УДК 378.147
DOI: 10.31673/2412-9070.2021.054853

І. М. ГАМАНЮК, ст. викладач,
Державний університет телекомунікацій, Київ

УПРОВАДЖЕННЯ НАСКРІЗНОГО ПРОЄКТУ В ПРОЦЕС ПІДГОТОВКИ СТУДЕНТІВ У ГАЛУЗІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті досліджено застосування наскрізного проєкту в процес підготовки студентів у галузі інженерії програмного забезпечення, його позитивний вплив на підвищення якості освіти.

Ключові слова: процес підготовки; інженерія програмного забезпечення; застосування наскрізного проєкту.

Вступ

З кожним днем потреба у висококваліфікованих фахівцях з інженерії програмного забезпечення дедалі стрімко зростає. Фахівці затребувані не тільки в Україні, а і за кордоном. Проте сьогодні існує проблема щодо підготовки зазначених фахівців. Тому деякі з провідних ІТ-компаній змушені розгортати на базі своїх фірм навчальні заклади для додаткової фахової підготовки (перепідготовки) інженерів програмного забезпечення. «Програмуванню ніхто не навчає» — дуже поширений вираз представників компаній. «Наша компанія не навчальний заклад» — вислів, що застосовують у разі пояснення відмови в прийнятті на роботу не досить досвідченому фахівцю. Виникає замкнуте коло: «Не беруть на роботу, бо немає досвіду. Немає досвіду, бо не беруть на роботу».

Нині інформаційні технології змінюються настільки швидко, що вчорашній фахівець із певної технології сьогодні стає вже не фахівцем. Збільшується кількість нових технологій, які ґрунтуються на базових. Їх також складно охопити в навчальному процесі через нестачу часу та фахівців із підготовки. Самостійно розподіляючись по різних нових технологіях, студенти фокусують свою увагу тільки на певних напрямках і не бажають розпорошувати свої зусилля. Враховуючи це, навчальні заклади вимушені приділяти увагу базовим технологіям, які певною мірою знадобляться всім.

Складність матеріалу, що викладається, також вносить певні обмеження на підготовку майбутніх фахівців. Так, матеріал, що стосується створення реальних проєктів, дуже рідко трапляється і є досить складним для освоєння. Зазвичай цей матеріал можна віднести до мистецтва, а не до ремесла. А отже, багато компаній не бажають ділитися своїми напрацюваннями, тим паче безкоштовно.

Для розв'язання цих та інших проблем і було запроваджено використання наскрізних проєктів масштабу семестру чи курсу.

Під час проєкту відпрацьовуються питання щодо програмування систем, де кількість класів та конструкцій лічиться десятками та використовуються деякі патерни проєктування. Усі вони мають бути узгодженими, займати своє місце і відігравати певну роль.

Набутий досвід надає студентові відчуття впевненості у своїх силах, оскільки він можливо вперше брав участь у створенні вагомого застосунку.

Підготовлені за певними напрямками студенти привносять до проєкту цікаві вирішення, а роль викладача полягає в поширені набутого досвіду на наступний рік (семестр), зменшуючи в такий спосіб залежність від досвідчених фахівців. Матеріал відпрацьовується передусім з використанням базових технологій.

Мета дослідження: створення методичної розробки з підготовки студентів за спеціальністю 121 «Інженерія програмного забезпечення».

Завдання методичної розробки: схарактеризувати шляхи забезпечення якості проведення практичних занять та впровадження проєктних технологій у педагогічній діяльності.

Основна частина

У межах інноваційного змісту навчання за дисциплінами Об'єктно-орієнтоване програмування, Моделювання та проєктування програмного забезпечення запроваджено проєктну технологію в навчальний процес.

Нині великої популярності набув уніфікований, тобто ітеративний та поступовий підхід (процес) до розроблення програмного забезпечення.

Уніфікований процес визначає життєвий цикл створення проєкту, що складається з чотирьох фаз: початок (*inception*), розвиток (уточнення) (*elaboration*), конструювання (побудова) (*construction*) та передавання (впровадження) (*transition*).

