

УДК 004.415.2

ПРОЄКТУВАННЯ ЯК НЕОБХІДНИЙ ЕТАП ЖИТТЄВОГО ЦИКЛУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Курлянський С. С.¹

e-mail: ss645wot@gmail.com

Шаров С. В.², к.пед.н.

e-mail: sergii.sharov@tsatu.edu.ua

¹КЗ ЗСО «Червоногригорівська загальноосвітня школа I-III ступенів Червоногригорівської селищної ради»

²Таврійський державний агротехнологічний університет імені Дмитра Моторного

Актуальність та постановка проблеми. На сьогодні інформаційно-комунікаційні технології глибинно-інтегровані у багатьох сферах діяльності людини. Чималу частину цих технологій складає програмне забезпечення (ПЗ), зокрема системні та прикладні програмні продукти. Обов'язковою умовою розробки якісного програмного продукту є дотримання етапів життєвого циклу. Одним із початкових етапів є процес проєктування програмного забезпечення.

Основні матеріали дослідження. Відомо, що процес розвитку та модифікації методів розробки прикладного програмного забезпечення (призначене для розв'язання конкретних функціональних задач) призвів до появи чималої кількості відповідних методологій та технологій. Під технологією розробки програмного забезпечення розуміється система інженерних принципів для розробки програмного забезпечення, яка ефективно та надійно працює на реальних комп'ютерах [6, с. 20].

Процес розробки програмного забезпечення містить дії та завдання розробника зі створення програмного продукту. Його важко уявити без розподілу цього процесу на певні етапи, які зазвичай називаються етапами життєвого циклу. Життєвий цикл зазвичай має ітераційний характер та утворюється у відповідності до принципу низхідного проєктування. Кожний етап життєвого циклу має певний перелік технічних рішень та документів, які, в свою чергу, є вихідними даними для документів і рішень наступного етапу життєвого циклу. Кожний етап з метою перевірки відповідності вихідним документам та рішенням завершується верифікацією породжених документів і рішень [2, с. 37].

Можна виділити такі етапи життєвого циклу: аналіз предметної області, на основі чого створюється технічне завдання; проєктування структури майбутнього програмного засобу; безпосередня розробка модулів програмного засобу згідно з проєктною документацією; тестування програмного засобу та його налагодження; впровадження комп'ютерної програми у виробництво; супровід програми під час її роботи в реальних умовах; переробка програмного засобу в інший програмний продукт або утилізація у випадку моральної застарілості [8].

Проєктування ПЗ передбачає процес створення специфікацій на основі переліку вимог, які формуються перед початком проєктування. Цей процес передбачає розробку структури програмного забезпечення, визначення взаємозв'язків його окремих структурних елементів та містить два основних етапи.

Етап розробки структури та інтерфейсів компонентів, узгодження функцій і технічних вимог, що висуваються до стандартів проєктування, компонентів, методів тощо. Цей етап називається проєктуванням архітектури системи.

Метою детального проєктування є розробка інтерфейсів між компонентами, специфікацій кожного компонента, вимог до тестів та аналогічні питання [2, с. 41].

Деякі дослідники вважають, що проєкт розробки програмного забезпечення слід будувати за модульною схемою, де власне програмна система розбивається на модулі – підсистеми. Водночас, підсистеми виділяються за принципами подібності технологій та функціональної спільності. Такий принцип організації проєктування надає можливості одночасного створення проєкту спеціалістами різних профілів. Це, в свою чергу, впливає на забезпечення можливості заміни окремих модулів у разі потреби, наприклад, з метою вдосконалення функціонала програмних систем. Крім того, у проєкті бажано передбачити поетапне впровадження підсистем, можливість їх масштабування та розвитку [9].

Зазвичай процес проєктування ПЗ передбачає розробку певних моделей, що відображають структуру програмної системи, функціональні можливості її складових частин, їх взаємодію з іншими компонентами, у тому числі зовнішніми. У результаті повинна бути побудована певна модель, яка демонструє, яким чином програмний засіб буде задовольняти сформованим на початку проєктування вимогам. Фактично буде побудована модель реалізації, яка є уточненням моделі вимог. Тобто проєктування виступає у якості зв'язку між аналізом і реалізацією вимог, що висуваються до майбутньої програмної системи [2, с. 41].

Виділяють два основних принципи розробки комп'ютерних програм: модульний і об'єктно-орієнтований. Перший принцип використовує структурні методи проєктування, які передбачають розбиття програмного засобу на структурні компоненти за деякими правилами. Об'єктно-орієнтована розробка використовує об'єктні методи, до яких відносяться методології об'єктно-орієнтованого проєктування, аналізу та програмування [3, с. 11]. В основі цього підходу лежить об'єктна декомпозиція, тобто ПЗ представляється у вигляді сукупності об'єктів, у процесі взаємодії яких відбувається виконання необхідних функцій. Слід зазначити, що об'єктно-орієнтоване програмування та проєктування є основною парадигмою розробки складних систем [5, с. 200], у тому числі програмних.

Моделі програмного забезпечення, які розробляються при об'єктному підході, засновані на предметах реального світу. В основі зазначених моделей лежить опис необхідної поведінки майбутнього програмного забезпечення, яка пов'язується із станами об'єктів конкретної предметної області. Тобто, на етапі аналізу потрібно уточнити необхідну поведінку майбутнього програмного забезпечення, на основі чого розробити концептуальну модель його предметної області з точки зору поставлених завдань.

