

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

по виконанню лабораторних робіт

з дисципліни

«Технологія створення програмних продуктів»

для студентів 3 курсу

Освітньо-кваліфікаційний рівень – «Бакалавр»

Напрямок підготовки – «Комп'ютерні науки»

Спеціальність – інформаційні управляючі системи та технології

Затверджено на засіданні  
методичної комісії ф-ту  
Комп'ютерних наук  
Протокол №\_\_ від \_\_\_\_\_  
Декан факультету  
\_\_\_\_\_ Коваленко Л.Б.

Затверджено на засіданні  
кафедри Інформатики  
Протокол №\_\_ від \_\_\_\_\_  
Зав. кафедри  
\_\_\_\_\_ Мещеряков В.І.

Одеса, 2014

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ  
по виконанню лабораторних робіт  
з дисципліни  
«Технологія створення програмних продуктів»  
для студентів 3 курсу  
Освітньо-кваліфікаційний рівень – «Бакалавр»  
Напрямок підготовки – «Комп'ютерні науки»  
Спеціальність – інформаційні управляючі системи та технології

Одеса, 2014

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ  
по виконанню лабораторних робіт  
з дисципліни  
«Технологія створення програмних продуктів»  
для студентів 3 курсу  
Освітньо-кваліфікаційний рівень – «Бакалавр»  
Напрямок підготовки – «Комп'ютерні науки»  
Спеціальність – інформаційні управляючі системи та технології  
напрямок підготовки – «Менеджмент»

Затверджено на засіданні  
методичної комісії ф-ту  
Комп'ютерних наук  
Протокол №\_\_ від \_\_\_\_\_

Одеса, 2014

Методичні вказівки до виконання лабораторних робіт з дисципліни «Технологія створення програмних продуктів» для студентів 3 курсу, Освітньо-кваліфікаційний рівень – «Бакалавр» Напрямок підготовки – «Комп'ютерні науки». Спеціальність – інформаційні управляючі системи та технології

Укладачі:

Гнатовська Г.А., к.т.н., доцент кафедри інформатики

Вохменцева Т.Б., ст.викл. кафедри інформатики.

## ЗМІСТ

Передмова .....	6
Лабораторна робота №1 .....	8
Лабораторна робота №2 .....	12
Лабораторна робота №3 .....	15
Лабораторна робота №4 .....	18
Лабораторна робота №5 .....	20
Лабораторна робота №6 .....	45
Лабораторна робота №7 .....	53
Лабораторна робота №8 .....	68
Лабораторна робота №9 .....	69
Література.....	88

## Передмова

Мета методичних вказівок – поглиблене вивчення та закріплення лекційного матеріалу щодо вивчення дисципліни «Технологія створення програмних продуктів»

Створення програмної системи - вельми трудомістке завдання, і фахівець повинен мати уявлення про методи аналізу, проектування, реалізації та тестування програмних систем, орієнтуватися в існуючих підходах і технологіях.

Проектування програмних продуктів, як і будь-яких інших складних систем, виконується поетапно з використанням блочно-ієрархічного підходу, який передбачає розробку продукту по частинах з наступною збіркою. На кожному етапі виконуються певні проектні операції, які відповідним чином документуються. Крім того програмний продукт повинен супроводжуватися різного роду програмою документацією (керівництво програміста, користувача, оператора). «Технологія створення програмних продуктів» - це дисципліна, яка розглядає додаток теорії, знань і практики для ефективної побудови програмних систем, що задовольняють вимогам користувача і замовника. У рамках дисципліни вивчаються всі процеси, що ведуть до створення програмних продуктів (ПП): від розробки вимог до ПП через проектування, розробку та атестацію до модернізації програмних систем.

*Мета дисципліни:* забезпечити отримання студентами теоретичних знань і практичних навичок щодо сучасних технологій створення програмних продуктів.

*Предмет дисципліни:* моделі життєвого циклу ПЗ та основні методології та засоби розробки ПЗ. Технологія створення та документування програмних продуктів.

*Структурно-логічне місце дисципліни:* попереднє вивчення дисциплін “Алгоритмізація та програмування”, “Об’єктно-орієнтоване програмування”, “Технології комп’ютерного проектування”, “Організація баз даних та знань”, “Управління ІТ-проектами”, “Моделювання систем”.

В результаті вивчення дисципліни студенти повинні:

**знати:**

- основні моделі ЖЦ програмних засобів,
- сучасні методології розробки, умови їх застосування,
- правила документування текстів програм та іменування змінних і об’єктів,
- основні моделі та методи проектування архітектури ПЗ, патерни та шаблони проектування,

- засоби автоматизації розробки програмних продуктів;

**вміти:**

- вибирати стратегії для планування життєвого циклу системи;
- визначати організаційну, економічну, технічну та операційну здійсненність проекту;

- реалізовувати та тестувати компоненти програмного забезпечення;
- аналізувати вимоги замовника до програмних продуктів;

**професійні компетенції:**

– знання стандартів, методів і засобів управління процесами життєвого циклу інформаційних систем, продуктів і сервісів інформаційних технологій; володіння технологією розроблення програмного забезпечення відповідно до вимог і обмежень замовника;

– знання сучасних технологій та інструментальних засобів розробки програмних систем, уміння їх застосовувати на всіх етапах життєвого циклу.

По кожній лабораторній роботі студент повинен скласти **звіт**, якій містить в собі:

1. Назву роботи. Мету.

2. Умову завдання згідно варіанту.
3. Хід виконання роботи.
4. Відповіді на контрольні питання.

Оформлений звіт захищається студентом усно.

Варіант індивідуального завдання надається викладачем.

Перед виконанням лабораторних робіт у комп'ютерному класі студенти зобов'язані пройти інструктаж з техніки безпеки та охорони праці.

### **Правила техніки безпеки та охорона праці**

Згідно з «Правилами техніки безпеки в лабораторіях інформатики» студентам забороняється:

- з'являтися та знаходитись приміщенні в нетверезому стані;
- ставити поруч з клавіатурою ємності з рідиною;
- перебувати в приміщенні в верхній одежі та завалювати нею робочі столи та стільці;
- працювати в лабораторії більше 6-ти годин на день (для вагітних жінок – більше 4-х годин);
- за власною ініціативою змінювати закріплені за ними робочі місця та знаходитись в приміщенні під час роботи іншої учбової групи;
- самостійно виконувати вмикання електроживлення лабораторії та заміну складових частин ПК, що вийшли із ладу.

У випадку виявлення несправностей обчислювальної техніки студент повинен сповістити про це викладача чи будь-кого з навчально-допоміжного персоналу лабораторії.

## Лабораторна робота №1

***Етапи розробки програмного забезпечення при структурному підході до програмування: Стадія «Технічне завдання» Вибір моделі ЖЦ. Визначення версій. Планування випуску версій. Розробка шаблону.***

**Мета:** ознакомиться с правилами написання технічного завдання. Розробка шаблону «Технічного завдання»

### Теоретичні відомості

#### Розробка технічного завдання

Технічне завдання являє собою документ, в якому сформульовані основні цілі розробки, вимоги до програмного продукту, визначено терміни та етапи розробки та регламентований процес приймально-здавальних випробувань. У розробці технічного завдання беруть участь як представники замовника, так і представники виконавця. В основі цього документа лежать вихідні вимоги замовника, аналіз передових досягнень техніки, результати виконання науково-дослідних робіт, передпроектних досліджень, наукового прогнозування і т. п.

#### Порядок розробки технічного завдання

Розробка технічного завдання виконується в наступній послідовності. Насамперед, встановлюють набір виконуваних функцій, а також перелік і характеристики вихідних даних. Потім визначають перелік результатів, їх характеристики і способи подання.

Далі уточнюють середовище функціонування програмного забезпечення: конкретну комплектацію і параметри технічних засобів, версію операційної системи і, можливо, версії і параметри іншого встановленого програмного забезпечення, з яким належить взаємодіяти майбутньому програмному продукту.

У випадках, коли розробляється програмне забезпечення збирає і зберігає деяку інформацію або включається в управління будь-яким технічним процесом, необхідно також чітко регламентувати дії програми у разі збоїв обладнання та енергопостачання.

#### **1. Загальні положення**

1.1. Технічне завдання оформляють відповідно до ГОСТ 19.106-78 на аркушах формату А4 та А3 по ГОСТ 2.301-68, як правило, без заповнення полів аркуша. Нумери аркушів (сторінок) проставляють у верхній частині аркуша над текстом.

1.2. Лист затвердження і титульний лист оформляють відповідно до ГОСТ 19.104-78. Інформаційну частину (анотацію і зміст), лист реєстрації змін допускається в документ не включати.

1.3. Для внесення змін і доповнень в технічне задні на наступних стадіях розробки програми або програмного виробу випускають доповнення до нього. Узгодження і затвердження доповнення до технічного завдання проводять у тому ж порядку, який встановлений для технічного завдання.

1.4. Технічне завдання повинне містити наступні розділи:

- введення;
- найменування та область застосування;
- підставу для розробки;



- призначення розробки;
- технічні вимоги до програми або програмного виробу;
- техніко-економічні показники;
- стадії і етапи розробки;
- порядок контролю та приймання;
- додатки.

Залежно від особливостей програми або програмного виробу допускається уточнювати зміст розділів, вводити нові розділи або об'єднувати окремі з них. При необхідності допускається в технічне завдання включати додатки.

## **2. Зміст розділів**

2.1. Вступ повинен включати коротку характеристику області застосування програми або програмного продукту, а також об'єкта (наприклад, системи), в якому передбачається їх використовувати. Основне призначення введення - продемонструвати актуальність даної розробки і показати, яке місце ця розробка займає в ряду подібних.

2.2. У розділі «Найменування та область застосування» вказують найменування, коротку характеристику області застосування програми або програмного виробу та об'єкта, в якому використовують програму або програмне виріб.

2.3. У розділі «Підстава для розробки» повинні бути зазначені:

- документ (документи), на підставі яких ведеться розробка. Таким документом може служити план, наказ, договір і т. п.
- організація, що затвердила цей документ, і дата його затвердження;
- найменування і (або) умовне позначення теми розробки.

2.4. У розділі «Призначення розробки» повинно бути вказано функціональне та експлуатаційне призначення програми або програмного виробу.

2.5. Розділ «Технічні вимоги до програми або програмного виробу» повинен містити такі підрозділи:

- вимоги до функціональних характеристик;
- вимоги до надійності;
- умови експлуатації;
- вимоги до складу і параметрів технічних засобів;
- вимоги до інформаційної та програмної сумісності;
- вимоги до маркування та упаковки;
- вимоги до транспортування і зберігання;
- спеціальні вимоги.

2.5.1. У підрозділі «Вимоги до функціональних характеристик» повинні бути зазначені вимоги до складу виконуваних функцій, організації вхідних та вихідних даних, тимчасовим характеристикам і т. п.

2.5.2. У підрозділі «Вимоги до надійності» повинні бути зазначені вимоги до забезпечення надійного функціонування (забезпечення сталого функціонування, контроль вхідної та вихідної інформації, час відновлення після відмови і т. п.).

2.5.3. У підрозділі «Умови експлуатації» повинні бути зазначені умови експлуатації (температура навколишнього повітря, відносна вологість і т. п. для обраних типів носіїв даних), при яких повинні забезпечуватися задані характеристики, а також вид обслуговування, необхідну кількість і кваліфікація персоналу.

2.5.4. У підрозділі «Вимоги до складу і параметрів технічних засобів» вказують необхідний склад технічних засобів із зазначенням їх технічних характеристик.

2.5.5. У підрозділі «Вимоги до інформаційної та програмної сумісності» мають бути вказані вимоги до інформаційних структур на вході і виході і методам вирішення, вихідних кодів, мов програмування. При необхідності повинна забезпечуватися захист інформації та програм.

2.5.6. У підрозділі «Вимоги до маркування та упаковки» в загальному випадку вказують вимоги до маркування програмного виробу, варіанти і способи упаковки.

2.5.7. У підрозділі «Вимоги до транспортування і зберігання» мають бути вказані для програмного виробу умови транспортування, місця зберігання, умови зберігання, умови складування, терміни зберігання в різних умовах.

2.5.8. У розділі «Техніко-економічні показники» повинні бути зазначені: орієнтовна економічна ефективність, передбачувана річна потреба, економічні переваги розробки в порівнянні з кращими вітчизняними і зарубіжними зразками або аналогами.

2.6. У розділі «Стадії та етапи розробки» встановлюють необхідні стадії розробки, етапи і зміст робіт (перелік програмних документів, які повинні бути розроблені, узгоджені та затверджені), а також, як правило, терміни розробки і визначають виконавців.

2.7. У розділі «Порядок контролю і приймання» повинні бути зазначені види випробувань і загальні вимоги до приймання роботи.

2.8. У додатках до технічним завданням при необхідності наводять:

- перелік науково-дослідних та інших робіт, що обґрунтовують розробку;
- схеми алгоритмів, таблиці, описи, обґрунтування, розрахунки та інші документи, які можуть бути використані при розробці;
- інші джерела розробки. У випадках, якщо будь-які вимоги, передбачені технічним завданням, замовник не пред'являє, слід у відповідному місці зазначити «Вимоги не пред'являються».

## **Контрольні питання**

1. Наведіть етапи розробки програмного забезпечення.
2. Що включає в себе постановка задачі та передпроектні дослідження?
3. Перерахуйте функціональні та експлуатаційні вимоги до програмному продукту.
4. Перерахуйте правила розробки технічного завдання.
5. Назвіть основні розділи технічного завдання.

## **Виконання роботи**

1. Розробити технічне завдання на програмний продукт
2. Оформити роботу відповідно до ГОСТ 19.106-78 (<http://www.gosthelp.ru>.) При оформленні використовувати MS Office.
3. Здати і захистити роботу.  
Захист звіту з лабораторної роботи  
Звіт з лабораторної роботи повинен складатися з:
  1. Постановки завдання.
  2. Технічного завдання на програмний продукт.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора і в друкованому вигляді), демонстрації отриманих навичок і відповідях на питання викладача.

### **ЗАВДАННЯ**

Розробити технічне завдання на програмний продукт, призначений для наочної демонстрації графіків функцій одного аргументу  $y = f(x)$ . Розробляється програма повинна розраховувати таблицю значень і будувати графік функцій на заданому відрізку за заданою формулою і міняти крок аргументу і межі відрізка. Крім цього, програма повинна запам'ятовувати введені формули.

## Лабораторна робота №2

### *Етапи розробки програмного забезпечення при структурному підході до програмування: Стадія «Ескізний проект»*

**Мета:** навчитися створювати формальні моделі і на їх основі визначати специфікації розроблюваного програмного забезпечення.

### Теоретичні відомості

Підготовка до лабораторної роботи

1. Ознайомитися з лекційним матеріалом по темі «Етапи розробки програмного забезпечення. Аналіз вимог і визначення специфікацій програмного забезпечення» навчальної дисципліни «Технологія розробки програмного забезпечення». 2. Вивчити відповідні розділи у виданнях [1,2, 39, 47, 53,61]. 3. Ознайомитися з розд. 3.5 даного навчального посібника.

#### Розробка специфікацій

Розробка програмного забезпечення починається з аналізу вимог до нього. У результаті аналізу отримують специфікації розроблюваного програмного забезпечення, будують загальну модель його взаємодії з користувачем або іншими програмами і конкретизують його основні функції. При структурному підході до програмування на етапі аналізу і визначення специфікацій розробляють три типи моделей: моделі функцій, моделі даних і моделі потоків даних.

Оскільки різні моделі описують проєктоване програмне забезпечення з різних сторін, рекомендується використовувати відразу кілька моделей, що розробляються у вигляді діаграм, і пояснювати їх текстовими описами, словниками і т. п.