Кожна фаза поділяється на ітерації. Результатом кожної ітерації є приріст розроблюваного проєкту у вигляді додавання нових або покращення наявних функціональних можливостей порівняно з попереднім випуском.

Під час кожної ітерації відпрацьовуються такі дисципліни проєкту: бізнес-моделювання; вимоги; проєктування; реалізація; тестування; розгортання; конфігурація і керування змінами; керування проєктом; середовище.

У межах практичних занять з кожної з навчальних дисциплін визначається окремий наскрізний проєкт, який потрібно відпрацювати.

Одне або два практичних заняття становить ітерацію з розроблення наскрізного проєкту, в рамках якої створюється нова функціональність або покращується наявна. При цьому в проєкт упроваджується нова конструкція з програмування, яку вивчають, або нова діаграма (документ), що відпрацьовується внаслідок проєктування.

Фаза початку триває одне практичне заняття. Під час цієї фази розглядається завдання зі створення наскрізного проєкту. Визначається початкове бачення проблеми, прецедентів, здійснюється оцінювання складності задачі.

У межах дисциплін проєкту відпрацьовується таке:

- бізнес-моделювання — обговорюється модель предметної галузі, яка є візуальним поданням найбільш важливих сутностей із предметної галузі і їх взаємозв'язків;
- вимоги — починається відпрацювання моделі прецедентів, що є основними функціональними вимогами, та відпрацювання основних нефункціональних вимог, які відображаються в додатковій специфікації;
- проєктування — обговорюються класи моделі;
- реалізація — створюються основні класи моделі;
- тестування — тестується реалізоване програмне забезпечення;
- розгортання — створюється акаунт на вебсервісі для спільного розроблення програмного забезпечення GitHub та розміщуються напрацьовані артефакти;
- конфігурація і керування змінами — здійснюються відповідні зміни за потреби;
- керування проєктом — визначаються студенти, які формують ядро команди чи команд;
- середовище — здійснюється ознайомлення з можливостями Microsoft Visual Studio та онлайн-середовища розробки dotnetfiddle.net, а також з можливостями онлайн-редактора UMLetino діаграм уніфікованої мови моделювання UML і вебсервісу для спільного розроблення програмного забезпечення GitHub. Відбувається ознайомлення з можливостями платформи для проведення онлайн-занять та відеоконференцій Zoom.

Фаза розвитку за часом відповідає кільком практичним заняттям. Під час цієї фази формується більш повне бачення проблеми, відбувається ітеративна реалізація базової архітектури, створюються найбільш критичні компоненти, здійснюється ідентифікація основних вимог, отримуються більш реалістичні оцінки щодо успішності проєкту.

У межах дисциплін проєкту освоюється таке:

- бізнес-моделювання — відпрацьовується модель предметної галузі;
- вимоги — відпрацьовуються модель основних прецедентів, що є основними функціональними вимогами, та основні нефункціональні вимоги, які відображаються в додатковій специфікації. Формується документ бачення, а також відпрацьовується діаграма основних варіантів використання;
- проєктування — створюється модель проєктування, яка відбиває програмні об'єкти, тобто відпрацьовуються діаграми взаємодії, діаграма класів та інші діаграми, що вивчаються;
- реалізація — створюються класи моделі (класи сутності), класи з керування та граничні класи;
- тестування — тестується реалізоване програмне забезпечення;
- розгортання — упроваджуються та вносяться зміни до проєкту на GitHub; розміщуються напрацьовані артефакти;
- конфігурація і керування змінами — здійснюються відповідні зміни за потреби;
- керування проєктом — активні кваліфіковані студенти залучаються для надання допомоги іншим;
- середовище — здійснюється розроблення проєкту (кодування) в середовищі розробки Microsoft Visual Studio або онлайн-середовищі розробки dotnetfiddle.net. Виконується розроблення проєкту (відпрацювання діаграм) в онлайн-редакторі UMLetino діаграм уніфікованої мови моделювання UML. Вносяться зміни до артефактів на вебсервісі для спільного розроблення програмного забезпечення GitHub. Здійснюється використання платформи для проведення онлайн-занять та відеоконференцій Zoom.