Розробка програмного забезпечення за допомогою об'єктного підходу, як правило, доволі складна. Тому для опису розробки часто використовують інтуїтивно-зрозумілу мову модулювання UML. З її допомогою можна візуалізувати моделі, описувати параметри, конструювати та документувати різні системи, у тому числі інформаційні. Графічне представлення моделі за допомогою мови UML отримало назву діаграми (diagram), які можна представити у вигляді спеціальних графічних конструкцій, які описують різні стани системи. Важливою складовою мови UML є графічна нотація (сукупність символів і правил їх застосування [10, с. 279]) за допомогою якої відбувається процес проєктування ПЗ через візуалізацію об'єктів проєктованої системи, функціональних можливостей цих об'єктів та їх поведінки. Для створення діаграм використовують чотири види графічних елементів: фігури, лінії, написи, позначення.

Зазвичай для побудови моделі майбутнього програмного засобу використовуються діаграми прецедентів, які визначають варіанти використання програмного засобу різними системами або підсистемами без розгляду внутрішньої структури [4, с. 17]. Крім того, обов'язковою діаграмою є діаграма класів, яка визначає внутрішній склад програмного засобу. Для створення UML-діаграм використовуються різноманітні Case-засоби, серед яких слід назвати Microsoft Visio [7, с. 68] Rational Rose; Sparx Systems Enterprise Architect. Вони володіють корисною функцією реверсивного інжиніринга та здатні генерувати програмний код у різних об'єктно-орієнтованих мовах.

Етап проєктування програмного забезпечення багато в чому визначає функціональність майбутнього програмного засобу, тому власне проєктуванням повинні займатися фахівці. Навіть було введено поняття «дефект проєктування», що охоплює проблеми різного рівня деталізації. Дефекти проєктування можуть бути помічені на рівні програмної реалізації у результаті помилок та колізій, які були допущені на фазі проєктування [5, с. 201].

Слід зазначити, що фахові компетентності з проєктування програмного забезпечення зазвичай формуються у вищому навчальному закладі. Водночас, впровадження інформаційно-комунікаційних технологій у навчальний процес позитивно впливає на якість вищої та середньої освіти. Комп'ютерна техніка може бути використана з різною метою, від звичайного довідника та електронного журналу до управління навчальним процесом у якості потужної інтелектуальної системи. Одним із сучасних видів навчально-методичного забезпечення є програмно-педагогічні засоби, які забезпечують вивчення навчального матеріалу, формування та розвиток окремих вмій тощо.

Окремі науковці відзначають недостатню кількість, а іноді відсутність навчальних комп'ютерних програм, які урізноманітнюють навчальний процес та підвищують його ефективність. Крім того, розробниками таких прикладних програмних продуктів часто є програмісти, які не обізнані у специфіці навчального-виховного процесу, не володіють основами дидактики, не мають досвіду педагогічної діяльності, не можуть врахувати психологічні та фізіологічні особливості студентів. Зважаючи на це, нами було розроблено програмно-педагогічний засіб з дисципліни «Проєктування програмного забезпечення», який містить навчальну інформацію про технології та моделі проєктування, дозволяє здійснити самоконтроль та контроль знань та умінь за допомогою комп'ютерного тестування та електронних навчальних тренажерів [3, с. 22]. Його можна використовувати під час самостійної роботи та в аудиторії.

Висновки. Отже, проєктування програмного забезпечення є необхідним структурним компонентом життєвого циклу програмного забезпечення. Дієвим засобом є об'єктно-орієнтоване проєктування програмного забезпечення, для якого можна використати уніфіковану мову моделювання UML. Мова UML призначена для візуалізації процесу проєктування та опису функціональних та інших можливостей майбутнього програмного комплексу. Для формування фахових компетентностей з проєктування можна використовувати програмно-педагогічні засоби.

Список використаних джерел:

1. Бахтизин В. В., Глухова Л. А. Технология разработки программного обеспечения: учеб. пособие. Минск: БГУИР, 2010. 267 с
2. Гужва В. М. Інформаційні системи і технології на підприємствах: навч. посіб. К.: КНЕУ, 2001. 400 с.

3. Колмакова В. О., Шаров С. В., Курлянський С. С. Використання програмно-педагогічного засобу з дисципліни «Проектування програмного забезпечення». *Вісник Кременчуцького національного університету імені Михайла Остроградського*. 2020. Вип. 4(123). С.17–24.

4. Лубко Д. В., Зінов'єва О. Г., Шаров С. В. Проектування та розробка експертної системи діагностування несправностей транспортних засобів. *Системи обробки інформації*. Харків: Харківський університет Повітряних Сил імені Івана Кожедуба, 2019. № 1(156). С. 15–21.

5. Нечай О. С., Сидоров М. О. Методи та засоби виявлення дефектів проектування об'єктно-орієнтованого програмного забезпечення. *Вісник Національного Авіаційного Університету*. 2009. Т.40. №3. С. 200–205.

6. Орлов С. А., Цилькер Б. Я. Технологии разработки программного обеспечения. СПб.: Питер, 2012. 608 с.

7. Поморцева Е. Е. Использование Visio при визуализации бизнес-процессов. *Восточно-Европейский журнал передовых технологий*. 2011. №1(2). С. 68–70.

8. Проектирование программного обеспечения. URL: <https://habr.com/post/74330>.

9. Спірін О. М., Саух В. М., Резніченко В. А., Новицький О. В. Проектування системи електронних бібліотек наукових і навчальних закладів АПН України. 2009. Т. 6. №14. URL: <http://eprints.zu.edu.ua/3924/>.

10. Томашевський О. М., Цегелик Г. Г., Вітер М. Б., Дудук В. І. Інформаційні технології та моделювання бізнес-процесів. К.: «Видавництво «Центр учбової літератури», 2012. 296 с.