Структурний аналіз передбачає використання наступних видів моделей:

- діаграм потоків даних (DFD - Data Flow Diagrams), описують взаємодію джерел і споживачів інформації через процеси, які повинні бути реалізовані в системі;
- діаграм «сутність-зв'язок» (ERD - Entity-Relationship Diagrams), що описують бази даних розроблюваної системи;
- діаграм переходів станів (STD - State Transition Diagrams), що характеризують поведінку системи в часу;
- функціональних діаграм (методика SADT);
- специфікацій процесів;
- словника термінів. специфікації процесів

Специфікації процесів зазвичай представляють у вигляді короткого текстового опису, схем алгоритмів, псевдокод, Flow-форм або діаграм Насс- Шнейдермана.

#### Словник термінів

Словник термінів являє собою короткий опис основних понять, що використовуються при складанні специфікацій. Він повинен включати визначення основних понять предметної області, опис структур елементів даних, їх типів і форматів, а також усіх скорочень і умовних позначень

#### Діаграми переходів станів

За допомогою діаграм переходів станів можна моделювати подальше функціонування системи на основі її попереднього та поточного функціонування. модельована система в будь-який заданий момент часу знаходиться точно в одному з кінцевого безлічі станів. З плином часу вона може змінити свій стан, при цьому переходи між станами повинні бути точно визначені

#### Функціональні діаграми

Функціональні діаграми відбивають взаємозв'язку функцій розроблюваного програмного забезпечення. Вони створюються на ранніх етапах проектування систем, для того щоб допомогти проектувальнику виявити основні функції і складові частини проектованої системи і, по можливості, виявити і усунути істотні помилки. для створення функціональних діаграм пропонується використовувати методологію SADT

#### Діаграми потоків даних

Для опису потоків інформації в системі застосовуються діаграми потоків даних (DFD - Data flow diagrams). DFD дозволяє описати необхідну поведінку системи у вигляді сукупності процесів, що взаємодіють за допомогою зв'язують їх потоків даних. DFD показує, як кожен з процесів перетворює свої вхідні потоки даних у вихідні потоки даних і як процеси взаємодіють між собою

#### Діаграми «сутність-зв'язок»

Діаграма сутність-зв'язок - інструмент розробки моделей даних, що забезпечує стандартний спосіб визначення даних і відносин між ними. Вона включає сутності та взаємозв'язку, що відображають основні бізнес-правила предметної області. Така діаграма не надто деталізована, в неї включаються основні сутності і зв'язки між ними, які задовольняють вимогам, що пред'являються до ІС

## **Виконання роботи**

1. На основі технічного завдання з лабораторної роботи № 1 виконати аналіз функціональних та експлуатаційних вимог до програмного продукту.
2. Визначити основні технічні рішення (вибір мови програмування, структура програмного продукту, склад функцій ПП, режими функціонування) і занести результати в документ, званий «Ескізним проектом»
3. Визначити діаграми потоків даних для розв'язуваної завдання.
4. Визначити діаграми «сутність-зв'язок», якщо програмний продукт містить базу даних.
5. Визначити функціональні діаграми.
6. Визначити діаграми переходів станів.
7. Визначити специфікації процесів.
8. Додати словник термінів.
9. Оформити результати, використовуючи MS Office в вигляді ескізного проекту.
10. Здати і захистити роботу.

Звіт з лабораторної роботи повинен складатися з:

1. Постановки завдання.
2. Документу «Ескізний проект», що містить:
  - вибір методу рішення і мови програмування;
  - специфікації процесів;

- всі отримані діаграми;
- словник термінів.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.

### **Контрольні питання**

1. Назвіть етапи розробки програмного забезпечення.
2. Що таке життєвий цикл програмного забезпечення?
3. У чому полягає постановка задачі та передпроектні дослідження?
4. Назвіть функціональні та експлуатаційні вимоги до програмному продукту.
5. Перелічіть складові ескізного проекту.
6. Охарактеризуйте специфікації і моделі.

## Лабораторна робота №3

### ***Структурний підхід до програмування: Стадія «Технічний проект».*** ***Проектування інтерфейсу користувача. Розробка прототипу ПЗ***

**Мета:** вивчити питання проектування програмного забезпечення.

### **Теоретичні відомості**

Теоретична частина. складові технічного проекту ПРОЕКТ ТЕХНІЧНИЙ - образ наміченого до створенню об'єкта, представлений у вигляді його опису, схем, креслень, розрахунків, обґрунтувань, числових показників. технічний проект

Мета технічного проекту - визначення основних методів, використовуваних при створенні інформаційної системи, і остаточне визначення її кошторисної вартості. Технічне проектування підсистем здійснюється в Відповідно до затвердженого технічним завданням.

Технічний проект програмної системи докладно описує:

- виконувані функції і варіанти їх використання;
- відповідні їм документи;
- структури оброблюваних баз даних;
- взаємозв'язку даних; • алгоритми їх обробки.

Технічний проект повинен включати дані про обсяги та інтенсивність потоків оброблюваної інформації, кількості користувачів програмної системи, характеристиках обладнання та програмного забезпечення, що взаємодіє з проектованим програмним продуктом.

#### ***При розробці технічного проекту оформляються:***

- відомість технічного проекту. Загальна інформація по проекту;
- пояснювальна записка до технічного проекту. Вступна інформація, що дозволяє її споживачеві швидко освоїти дані по конкретному проекту;
- опис систем класифікації та кодування;
- перелік вхідних даних (документів). перелік інформації, яка використовується як вхідний потік і служить джерелом накопичення;
- перелік вихідних даних (документів). перелік інформації, яка використовується для аналізу накопичених даних;
- опис використовуваного програмного забезпечення. Перелік програмного забезпечення і СУБД, які планується використовувати для створення інформаційної системи;
- опис використовуваних технічних засобів. перелік апаратних засобів, на яких планується робота проектованого програмного продукту;
- проектна оцінка надійності системи. експертна оцінка надійності з виявленням найбільш благополучних ділянок програмної системи та її вузьких місць;
- відомість обладнання та матеріалів. перелік обладнання та матеріалів, які будуть потрібні в ході реалізації проекту.

*структурна схема*

Структурної називають схему, що відображає склад і взаємодія з управління частинами розроблюваного програмного забезпечення. Структурна схема визначається архітектурою розроблюваного ПЗ.

#### функціональна схема

Функціональна схема - це схема взаємодії компонентів програмного забезпечення з описом інформаційних потоків, складу даних в потоках та зазначенням використовуваних файлів і пристроїв.

#### Розробка алгоритмів

Метод покрокової деталізації реалізує спадний підхід до програмування і передбачає покрокову розробку алгоритму.

#### структурні карти

Методика структурних карт використовується на етапі проектування ПЗ для того, щоб продемонструвати, яким чином програмний продукт виконує системні вимоги. Структурні карти Константайна призначені для опису відносин між модулями.

Техніка структурних карт Джексона заснована на методі структурного програмування Джексона, який виявляє відповідність між структурою потоків даних і структурою програми. Основна увага в методі сконцентровано на відповідності вхідних і вихідних потоків даних.

## **Виконання роботи**

1. На основі технічного завдання з лабораторної роботи № 1 і специфікацій з лабораторної роботи № 2 розробити уточнені алгоритми програм, які складають цей програмний модуль. Використовувати метод покрокової деталізації .

2. На основі уточнених та доопрацьованих алгоритмів розробити структурну схему програмного продукту.

3. Розробити функціональну схему програмного продукту.

4. Представити структурну схему у вигляді структурних карт Константайна.

5. Представити структурну схему у вигляді структурних карт Джексона

6. Оформити результати, використовуючи MS Office в вигляді технічного проекту.

7. Здати і захистити роботу.

## **Завдання**

Звіт з лабораторної роботи повинен складатися з:

1. Структурною схеми програмного продукту.

2. Функціональної схеми.

3. Алгоритму програми.

4. Структурною карти Константайна.

5. Структурною карти Джексона.

6. Закінченої технічного проекту програмного модуля.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.



## **Контрольні питання**

1. Назвіть етапи розробки програмного забезпечення.
2. У чому полягає проектування програмного забезпечення?
3. Перерахуйте складові технічного проекту.
4. Охарактеризуйте структурний підхід до програмування.
5. З чого складаються структурна і функціональна схеми?
6. Охарактеризуйте метод покрокової деталізації при складанні алгоритмів програм.
7. Наведіть поняття псевдокоду.
8. У чому полягає методика Константайна?
9. У чому полягає методика Джексона?

## Лабораторна робота №4

### *Етапи розробки програмного забезпечення Стадія «Реалізація». Проектування та реалізація простого проекту ПЗ з інтерфейсом користувача*

**Мета:** розробити програмний продукт в відповідно до заданого варіантом.

### Теоретичні відомості

1. Ознайомитися з лекційним матеріалом по темі «Етапи розробки програмного забезпечення. Структурний підхід до програмування »навчальної дисципліни« Технологія розробки програмного забезпечення ».
2. Вивчити відповідні розділи у виданнях [1, 2, 5, 7, 40].
3. Ознайомитися з гл. 6 цього посібника.

### *Складання програмної документації*

Важливим етапом розробки програмного продукту є складання програмної документації. життєвий цикл програмного забезпечення містить спеціальний процес, присвячений цьому питанню. На кожен програмний продукт повинні складатися два типи документації - для розробників і для різних груп користувачів. програмна документація користувачів повинна містити всі необхідні відомості по експлуатації ПЗ. Аналогічно, документація розробника повинна містити відомості, необхідні для розробки та супроводження програмного забезпечення.

### *Види програмних документів*

Документування програмного забезпечення здійснюється відповідно до Єдиної системою програмної документації (ГОСТ 19.XXX). ГОСТ 19.101-77 містить види програмних документів для програмного забезпечення різних типів. У даному ГОСТі перераховані документи наступних типів:

- специфікація повинна містити перелік і короткий опис призначення всіх файлів програмного забезпечення, в тому числі і файлів документації на нього, і є обов'язковою для програмних систем, а також їх компонентів, що мають самостійне застосування;
- відомість утримувачів оригіналів (код виду документа - 05) повинна містити список підприємств, на яких зберігаються оригінали програмних документів. Необхідність цього документа визначається на етапі розробки і затвердження технічного завдання тільки для програмного забезпечення зі складною архітектурою;
- текст програми (код виду документа - 12) повинен містити текст програми з необхідними коментарями. Необхідність цього документа визначається на етапі розробки і затвердження технічного завдання;
- опис програми (код виду документа - 13) має містити відомості про логічну структуру і функціонуванні програми. Необхідність даного документа також визначається на етапі розробки та затвердження технічного завдання;
- відомість експлуатаційних документів (код виду документа - 20) повинна містити перелік експлуатаційних документів на програму, до яких належать документи з кодами 30, 31, 32, 33, 34, 35, 46. Необхідність цього документа також визначається на етапі розробки і затвердження технічного завдання;
- формуляр (код виду документа - 30) повинен містити основні характеристики програмного забезпечення, комплектність і відомості про експлуатацію програми;
- опис застосування (код виду документа - 31) має містити відомості про призначення програмного забезпечення, галузі застосування, застосовуваних методах,

класі розв'язуваних завдань, обмеженнях для застосування, мінімальній конфігурації технічних засобів;

- керівництво системного програміста (код виду документа - 32) має містити відомості для перевірки, забезпечення функціонування та налаштування програми на умови конкретного застосування;

- керівництво програміста (код виду документа 33) має містити відомості для експлуатації програмного забезпечення;

- керівництво оператора (код виду документа - 34) містить відомості для забезпечення процедури спілкування оператора з обчислювальною системою в процесі виконання програми;

- опис мови (код виду документа - 35) - опис синтаксису і семантики мови програми;

- керівництво з технічного обслуговування (код виду документа - 46) містить відомості для застосування програми при обслуговуванні технічних засобів.

## **Виконання роботи**

1. За результатами лабораторних робіт № 1-3 написати код програм для вирішення поставленого завдання на мові програмування, обраному на етапі ескізного проектування.

2. Налагодити програмний модуль.

3. Отримати результати роботи.

4. Оформити документацію до розробленого програмного забезпечення.

5. Здати і захистити роботу.

## **Завдання**

Звіт з лабораторної роботи повинен складатися з:

1. Лістинг програм.

2. Інтерфейсу користувача.

3. Документації до програмного забезпечення (керівництво користувача, керівництво системного програміста, керівництво програміста, керівництво оператора).

4. Результатів роботи програм.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.

## **Контрольні питання**

1. У чому полягає етап реалізації та налагодження програмного забезпечення?

2. Які існують інструментальні засоби розробки?

3. Охарактеризуйте етап стихійного програмування.

4. Охарактеризуйте етапи структурного та модульного програмування.

5. Що таке документація до програмного забезпечення?

## Лабораторна робота №5

### *Проектування програмної системи при об'єктом підході до програмування*

**Мета:** познайомити студентів та здійснити проектування згідно методології MSF – Microsoft Solutions Framework. За завданням викладача створити командний проект з розробки програмного продукту, використовуючи інтегровану середу Visual Studio 2012 та Team Foundation Server. Налаштувати параметри створеного командного проекту.

### Теоретичні відомості

Microsoft Solutions Framework (MSF) є методологією розробки ПЗ, яка являє собою узагальнення кращих проектних практик, які використовувалися командами розробників Microsoft. Дана методологія описує управління людьми і робочими процесами при розробці IT-рішень.

IT-рішення - розуміється як скоординована поставка набору елементів (таких як програмні засоби, документація, навчання та супровід), необхідних для задоволення бізнес-потреби конкретного замовника.

Концепція управління життєвим циклом додатків, прийнята розробниками ПЗ, призвела до того, що методологія MSF стала складовою частиною продукту Visual Studio Team System (VSTS), який реалізовував підхід Microsoft в плані - ALM.

У продукт VSTS увійшли шаблони процесів для реалізації положень моделі CMMI - MSF for CMMI і моделей гнучкої розробки ПЗ - MSF for Agile Scrum. Таким чином, VSTS є інструментарієм управління життєвим циклом додатків, який дозволяє створювати програмні системи різного призначення в командах, які дотримуються різних підходів до управління процесами розробки ПЗ.

Основними є такі принципи MSF.

1. Єдине бачення проекту, яке передбачає розуміння усіма зацікавленими особами цілей і завдань створення ПЗ.

2. Гнучкість - готовність до змін, що забезпечує можливість уточнення і зміни вимог у процесі розробки ПЗ, оперативного і швидкого реагування на поточні зміни умов проекту при незмінній ефективності управлінської діяльності.

3. Концентрація на бізнес-пріоритетах, що передбачає створення продукту з високим споживчим якістю і формування певної вигоди або віддачі. Для організацій, як правило, це отримання прибутку.

4. Заохочення вільного спілкування, що припускає відкритий і чесний обмін інформацією як усередині команди, так і з ключовими зацікавленими особами.

Універсальність моделі MSF визначається тим, що завдяки своїй гнучкості і відсутності жорстко встановлених зв'язків і процедур вона може бути застосована при розробці різних програмних додатків, які можуть використовуватися в бізнесі і повсякденному житті. Модель MSF базується на поєднанні двох моделей життєвого циклу програмних систем: каскадної і спіральної.

Завдяки своїй гнучкості дана модель може використовуватися для розробки широкого кола IT-рішень, вона охоплює весь життєвий цикл створення рішення, з самих ранніх етапів до запровадження. Модель процесів MSF поєднує в собі якості двох класичних моделей: каскадної і спіральної.

Процес MSF орієнтований на "віхи" (milestones). Віхи – ключові точки процесу розробки, які характеризують досягнення якого-небудь істотного результату. Модель процесів MSF враховує постійні зміни вимог до кінцевого продукту, процес розробки

складається з коротких циклів і являє собою поступальний рух від найпростіших ранніх версій продукту до його остаточного виду.