Фаза конструювання охоплює більшість практичних занять. Під час цієї фази здійснюється ітеративна реалізація інших критичних і нових елементів, які залишаються не реалізованими, підвищується якість програмного забезпечення завдяки реалізації нових конструкцій мови програмування та покращення функціональних можливостей.

У межах дисциплін проєкту відпрацьовується таке:

- бізнес-моделювання — оновлення (за потреби) моделі предметної галузі;
- вимоги — оновлюється модель прецедентів, додаткова специфікація та документ бачення;
- проєктування — оновлюються діаграми взаємодії та діаграма класів, а також відпрацьовуються інші діаграми, що вивчаються;
- реалізація — програмуються прецеденти;
- тестування — тестується реалізоване програмне забезпечення;
- розгортання — упроваджуються та вносяться зміни до проєкту на GitHub або на Bitbucket; розміщуються напрацьовані артефакти;

• конфігурація і керування змінами — здійснюються відповідні зміни за потреби;

• керування проєктом — активні кваліфіковані студенти залучаються до застосування нових технологій та для надання допомоги іншим;

• середовище — здійснюється розроблення проєкту (кодування) в середовищі розробки Microsoft Visual Studio або онлайн-середовищі розробки dotnetfiddle.net. Виконується розроблення проєкту (відпрацювання діаграм) в онлайн-редакторі UMLetino діаграм уніфікованої мови моделювання UML. Вносяться зміни до артефактів на вебсервісі для спільного розроблення програмного забезпечення GitHub. Використовується платформа для проведення онлайн-занять і відеоконференцій Zoom та для запису розроблення частини проєкту.

Фаза передавання (впровадження) відбувається на кінцевому практичному занятті та на іспиті, коли кожний студент має захистити свій проєкт і дістати відповідну оцінку.

Наприкінці кожної ітерації (практичного заняття) студенти отримують працюючу частину проєкту.

Під час наскрізного проєкту студенти запрошуються для підготовки та здійснення інформування інших студентів щодо тих чи тих патернів проєктування або підходів до опису бізнес-процесів.

З метою підвищення якості навчання після закінчення певних ітерацій відбувається відеофіксація розроблення опрацьованої частини проєкту.

Студентів поділяють на групи. Найбільш підготовлений студент очолює розроблення. Він повністю розуміє процес і готовий надати дієву допомогу з розроблення артефактів — тієї чи тієї діаграми (документа) або того чи того програмного модуля (рис. 1).

Студенти опрацьовують діаграму артефактів і з'ясовують, що саме їхня група буде робити, від яких артефактів залежить їхній артефакт. Це впливає на момент початку розроблення артефакту.

