.
5. Dorenskyi, O.P. The Methodology of Evaluating the Test Cases Quality for Simple IT Monoprojects Software Testing / O.P. Dorenskyi // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій : тези доп. VIII Міжнар. наук.-практ. конф. (21-23 вер. 2016 р., м. Запоріжжя). – Запоріжжя : ЗНТУ, 2016. – С. 111-112.

AR-ТЕХНОЛОГІЇ В ОСВІТНЬОМУ ПРОЦЕСІ

Чернецька Ю. М.

Тернопільський національний педагогічний університет ім. В.Гнатюка, м. Тернопіль

Сучасна молодь не може уявити життя без додаткових гаджетів. За статистикою четверо з п’яти студентів володіють смартфоном. Найбільший відсоток студентів використовують смартфон для соціальних мереж, трохи менше, щоб грати в ігри, але, найменший відсоток студентів використовують смартфон в навчальних цілях. Сьогодні традиційне навчання для студентів вже не цікаве, тому, основне завдання сучасного викладача активізувати у студентів цікавість до навчальної діяльності.