## Виконання роботи

### Підключення до Team Foundation Server

Для підключення до Team Foundation Server запустіть інтегровану середу Visual Studio 2012. На початковій сторінці виберіть посилання Підключитися до Team Foundation Server (рис. 4.1).

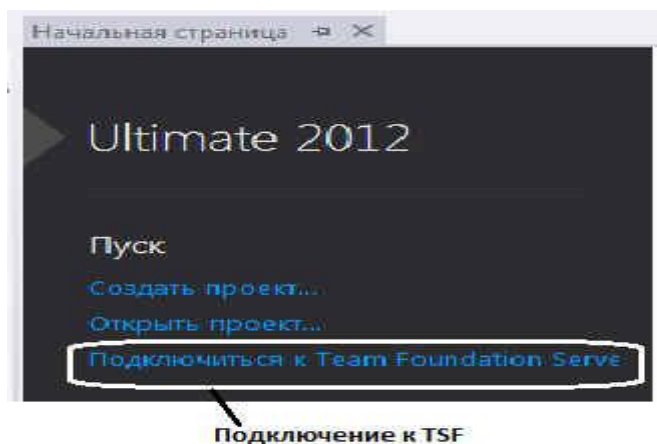


Рис. 4.1. Початкова сторінка Visual Studio 2012

У діалоговому вікні Підключення до Team Foundation Server натисніть кнопку Сервери (рис. 4.2).

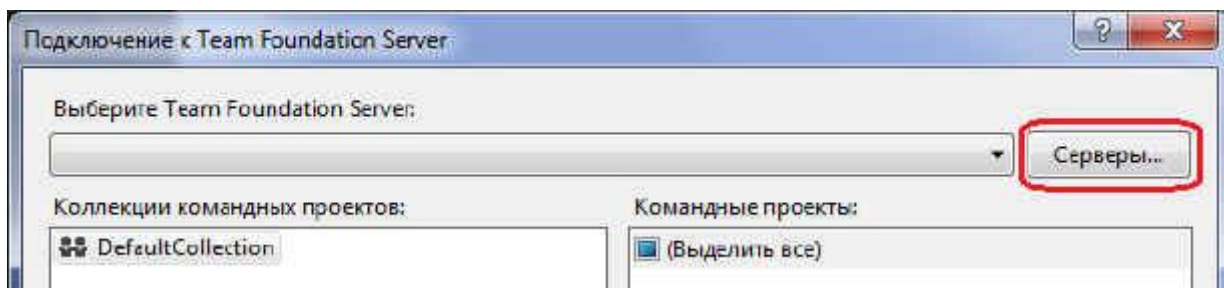


Рис. 4.2. Діалогове вікно Підключення до Team Foundation Server

У діалоговому вікні Додати або видалити Team Foundation Server натисніть кнопку Додати.

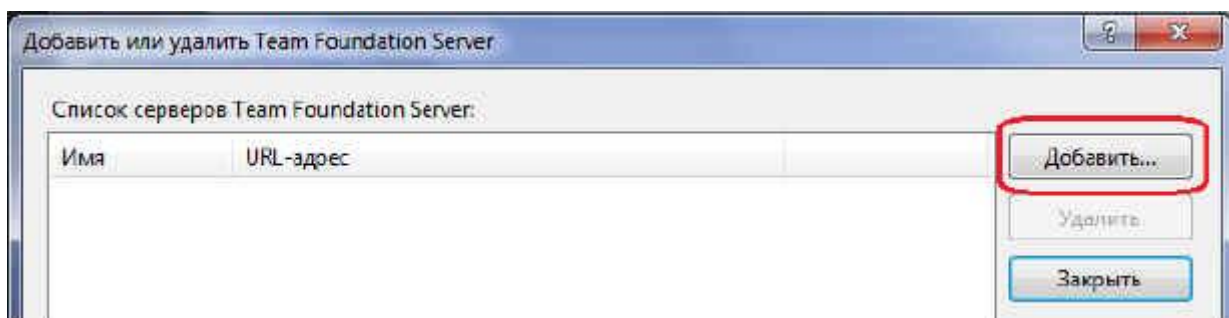


Рис. 4.3. Діалогове вікно Додати або видалити Team Foundation Server

У діалоговому вікні Додати Team Foundation Server введіть ім'я або URL-адресу сервера TFS. Для отримання параметрів підключення зверніться до адміністратора Team Foundation Server або адміністратору командного проекту. На рис. 4.4 вказано сервер localhost.

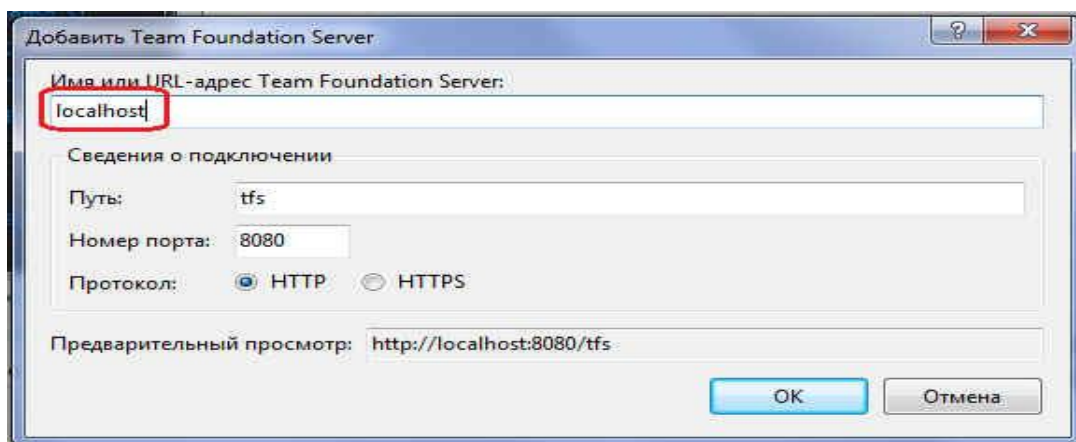


Рис. 4.4. Діалогове вікно Додати Team Foundation Server

При введенні імені сервера, поле Список серверів Team Foundation Server діалогового вікна Додати або видалити Team Foundation Server автоматично відображає формат url-адреси (рис. 4.5), наприклад: http://Імя\_сервера:порт/tfs, де ім'я\_сервера - це ім'я сервера, де розміщується Team Foundation Server; Порт - порт, який Team Foundation Server використовує (значення за замовчуванням 8080). Якщо сервер використовує інший номер порту, його необхідно вказати; tfs - шлях за замовчуванням до колекцій проектів, що зберігаються на сервері. Якщо у вашій групі використовується інший шлях, то його необхідно ввести. Після перевірки імені та URL-адреси TFS необхідно натиснути кнопку Закрити.

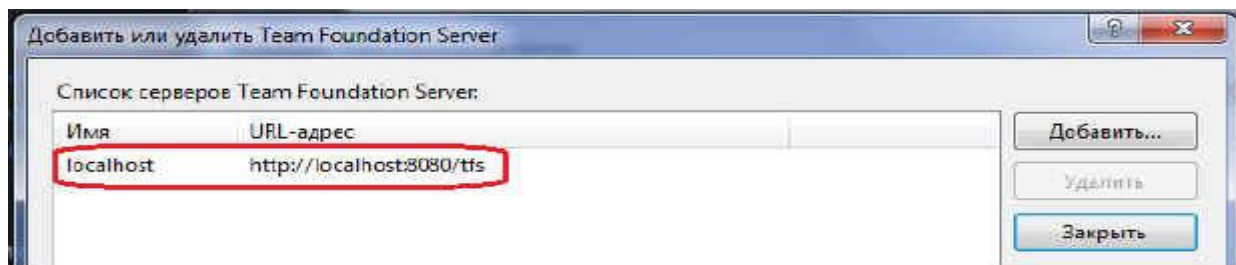


Рис. 4.5. Перевірка імені та URL-адреси TFS

Виберіть зі списку сервер TFS (рис. 4.6). Натисніть кнопку Підключитися. Якщо необхідно підключитися до існуючого командного проекту, поставте галочку перед потрібним проектом (наприклад, перед проектом Demo).

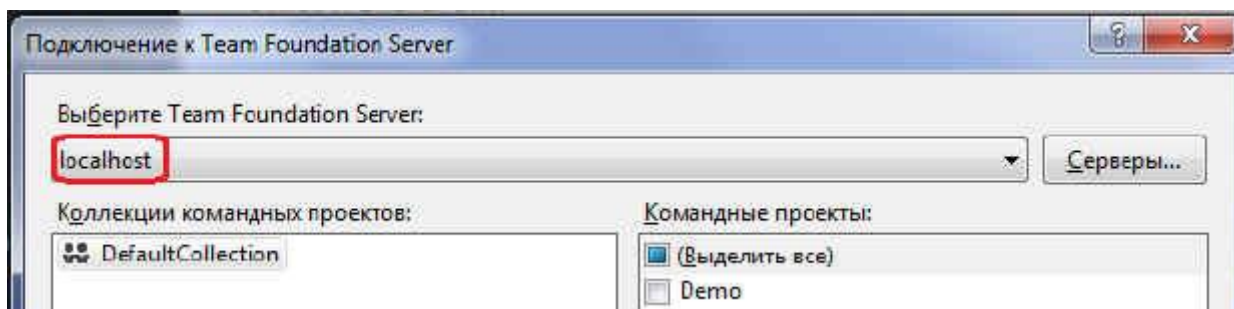


Рис. 4.6. Підключення до TFS і командним проектам

Результатом створення командного проекту є згенерувала інфраструктура, головна вкладка якої наведена на рис. 4.7.

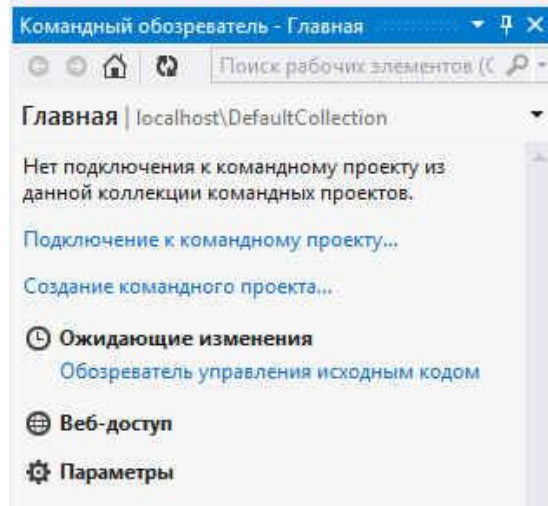


Рис. 4.7. Головна вкладка Командного оглядача

За допомогою посилань головною вкладкою Командного оглядача проектів можна підключитися до раніше створеного проекту (Підключення до командного проекту ...), створити новий проект (Створення командного проекту), отримати доступ до системи управління версіями (компонент Очікують зміни) через Обозреватель управління вихідним кодом, підключитися до командного проекту через веб-додомогою компонента веб-доступу і налаштувати параметри проекту через компонент параметри.

#### *Створення командного проекту*

Для створення командного проекту необхідно бути членом групи Адміністратори проекту. При створенні командного проекту виберіть посилання Створення командного проекту в Командному оглядачі (рис. 4.8), що призведе до відкриття майстра створення командного проекту.

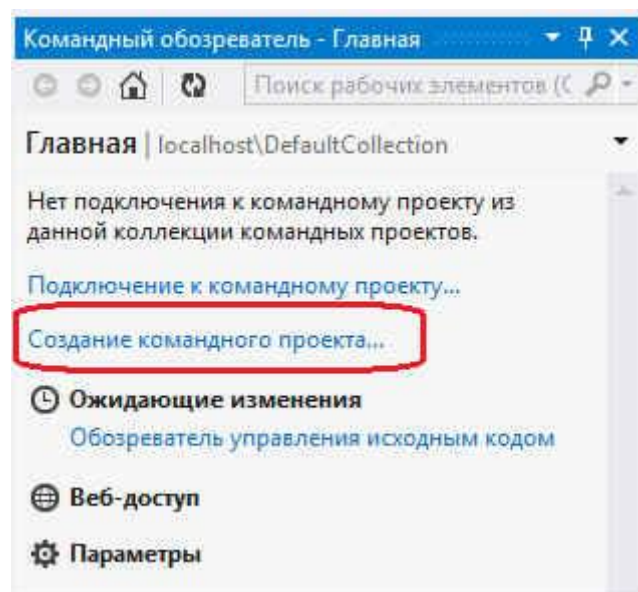


Рис. 4.8. Створення командного проекту

На сторінці Вкажіть параметри командного проекту введіть ім'я та опис для створюваного командного проекту (рис. 4.9). Ім'я проекту має бути унікальним і включати не більше ніж 64 символу. Доцільно проектам давати такі імена, які відображали б

основну мету створення програмного продукту. Це дозволить всім учасникам проекту легше асоціювати проект з його призначенням.

Опис командного проекту зберігається в Team Foundation Server і являє собою опис сайту SharePoint для не обов'язкового порталу командного проекту. Після введення імені та опису створюваного командного проекту натисніть кнопку Далі.

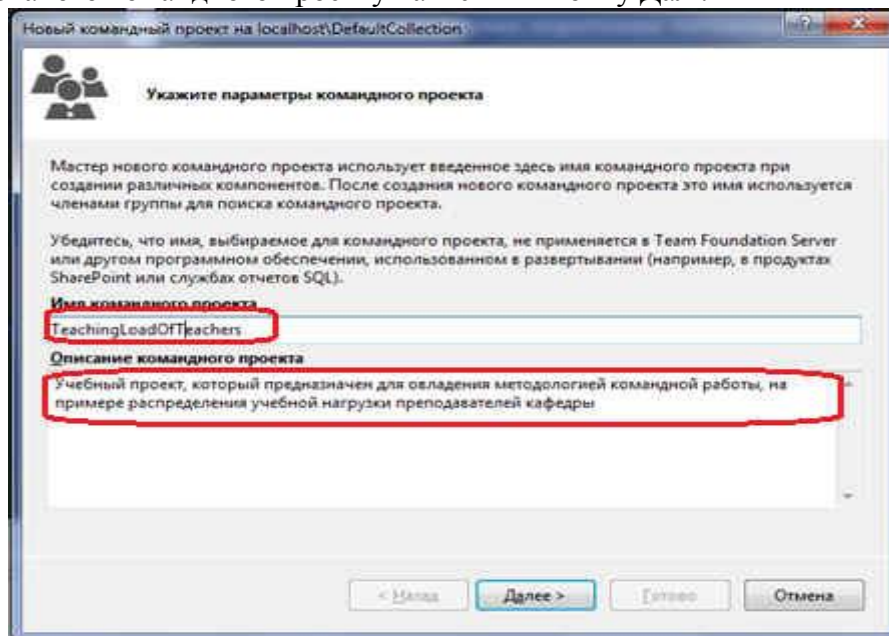


Рис. 4.9. Формування імені та описи командного проекту

На сторінці Вибрати шаблон процесів задається шаблон процесів проекту. У Visual Studio 2012 Team Foundation Server є три шаблону. Шаблон MSF for CMMI Process Improvement 6.0 призначений для великих команд зі строго формальним підходом до управління проектами на основі моделі CMM / CMMI. Шаблон MSF for Agile Software Development 6.0 визначає гнучкий підхід до управління проектами розробки програмного забезпечення.

Шаблон Microsoft Visual Studio Scrum 2.0. призначений для невеликих команд (до 7 - 10 учасників), які використовують гнучку методологію і термінологію Scrum. У лабораторному практикумі будемо використовувати шаблон Scrum 2.0 (рис. 4.10), так як даний шаблон найбільш підходить для вивчення управління командними проектами, на прикладі навчальних проектів, в студентських групах. Після шаблону командного проекту натисніть кнопку Далі.