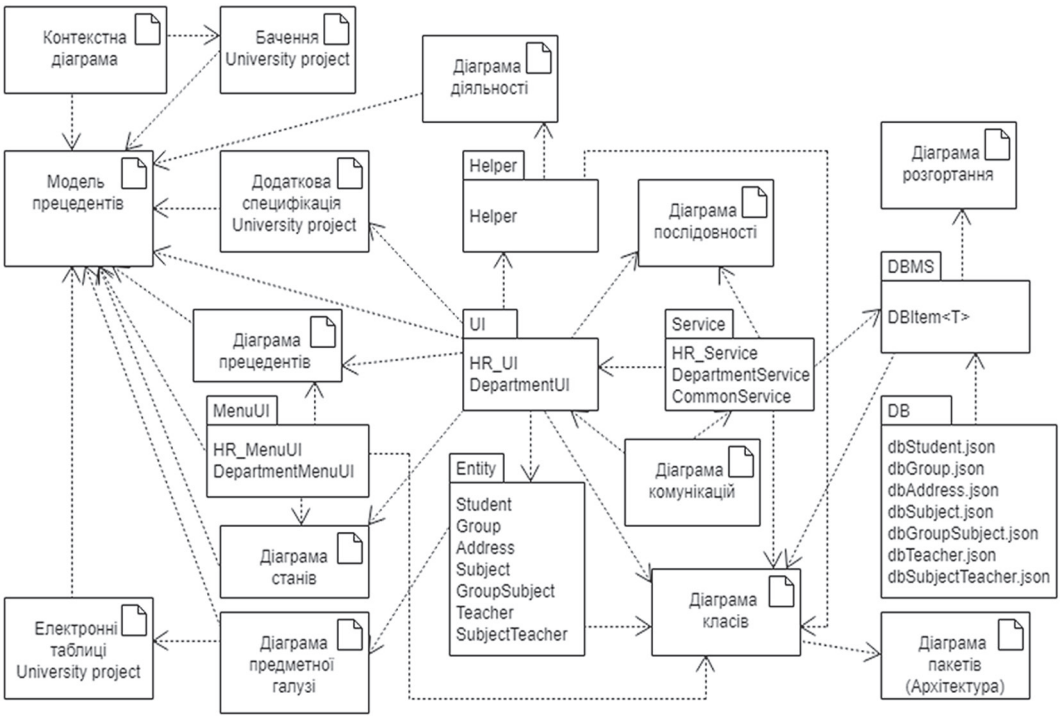

Рис. 1. Місце та роль артефакту

Артефакт пропонується до розроблення після певної готовності артефактів, на основі яких розробляється необхідний артефакт. На початкових ітераціях потрібно складати План розроблення, де зафіксовано процедуру розроблення артефактів. Але, набувши досвіду, студенти розуміють, від чого залежать їхні дії, і такий документ стає необов'язковим (тим паче, що існує діаграма артефактів, де всі залежності позначено).

Розроблення програмного забезпечення під час відеофіксації здійснюється за вертикаллю з мінімальним набором функціоналу горизонтальних рівнів. Це дає змогу нарощувати проєкт за горизонталлю (рис. 2, рис 3). У процесі нарощування на початкових ітераціях студентів бажано не замінювати, щоб вони чітко зафіксували своє місце роботи в архітектурі проєкту і відчули як варіативну частину своєї роботи, так і незмінну її частину.

Після перших ітерацій студенти починають розуміти, що нічого складного немає і в подальшому все більша кількість студентів виявляє бажання працювати безпосередньо над проєктом. Керівники груп розподіляють роботу над артефактом і забезпечують його відпрацювання, зі свого боку в такий спосіб підвищуючи загальну швидкість розроблення проєкту.