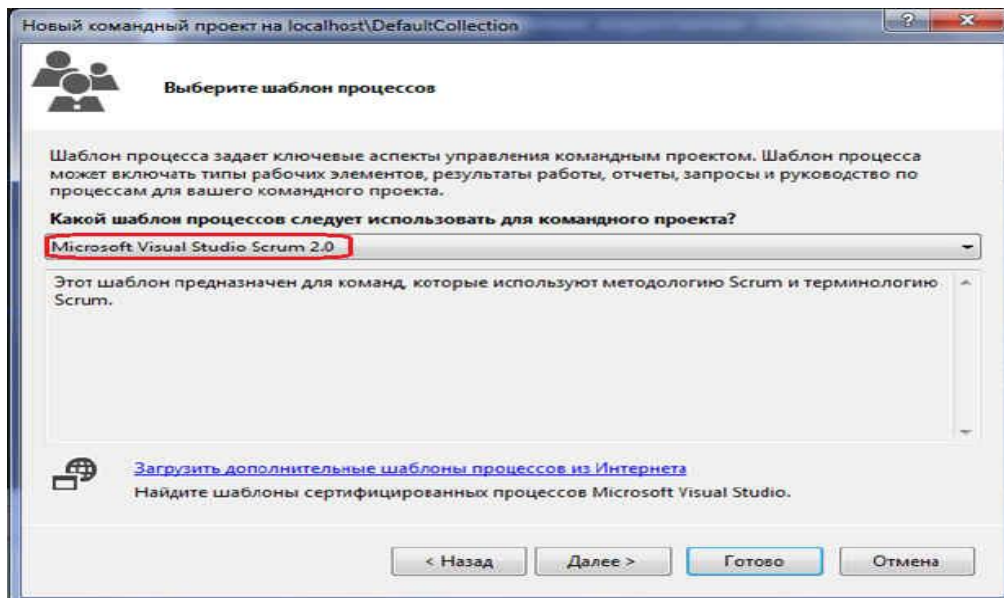


Рис. 4.10. Завдання шаблона процесів командного проекту

На сторінці Вказівка параметрів системи управління версіями задаються параметри системи управління версіями. Для навчального командного проекту доцільно створити порожню папку системи управління версіями, зазначивши перемикач перед відповідним рядком (рис. 4.11).

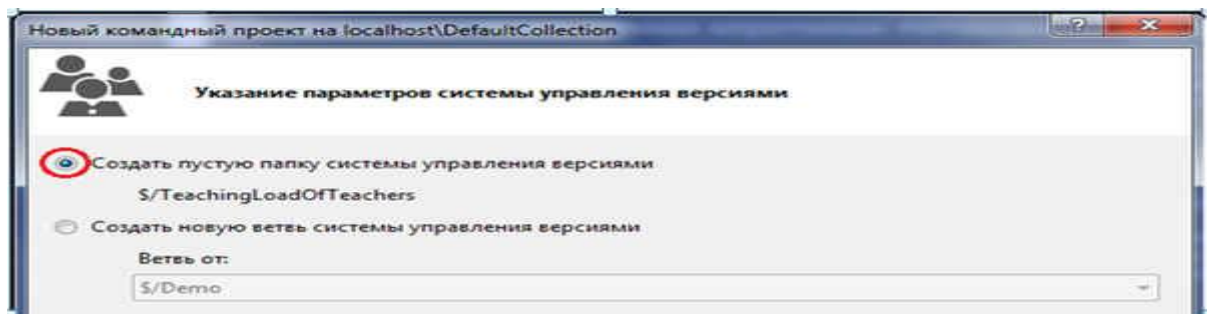


Рис. 4.11. Формування параметрів системи управління версіями

На сторінці Підтвердіть параметри командного проекту перевірте параметри проекту і якщо вони відповідають заданим значенням, то натисніть кнопку Готово (рис. 4.12).

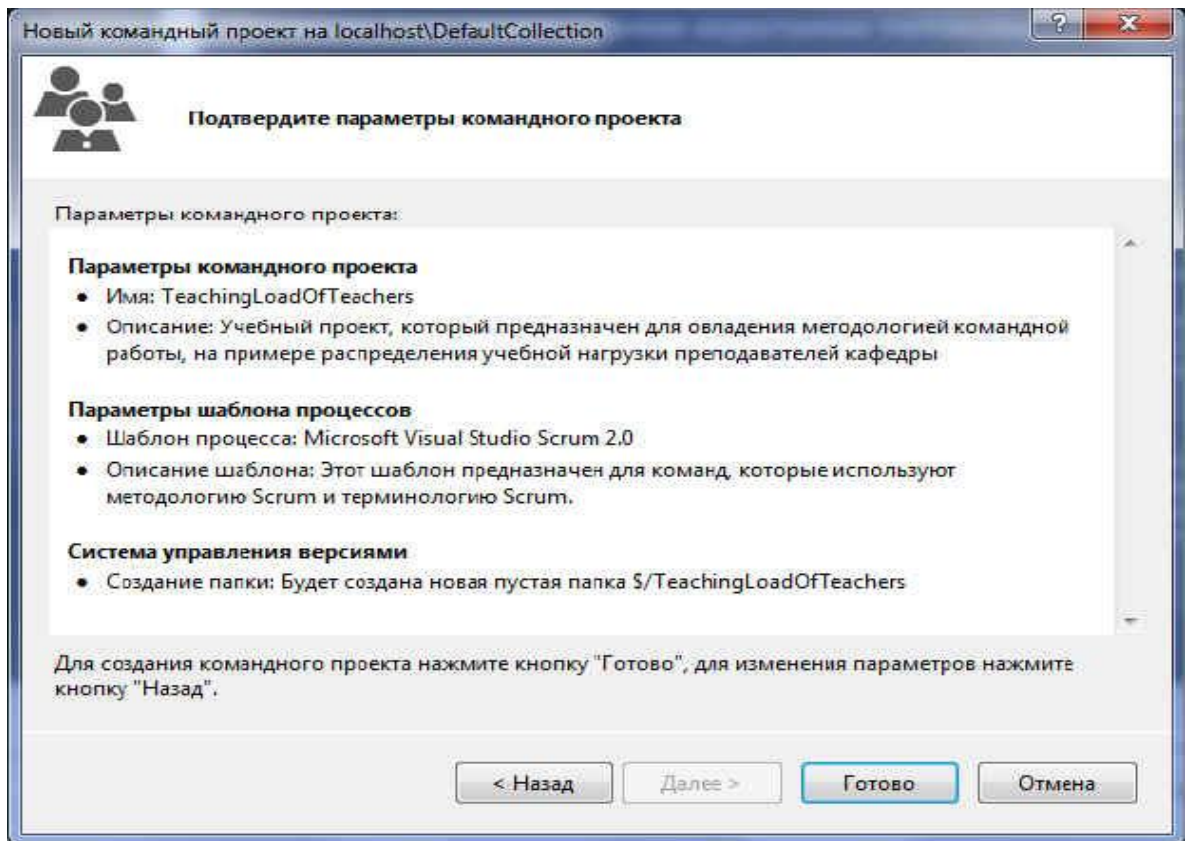


Рис. 4.12. Параметры командного проекту

На сторінці Стан створення командного проекту відображається процес побудови проекту (рис. 4.13).

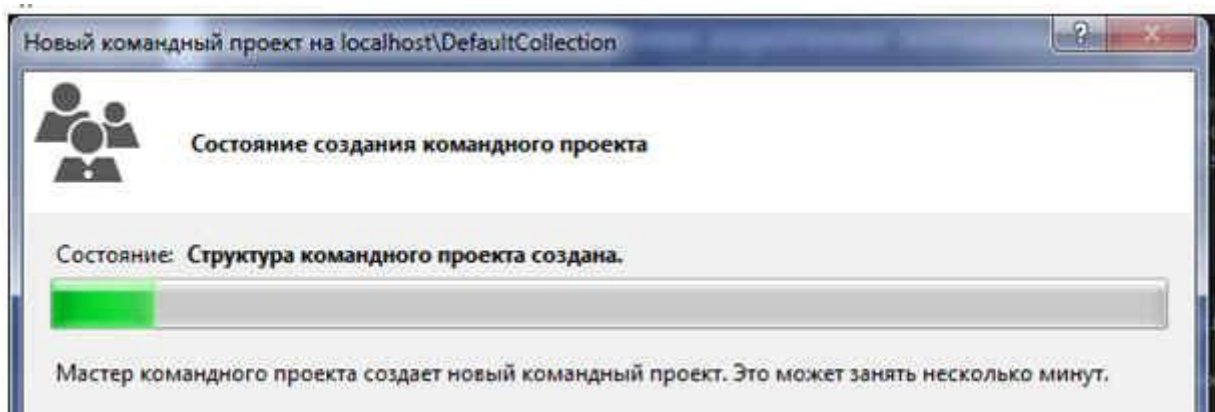


Рис. 4.13. Процесс створення командного проекту

На сторінці Командний проект створений надається інформація про місце створення проекту і посилання на журнал його створення (рис. 4.14). Натисніть кнопку Закрити. Якщо при створенні командного проекту майстер виявить якусь неполадку, то з'явиться повідомлення про помилку з описом цієї неполадки і пропозицією дії з її усунення. У цьому випадку виберіть посилання Переглянути журнал створення командного проекту та перегляньте журнал помилок і виключень.

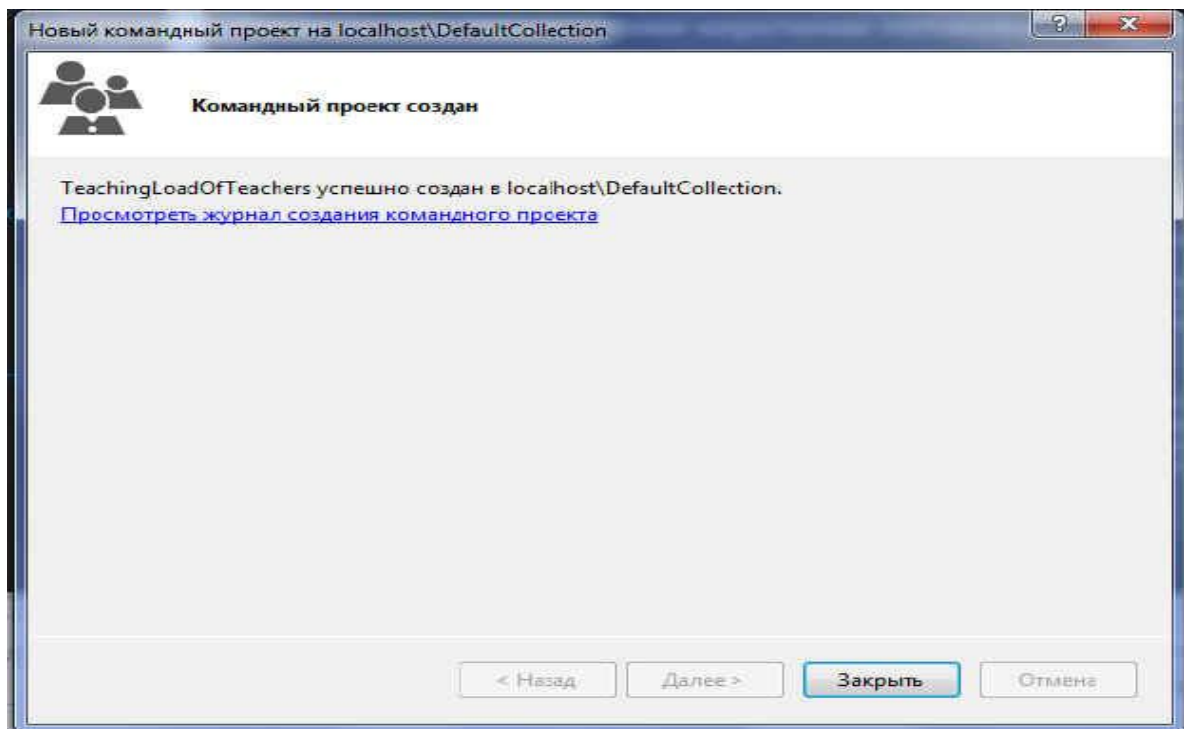


Рис. 4.14. Завершения работы мастера створення командного проекту

Результатом роботи майстра створення нового командного проекту є створення інфраструктури проекту, головна сторінка якого наведена на рис. 4.15.

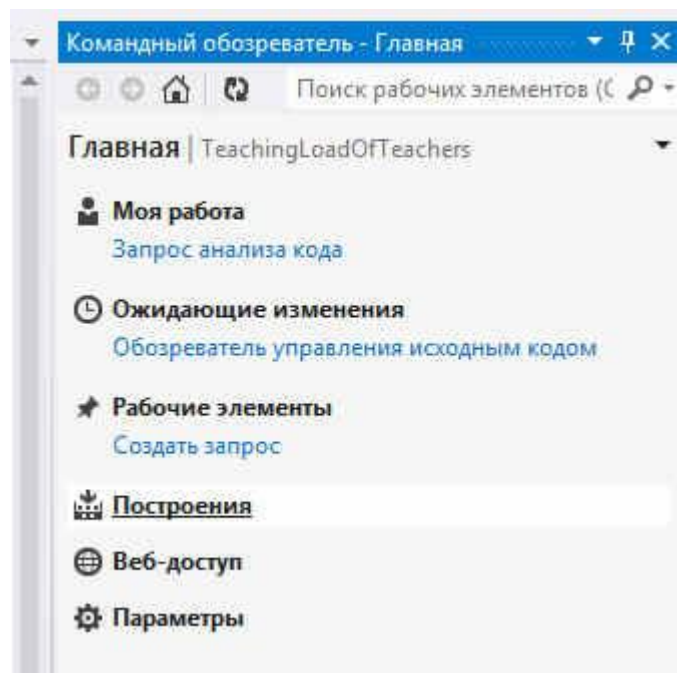


Рис. 4.15. Головна сторінка командного проекту

На будь-якому кроці створення командного проекту можна повернутися на крок назад і відредагувати раніше проведені дії. Якщо операція по створенню командного проекту не завершується успішно, можливо, деякі компоненти створені, а інші ні. У такому випадку слід визначити, що викликало часткове створення, видалити частково створений проект, вирішити можливі проблеми, а потім створити інший командний проект.

## Налаштування області та ітерацій

Після створення командного проекту необхідно налаштувати параметри проекту. Для цього на вкладці Головна командного оглядача виберіть компонент Параметри (рис. 5.1).

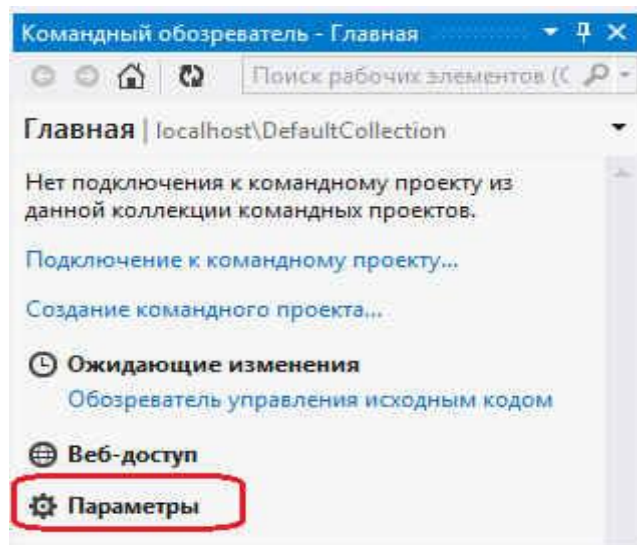


Рис. 5.1. Вибір компонента Параметри

На вкладці Параметри командного оглядача є посилання для установки значень параметрів проекту. Виберіть посилання Области рабочих элементов (рис. 5.2), що призведе до відкриття сторінки командного проекту.