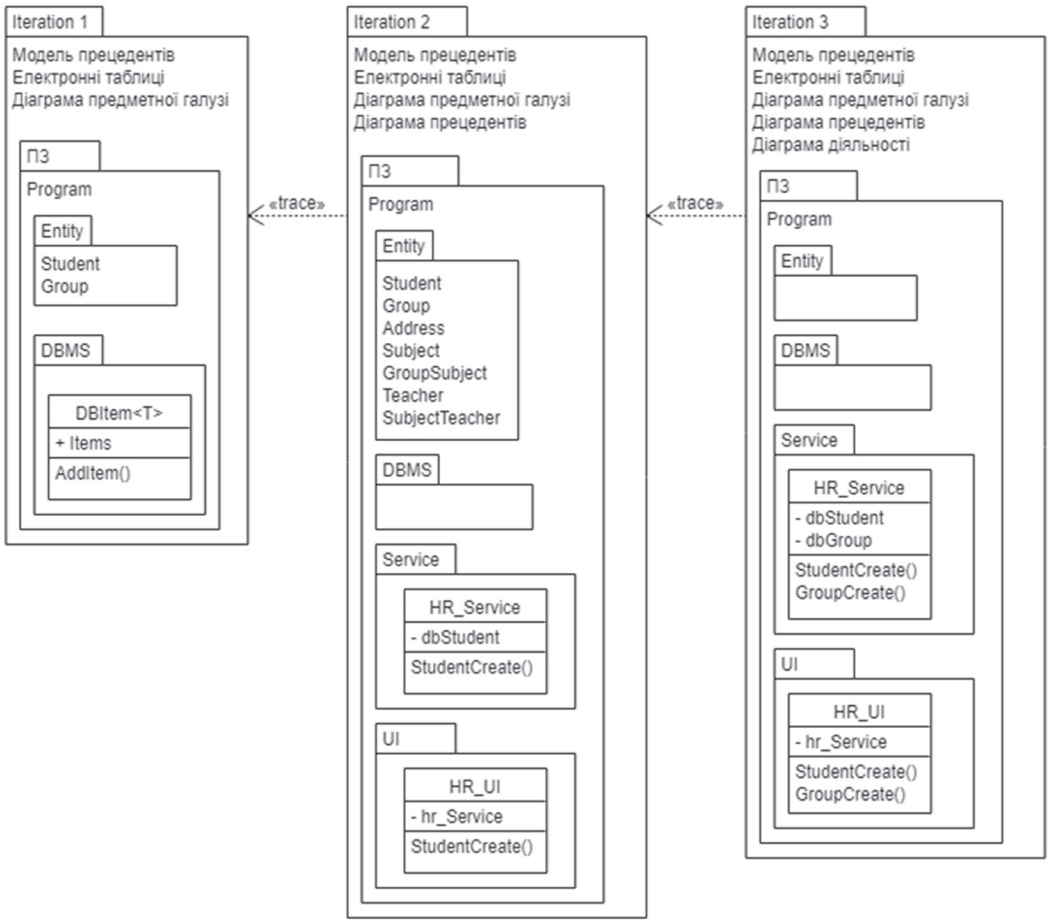

Рис. 2. Нарощування проєкту

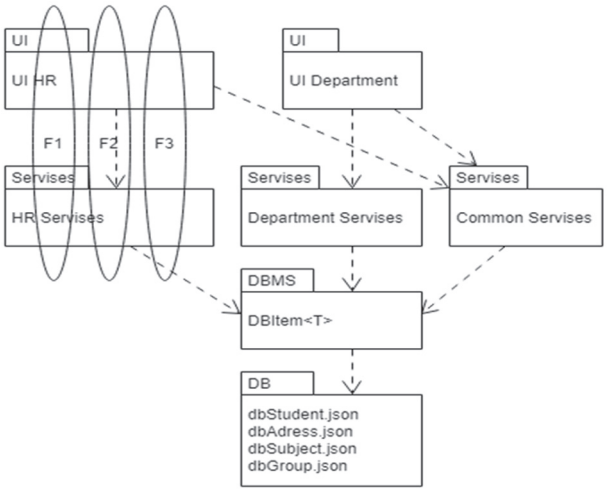

Рис. 3. Нарощування функціональності за вертикалями

На загальну швидкість розроблення проєкту також позитивно впливає передавання на узагальнення програмних артефактів (класів) за двома рівнями відпрацьованості. Перший рівень: інтерфейс та клас із позначеними методами, що будуть реалізувати інтерфейс. Другий рівень: інтерфейс, клас із методами, які реалізують інтерфейс, та частиною коду, що перевіряє працездатність класу. Першого рівня достатньо для початку відпрацювання інших артефактів, що залежать від нього. Отже, здійснюється нашарування відпрацювання артефактів, що дає змогу одночасно спостерігати за роботою до чотирьох студентів.

Висновки

1. Методична розробка з розглядуваної тематики здобула призове місце в номінації «Впровадження методу проєктів у навчально-виховній діяльності» у Конкурсі методичних розробок серед викладачів інформатики та програмування закладів фахової передвищої освіти м. Києва.

2. Підготовка студентів проєктним технологіям висувають перед студентами такі запитання, на які вони постійно шукають і знаходять відповіді під час навчання.

3. Підвищується зацікавленість до навчання.

4. З'являється розуміння роботи в команді.