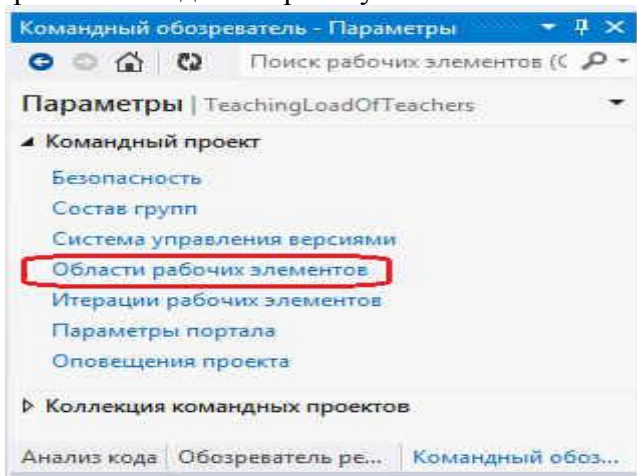


Рис. 5.2. Вкладка Параметри головної сторінки командного оглядача

На сторінці командного проекту буде відображена вкладка області (рис. 5.3), яка дозволяє формувати і встановлювати області відповідальності команди проекту. У навчальному проекті є тільки одна область (область за замовчуванням), з якою працює команда - TeachingLoadOfTeachers.

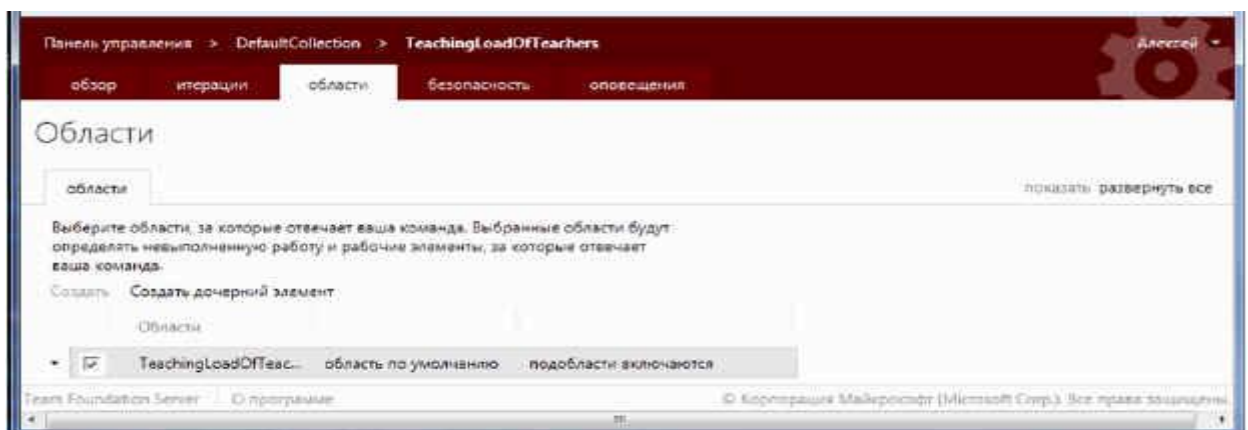


Рис. 5.3. Вкладка области сторінки проекту

На вкладці огляд представлена інформація, що описує профіль проекту і команду (рис. 5.4).

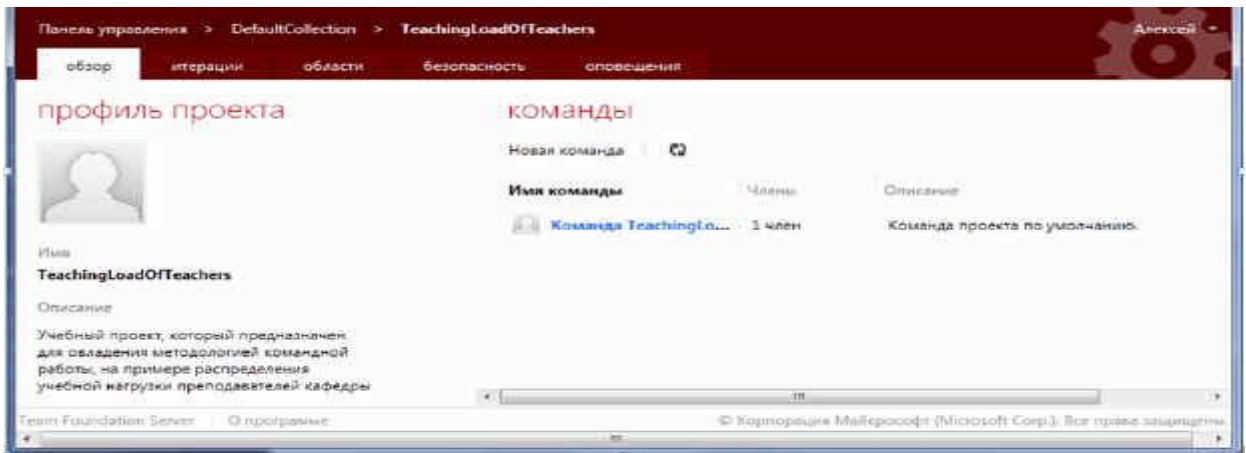


Рис. 5.4. Вкладка огляд сторінки проекту

Для профілю проекту наводиться ім'я та короткий опис. Крім того, для профілю проекту можна задати зображення. Для цього клацніть на поле зображення, виберіть графічний файл і натисніть кнопку Зберегти зміни (рис. 5.5).

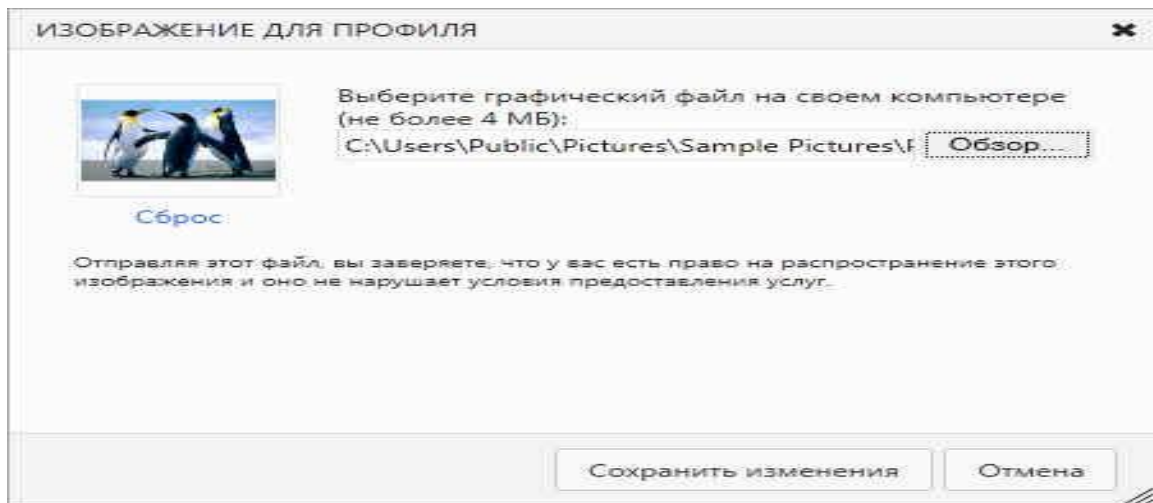


Рис. 5.5. Вибір зображення профілю проекту



Після введення зображення профілю проекту вкладка огляд сторінки командного проекту прийме вигляд, наведений на рис. 5.6.

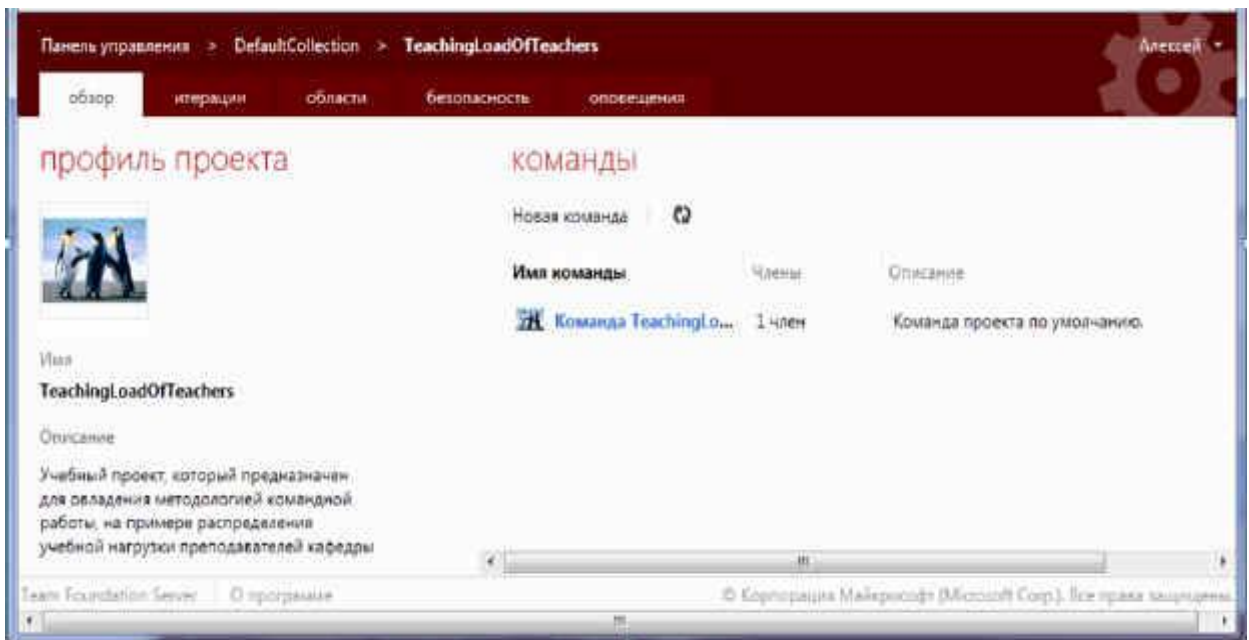


Рис. 5.6. Вкладка огляд сторінки проекту з заданим зображення профілю.

Для визначення параметрів ітерацій проекту перейдіть на вкладку ітерації сторінки проекту (рис. 5.7). У навчальному проекті для випуску 1 будуть використані три ітерації - спринту.

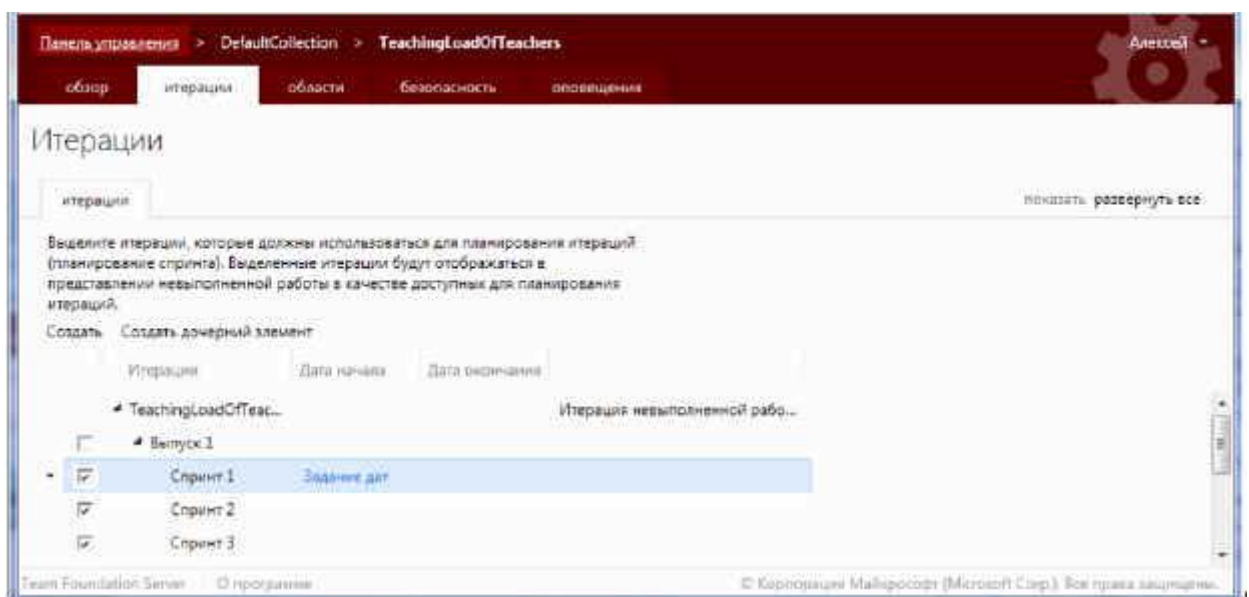


Рис. 5.7. Вкладка ітерації сторінки проекту

Завдання дат початку і закінчення спринтів проводиться при переході за посиланням Завдання дат. У діалоговому вікні ЗМІНІТЬ ітерацій встановіть дати початку і закінчення відповідного спринту (рис. 5.8).

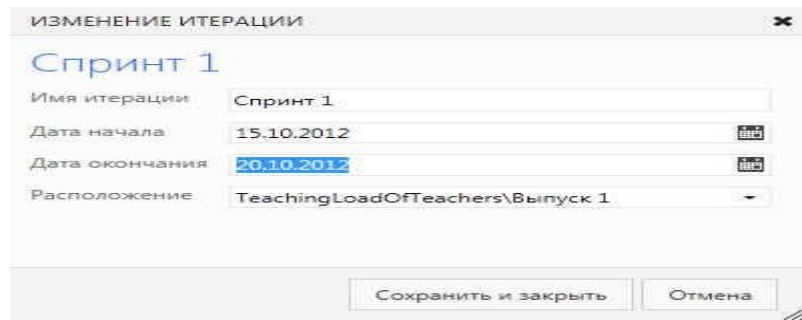


Рис. 5.8. Введення дати початку і закінчення спринту

Результат введення дат початку і закінчення спринтів наведено на рис. 5.9.

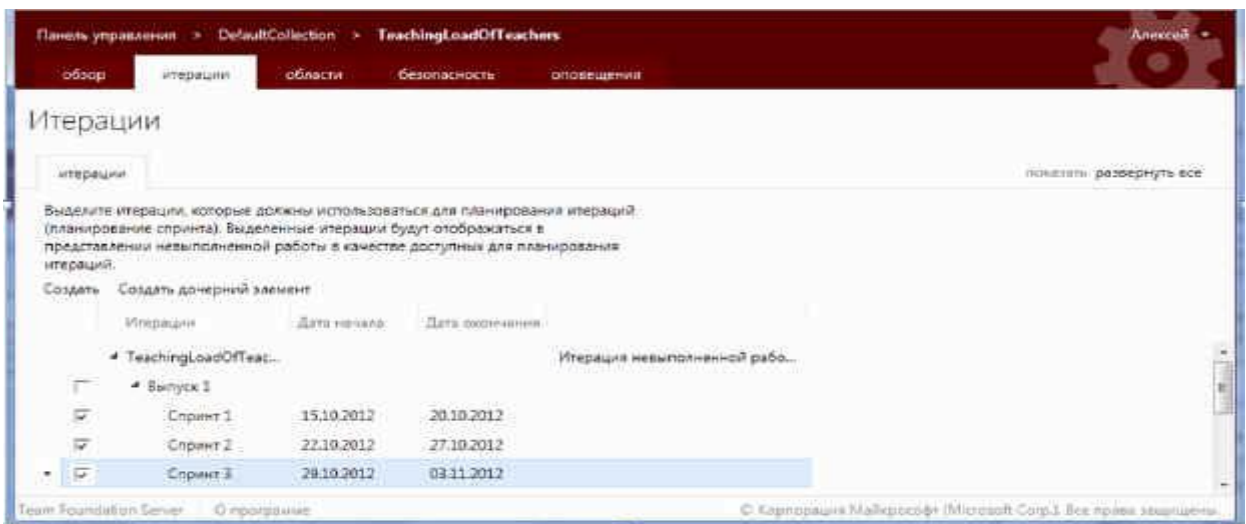


Рис. 5.9. Результат введення дат початку і закінчення спринтів

Налаштування параметрів команди.

Вкладка безпеки дозволяє задати параметри команд, груп TFS, членів команд і груп, а також сформувані для них дозволу (рис. 5.10).

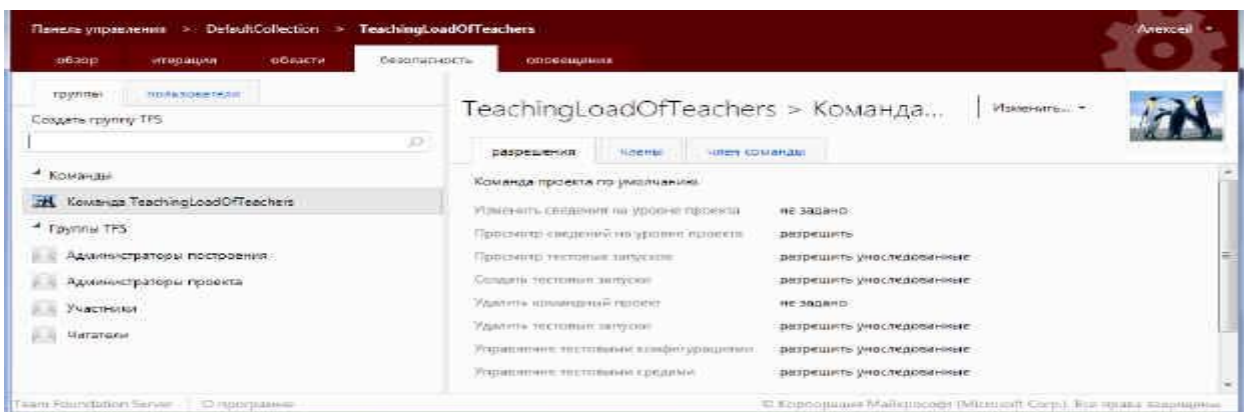


Рис. 5.10. Вкладка безпеки сторінки проекту

При створенні командного проекту членом команди є тільки той користувач, який створив проект, тобто адміністратор проекту (рис. 5.11). При додаванні користувачів в команду проекту на вкладці члени команди проекту в випадаючому списку Додати виберіть Додавання користувача або групи Windows (рис. 5.12).

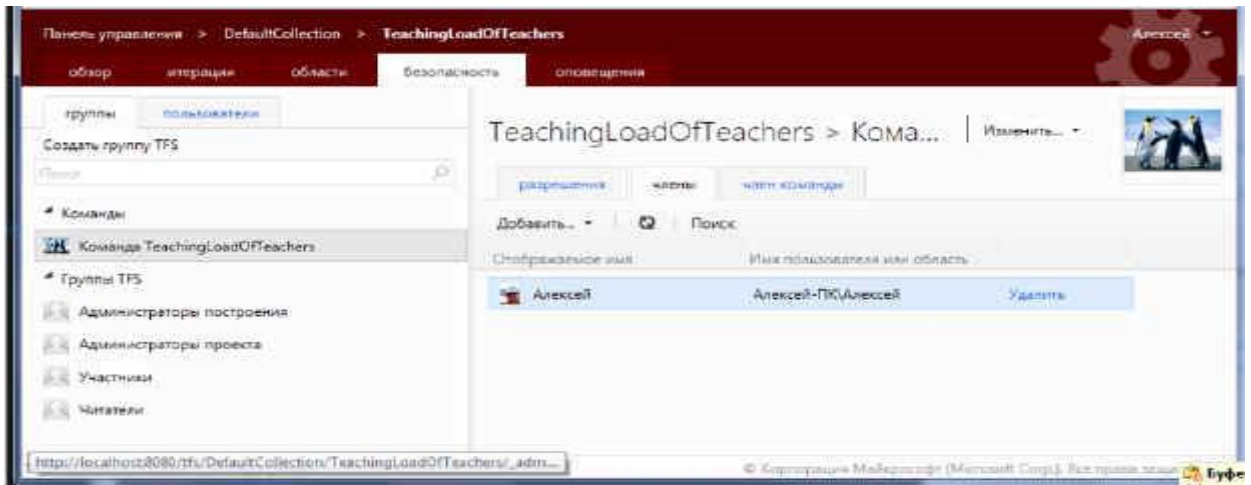


Рис. 5.11. Члени команди проекту

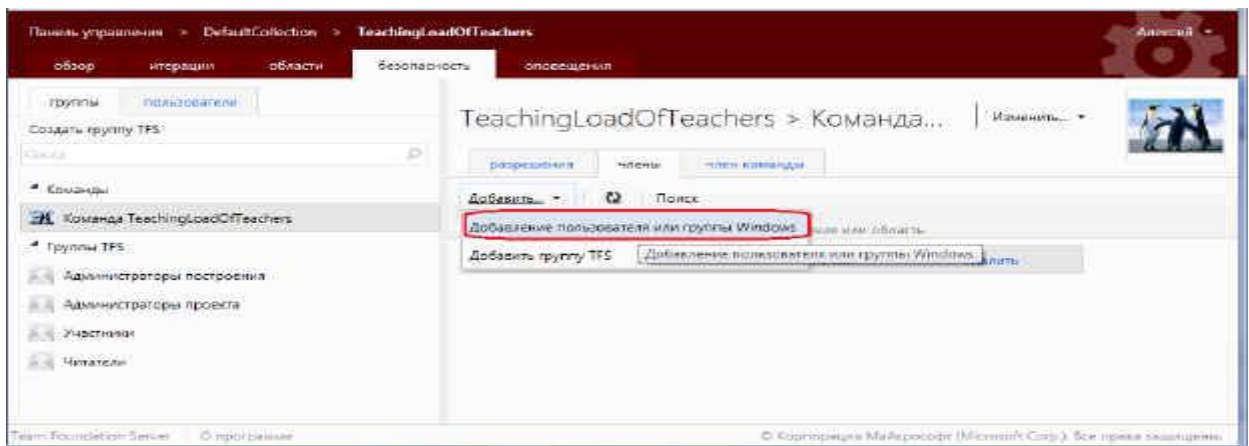


Рис. 5.12. Додавання користувача

У діалоговому вікні ДОДАТИ КОРИСТУВАЧА АБО ГРУПУ WINDOWS виберіть посилання огляд (рис. 5.13), відзначте галочками імена користувачів, що включаються в команду проекту (рис. 5.14) і натисніть кнопку Додати і потім Зберегти зміни.

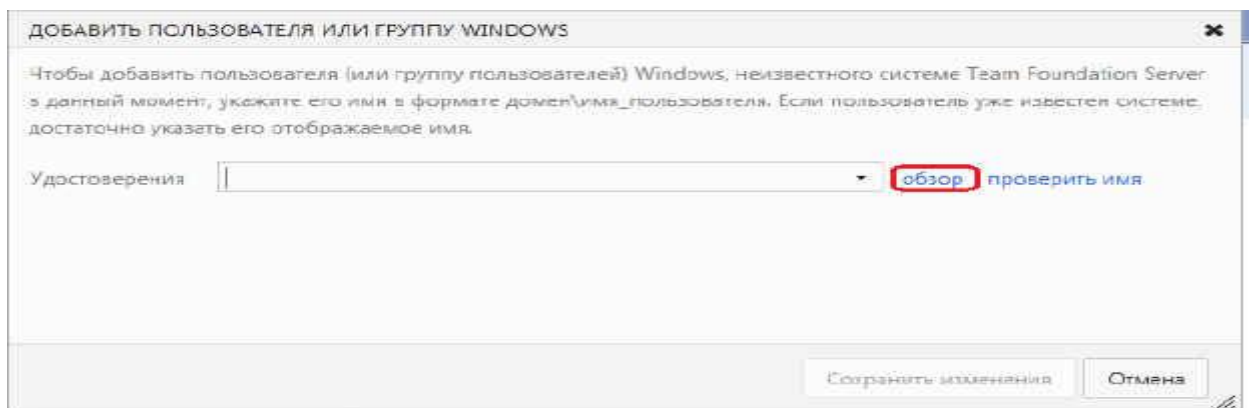


Рис. 5.13. Діалогове вікно ДОДАТИ КОРИСТУВАЧА АБО ГРУПУ WINDOWS



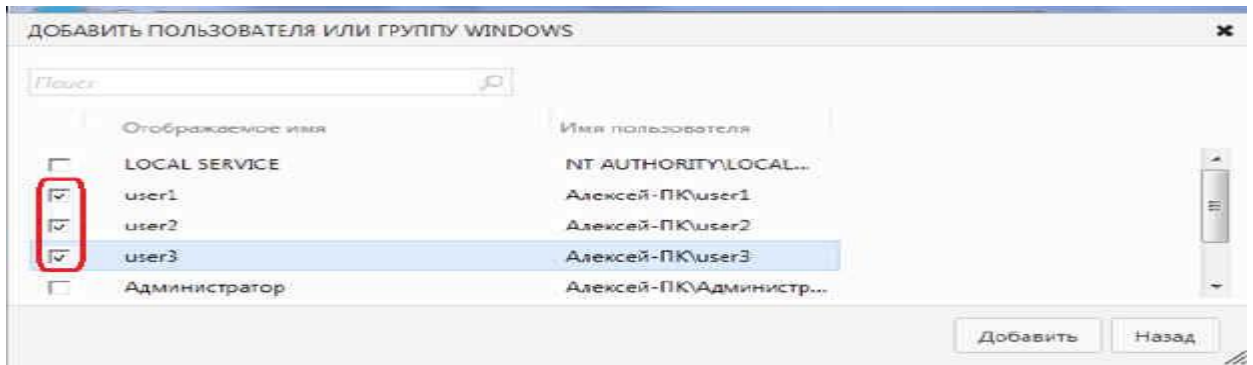


Рис. 5.14. Формування списку користувачів

У список членів команди проекту будуть додані нові користувачі (рис. 5.15).

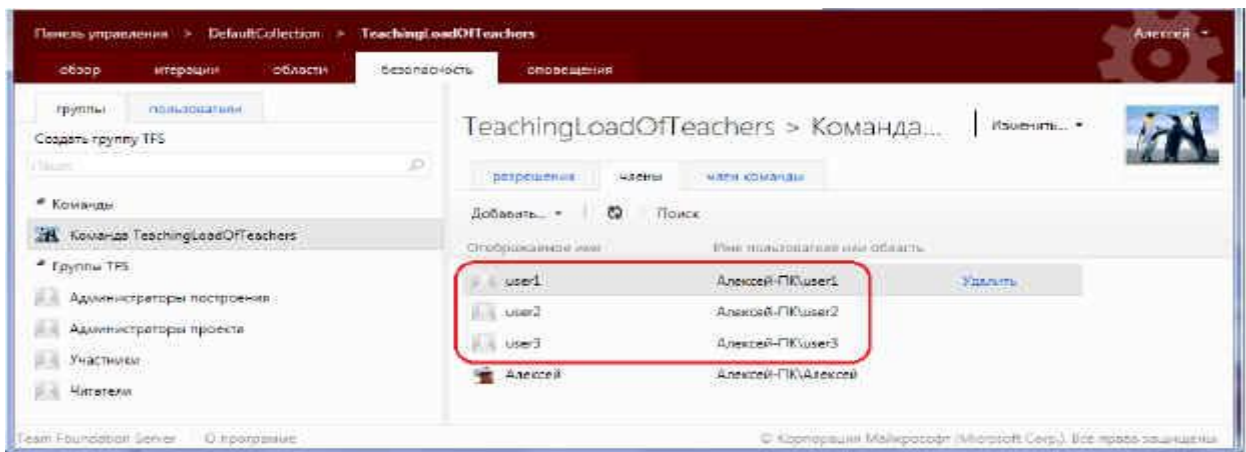


Рис. 5.15. Оновлений склад команди проекту

### Настройка сповіщень

Вкладка оповіщення дозволяє створити ряд оповіщень для членів команди при зміні деяких артефактів проекту. Припустимо, що потрібно проінформувати команду проекту при будь-якому поверненні коду в систему управління версіями. Для формування такого оповіщення перейдіть за посиланням Створити (рис. 5.16).

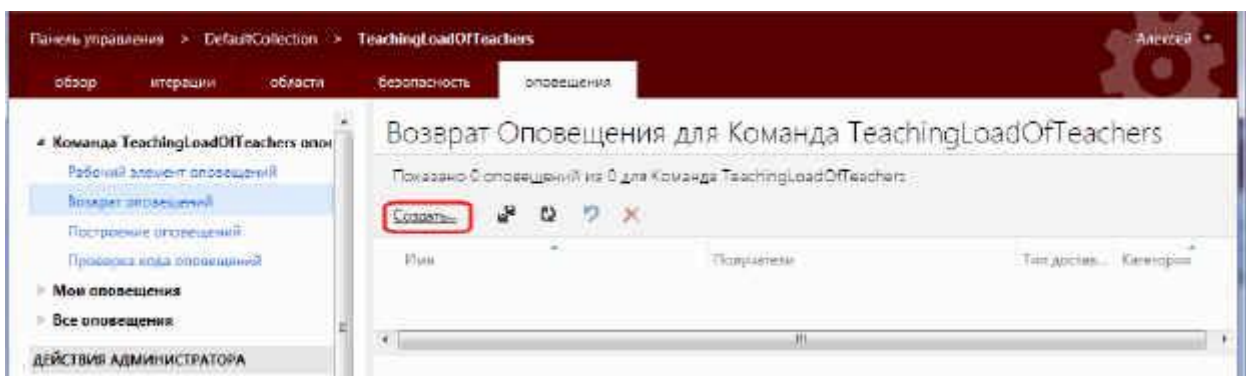


Рис. 5.16. Створення оповіщень

У діалоговому вікні Вибір шаблону НОВОГО ОПОВІЩЕННЯ задайте область і категорію оповіщення (рис. 5.17).

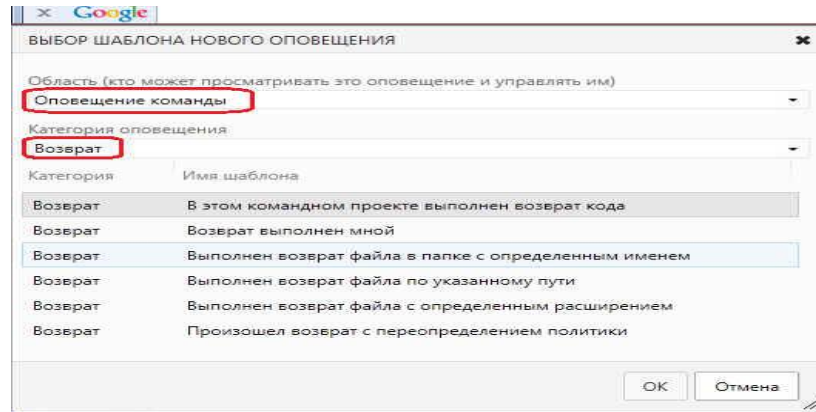


Рис. 5.17. Диалоговое окно выбора шаблона НОВОГО ОПОВЕЩЕНИЯ

На рис. 5.18 приведен сформованный запит на сповіщення при поверненні коду, а на рис. 5.19 - результат сформованого оповіщення.