5. Підвищується творчий потенціал.

6. Формується реальний підхід до розв'язання тих чи тих питань проєктів, зокрема організаційних, фахових та інших складових.

7. Упровадження наскрізних проєктів масштабу семестру чи курсу дають можливість отримати досить об'ємний робочий проєкт, який меншою мірою залежить від масштабу. Відпрацьована

архітектура програмного забезпечення дає змогу нарощувати нову функціональність і вносити нові дані.

8. Питання, які потрібно розв'язати, виникають раніше отриманої інформації щодо того, як саме їх вирішити, і дуже часто студенти дістають інформацію із загальновідомих інтернет-джерел (іноді англомовних), що зі свого боку привчає їх до самостійної роботи.

9. Масштабні проєкти завжди потребують правильних назв елементів програмного забезпечення. Це дисциплінує студентів та примушує їх замислюватися над назвами.

10. Масштабні проєкти та зміни до вимог зумовлюють потребу здійснювати рефакторинг проєкту (зміну назв, архітектури проєкту). Студенти набувають досвіду рефакторингу, вони відчувають переваги уніфікованого процесу перед каскадним.

11. Студенти протягом наскрізного проєкту самі вирішують чим їм займатися в цьому проєкті. Деякі студенти посідають роль керівників, деякі — роль виконавців із написання коду, а деякі — з відпрацювання діаграм та інших артефактів (прецедентів, додаткової специфікації тощо). Це дає можливість студентам набути навичок формування самоорганізованих груп та роботи в команді.

12. Деякі студенти потребують додаткової уваги викладача і можуть розкрити свій потенціал завдяки тісній співпраці з викладачем або більш краще підготовленим студентом.

13. Масштабні наскрізні проєкти потребують зберігання змін, що було здійснено під час ітерацій (практичних занять). А отже, студенти мають зберігати свій проєкт на GitHub. Їх ніхто не навчає цьому, вони самі освоюють таку можливість і користуються нею. Студенти набувають навичок роботи з GitHub, а також привчаються до самостійного освоєння матеріалу.

14. Використання хмарного середовища, а саме: онлайн-редактора мови моделювання UML UMLetino, компілятора C# dotnetfiddle, вебсервісу для хостингу IT-проєктів і їх спільної розробки GitHub, платформи для проведення онлайн-занять та відеоконференцій Zoom тощо дає можливість студентам освоювати нові тенденції в підході до процесу створення програмного забезпечення.

15. Упровадження наскрізних проєктів масштабу семестру чи курсу та використання хмарного середовища дають змогу набути істотного досвіду практичної роботи та портфоліо, на які студенти можуть покладатися під час пошуку роботи.

Пропозиції:

1. Ширше впроваджувати наскрізні проєкти до навчального процесу.

2. Залучати компанії, що створюють програмне забезпечення, до розроблення матеріалу, який викладається. Це може бути проєкт, котрий створюється протягом приблизно 18 (9) ітерацій, із постійним нарощенням функціоналу та покращенням якості. Проєкт може мати цікавий інтерфейс, застосовувати кілька мов програмування (технологій програмування), складатися з кількох частин та мати засоби сумісності між своїми частинами та іншими проєктами.

Список використаної літератури

*1. **Вебсервіс** для спільної розробки програмного забезпечення GitHub: інформаційна сторінка. URL: https://github.com (дата звернення: 04.04.2022).*

*2. **Онлайн**-компілятор C#: інформаційна сторінка. URL: https://dotnetfiddle.net (дата звернення: 04.04.2022).*

*3. **Онлайн**-редактор мови моделювання UML: інформаційна сторінка. URL: http://www.umletino.com/umletino.html (дата звернення: 04.04.2022).*

*4. **Опис** уніфікованого процесу створення програмного забезпечення: інформаційна сторінка. URL: https://en.wikipedia.org/wiki/Unified_Process (дата звернення: 04.04.2022).*

*5. **Microsoft** Visual Studio: інформаційна сторінка. URL: https://visualstudio.microsoft.com (дата звернення: 26.11.2021).*