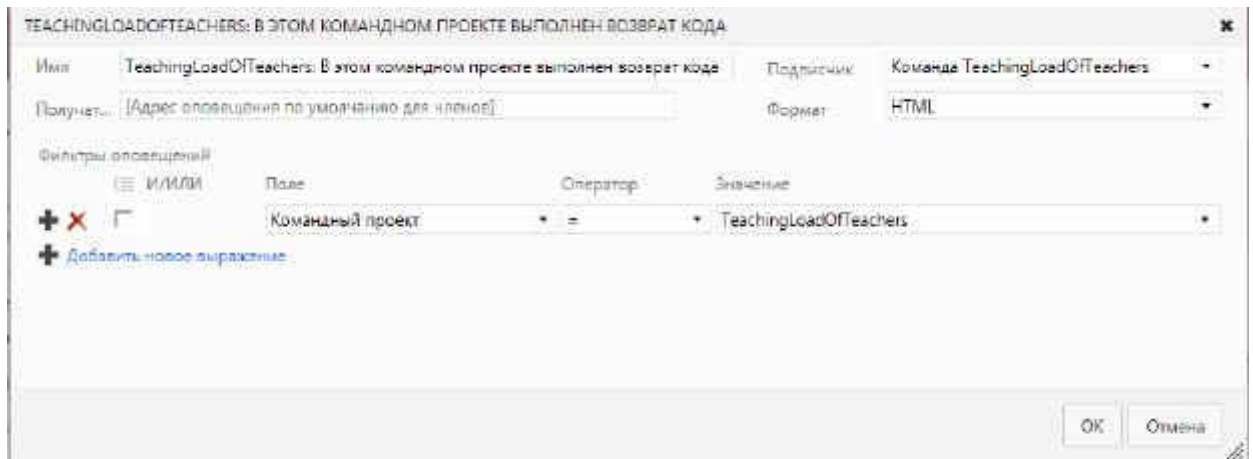


Рис. 5.18. Запит повернення коду

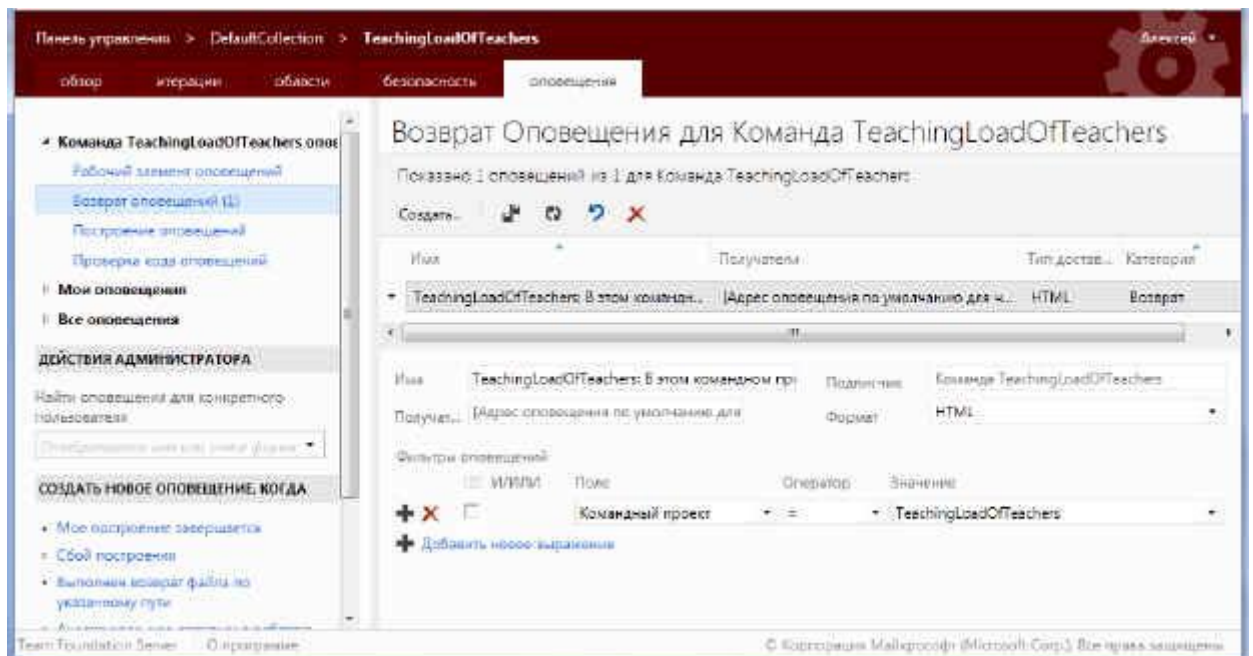


Рис. 5.19. Вкладка огляд сайту проекту із заданим зображення профілю

### Веб-доступ до параметрів проекту

Настройку параметрів проекту можна починати проводити не через командний оглядач Visual Studio, а через веб-сайт. Для цього відкрийте браузер і введіть URL-адресу проекту в наступному форматі: `http://ім 'сервера: порт / tfs /`. У навчальному прикладі заданий URL-адреса: `http://localhost:8080/tfs/`.

На домашній сторінці сайту TFS виводиться найменування останнього проекту і є можливість переглянути всі проекти, вибравши посилання Проекти, і в випадяючому меню вказати необхідний проект (рис. 5.20).

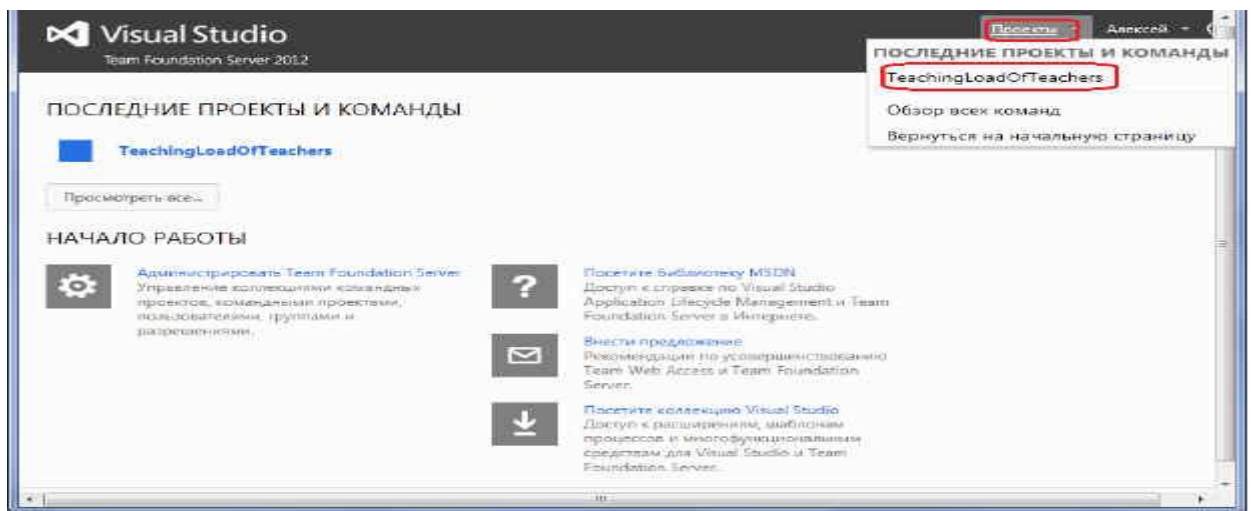


Рис. 5.20. Домашня сторінка сайту TFS

При переході на головну сторінку проекту (рис. 5.21) можна управляти членами команди проекту, вибравши посилання Керувати усіма членами (1), налаштовувати ітерації, вибравши посилання Налаштувати розклад і ітерації (2), а також налаштовувати робочу область, вибравши посилання Налаштувати робочі області (3).

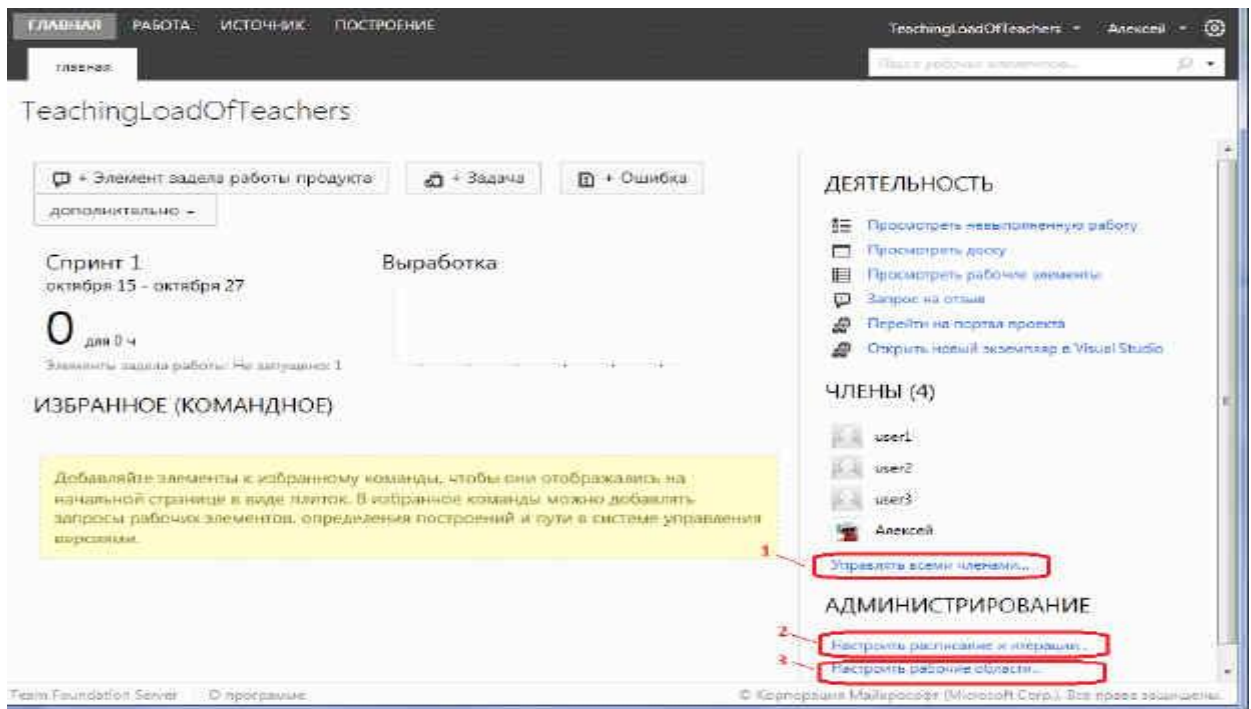


Рис. 5.21. Головна сторінка проекту TeachingLoadOfTeachers

При виборі посилання Керувати всіма членами надається можливість додавати і видаляти членів команди проекту в діалоговому вікні УПРАВЛІННЯ ЧЛЕНАМИ КОМАНДА TEACHINGLOADOFTEACHERS (рис. 5.22).

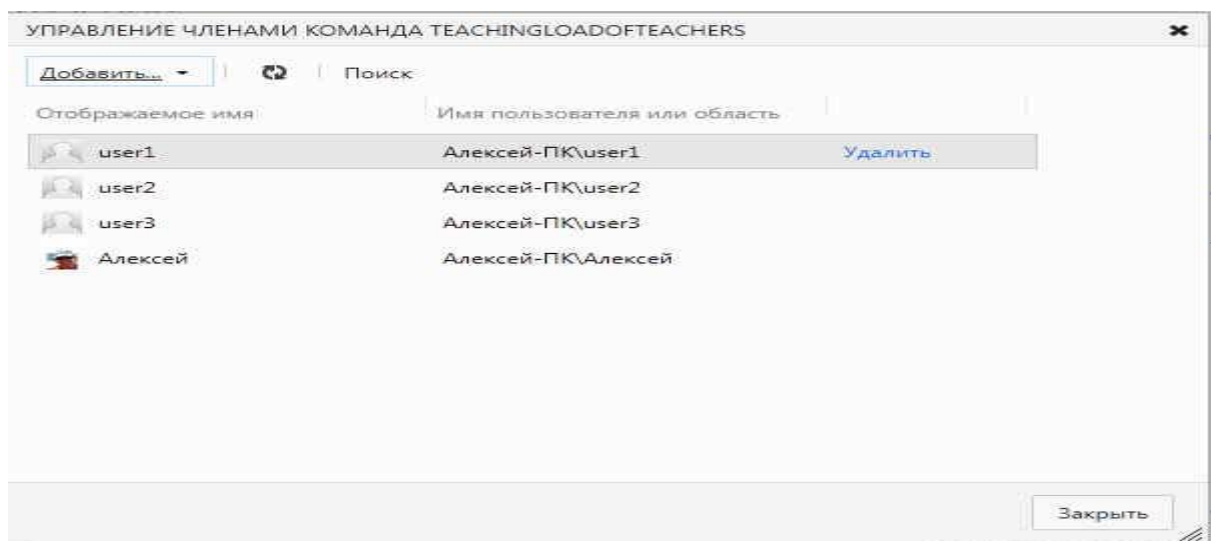


Рис. 5.22. Діалогове вікно УПРАВЛІННЯ ЧЛЕНАМИ КОМАНДА TEACHINGLOADOFTEACHERS

При виборі посилання Налаштувати розклад і ітерації надається можливість редагувати складу випусків, ітерацій і часу проведення ітерацій, створювати дочірні елементи (рис. 5.23).

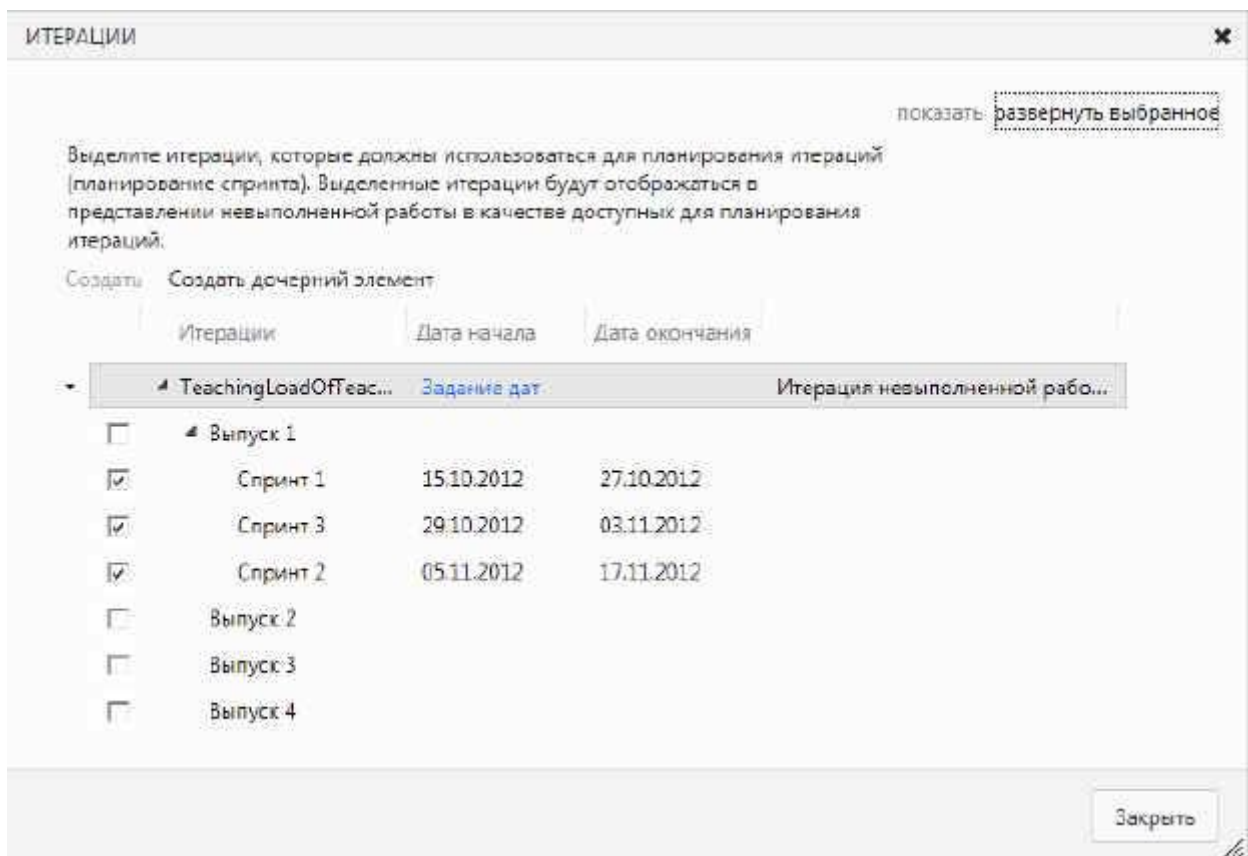


Рис. 5.23. Диалогове вікно ітерацій

При виборі посилання Налаштувати робочі області надається можливість редагувати робочі області проекту (рис. 5.24).