И. М. Гаманюк

ВНЕДРЕНИЕ СКВОЗНОГО ПРОЕКТА В ПРОЦЕСС ПОДГОТОВКИ СТУДЕНТОВ В ОБЛАСТИ ИНЖЕНЕРИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Материал, касающийся создания реальных проектов, очень редко случается. Для решения этой и других проблем было введено в процесс подготовки студентов использование сквозных проектов масштаба семестра или курса. Во время проекта отрабатываются вопросы итерационной разработки информационных систем, где количество классов и конструкций считается десятками и используются паттерны проектирования. Все они должны быть согласованы и занимать свое место и роль. После каждой итерации студенты имеют работоспособный программный модуль. Полученный опыт придает студенту чувство уверенности в своих силах, потому что он принимает участие, возможно впервые, в создании существенного приложения. Подготовленные студенты привносят в проект интересные решения, и роль преподавателя состоит в распространении полученного опыта на следующий год (семестр), уменьшая тем самым зависимость от опытных специалистов.

Ключевые слова: подготовка студентов; сквозной проект; инженерия программного обеспечения; итерационная разработка.

I. M. Gamaniuk

OPTION FOR EVALUATION OF DEVELOPMENT OF DECISION SUPPORT SYSTEM REQUIREMENTS

Material relating to the creation of real projects is very rare. To solve this and other problems, the use of end-to-end projects of the semester or course scale was introduced into the process of preparing students. During the project, issues of iterative development of information systems are worked out, where the number of classes and structures is considered to be dozens and design patterns are used. All of them must be coordinated and take their place and role. After each iteration, students have a workable software module. The experience gained gives the student a sense of self-confidence, because he takes part, perhaps for the first time, in the creation of a significant application. Prepared students bring interesting solutions to the project and the role of the teacher is to extend the experience gained to the next year (semester), thereby reducing dependence on experienced professionals.

More and more activities are moving to the electronic form, the implementation of the function of estimating the processing of the requirement is becoming easier, so research in this area has prospects.

The purpose of the study: the creation of a methodological development for the preparation of students in the field of software engineering.

Task: describe ways to ensure the quality of practical classes and the introduction of project technologies in pedagogical activities.

Within the framework of the innovative content of education in the disciplines of Object-Oriented Programming, Software Modeling and Design, project technology has been introduced into the educational process.

At this time, a unified approach to software development has become very popular.

Within the framework of practical classes in each of the disciplines, a separate cross-cutting project is determined, which must be worked out. One or two practical sessions are an iteration of a cross-cutting project that creates new functionality or improves existing one. At the same time, a new construction of the studied programming or a new diagram (document) is being implemented in the project, which is being worked out as a result of the design.

In order to improve the quality of training at the end of certain iterations, video recording of the development of the developed part of the project is carried out. Students are divided into groups. The most prepared student leads the development. He fully understands the process and is ready to provide effective assistance in the development of artifacts - one or another diagram (document), or one or another software module.

Students work on a chart of artifacts and find out what exactly their group will do, on which artifacts their artifact depends. This affects the time the artifact is first developed. The artifact is proposed for development after a certain readiness of artifacts, on the basis of which the necessary artifact is developed. At the initial iterations it is necessary to make a Development Plan, which records the order of development of artifacts.

Software development during video capture is carried out vertically. When building, it is desirable not to change the students during the initial iterations, so that they clearly fix their place of work in the architecture of the project and feel the variable part of their work and the constant part of their work.

Implementing end-to-end semester or course projects and using a cloud environment provides significant hands-on experience and a portfolio that students can rely on when looking for work.

Suggestions: Involve software companies in the development of the material taught. This can be a project that is created over approximately 18 (9) iterations, with continuous functionality and quality improvements. A project can have an interesting interface, use several programming languages (programming technologies), consist of several parts and have the means of compatibility between its parts and other projects.

Keywords: student preparation; end-to-end project; software engineering; iterative development.