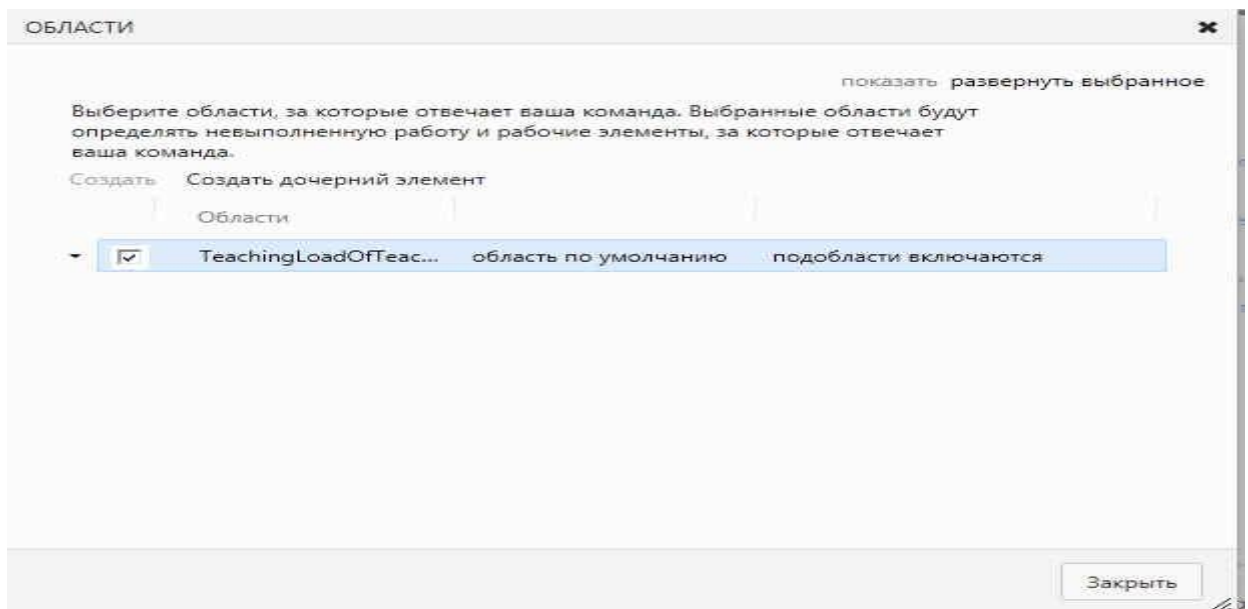


Рис. 5.24. Диалогове вікно ОБЛАСТІ



При проектуванні програмного продукту спочатку визначається мета його створення. Далі необхідно скласти перелік вимог користувача до програмного продукту. За цю роботу відповідає Власник продукту (Product owner). При використанні командою методології та шаблону Scrum користувальницькі вимоги формуються в робочому елементі Елемент заділ роботи продукту (ЕЗРП). ЕЗРП можна створити за допомогою Visual Studio, Team Web Access, Microsoft Excel і Microsoft Project. У даній лабораторній роботі розглянемо можливості Visual Studio і Team Web Access при формуванні вимог користувача.

Створення користувальницьких вимог за допомогою Visual Studio. Для формування користувальницький вимог у командному браузері Visual Studio клацніть на посиланні Створити робочий елемент і в випадаючому меню виберіть пункт Елемент заділ роботи продукту (рис. 7.1).

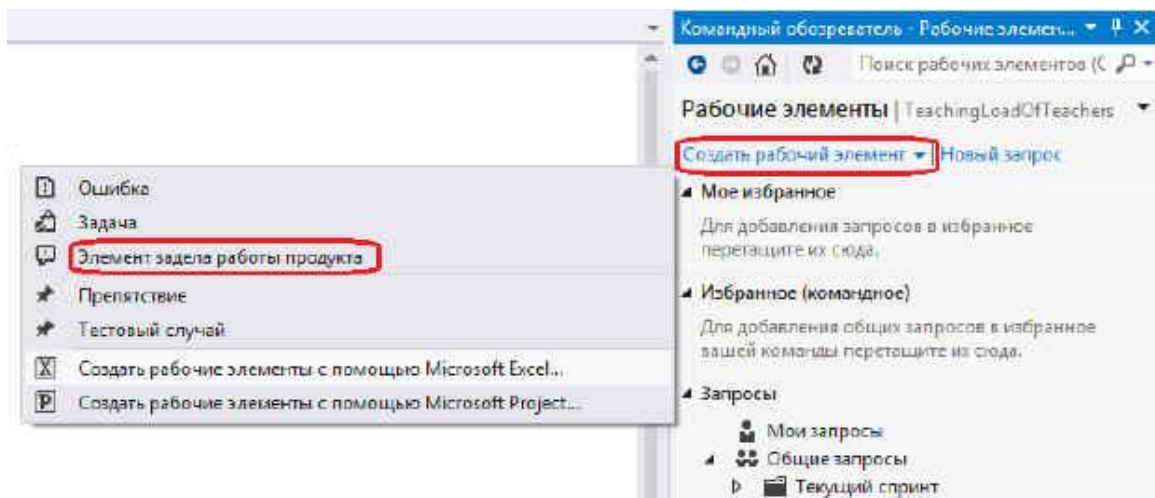


Рис. 7.1. Меню створення Елемента заділ роботи продукту

Для створення робочого елемента відкривається закладка з формою введення даних по Елементу заділ роботи продукту (рис. 7.2).

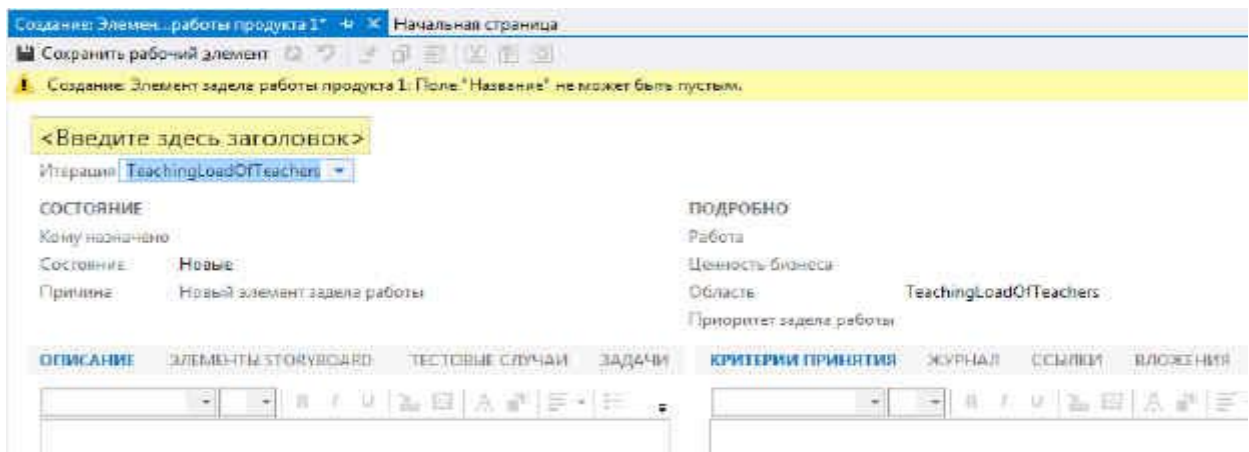


Рис. 7.2. Закладка Створення: Елемент заділ роботи продукту

На початковому етапі необхідно ввести ім'я вимоги (поле "Назва") і бажано привести короткий опис. На рис. 7.3 введено ім'я вимоги - Авторизація користувача (1), короткий опис вимоги (2). Після збереження робочого елемента інформація по ньому зберігається в базі даних TSF і дії члена команди фіксуються у журналі (3).

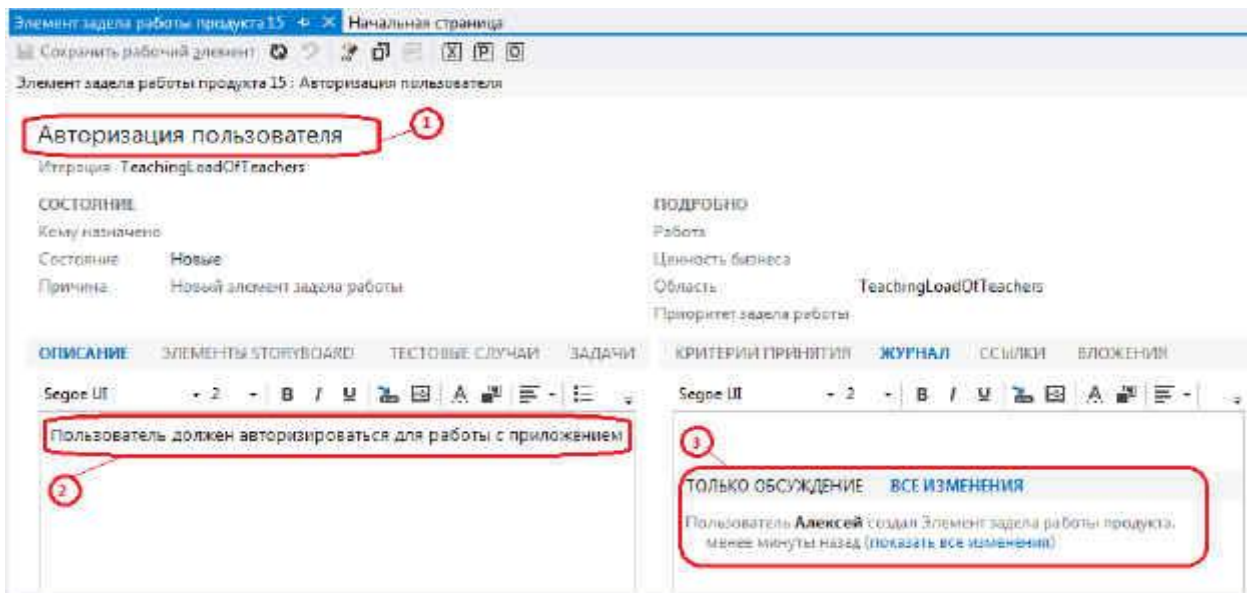


Рис. 7.3. Закладка Створення: Елемент заділ роботи продукту

Як видно на рис. 7.3 у графі Стан параметру Стан встановлено значення за умовчанням Нові, Причина - Новий елемент заділ роботи, параметру Кому призначено не задано значення. У графі ПОДРОБНО вказана тільки область команди - TeachingLoadOfTeachers, а іншим параметрам Робота, Цінність бізнесу та Пріоритет заділ роботи значення не встановлені. Чи не визначені на даному етапі параметри робочого елемента будуть задані пізніше при формуванні робіт зі спринту.

Введемо дані ще по одному користувальницькому вимогу - Формування даних по викладачам (рис. 7.4).

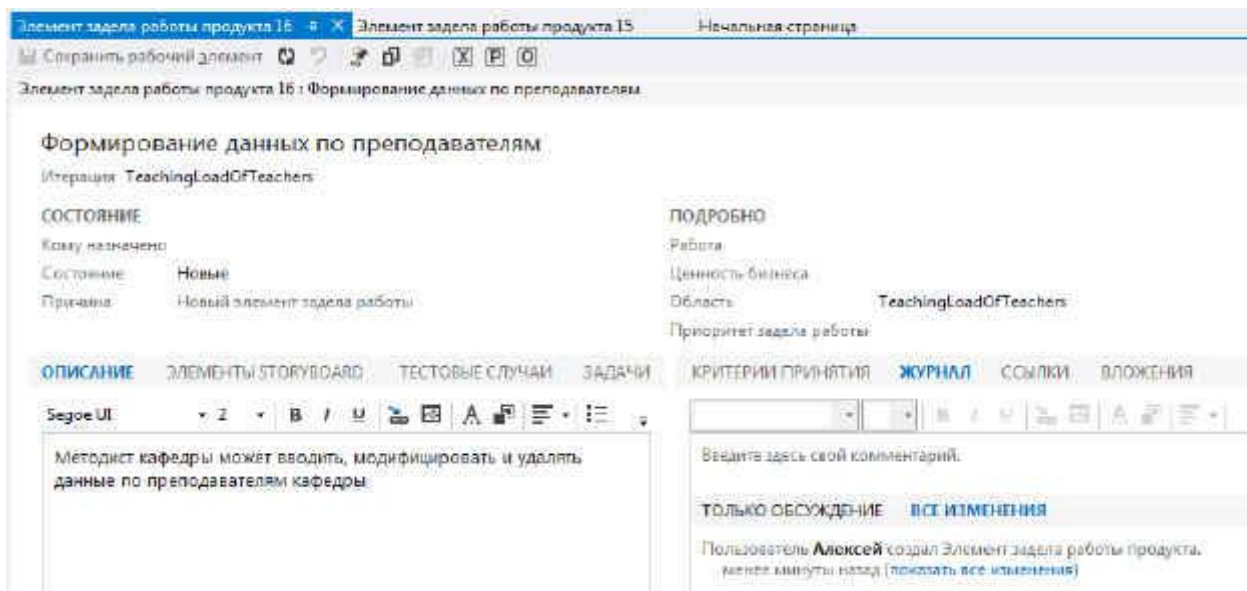


Рис. 7.4. Створення робочого елемента Формування даних по викладачам

Аналогічним чином можна ввести всі відомі на поточний момент користувальницькі вимоги. Користувальницькі вимоги - ЕЗРП формуються в список вимог - невиконані роботи по продукту. Для перегляду списку вимог в командному браузері виберіть посилання невиконані роботи по продукту (рис. 7.5).

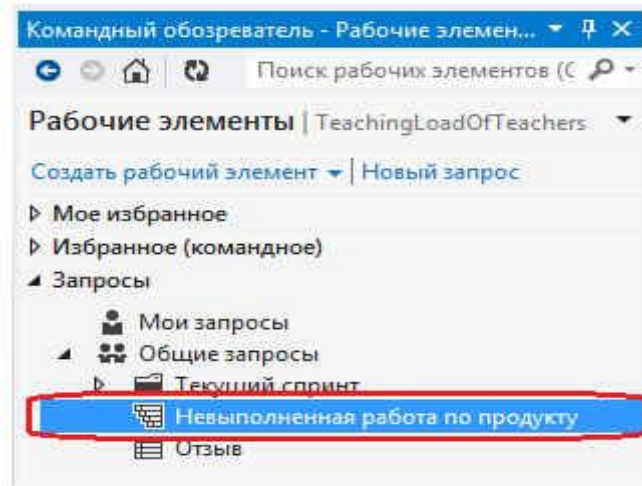


Рис. 7.5. Виклик компонента невиконані роботи по продукту

В результаті виконання запиту буде виведений сформований список користувача вимог до програмної системи (рис. 7.6).

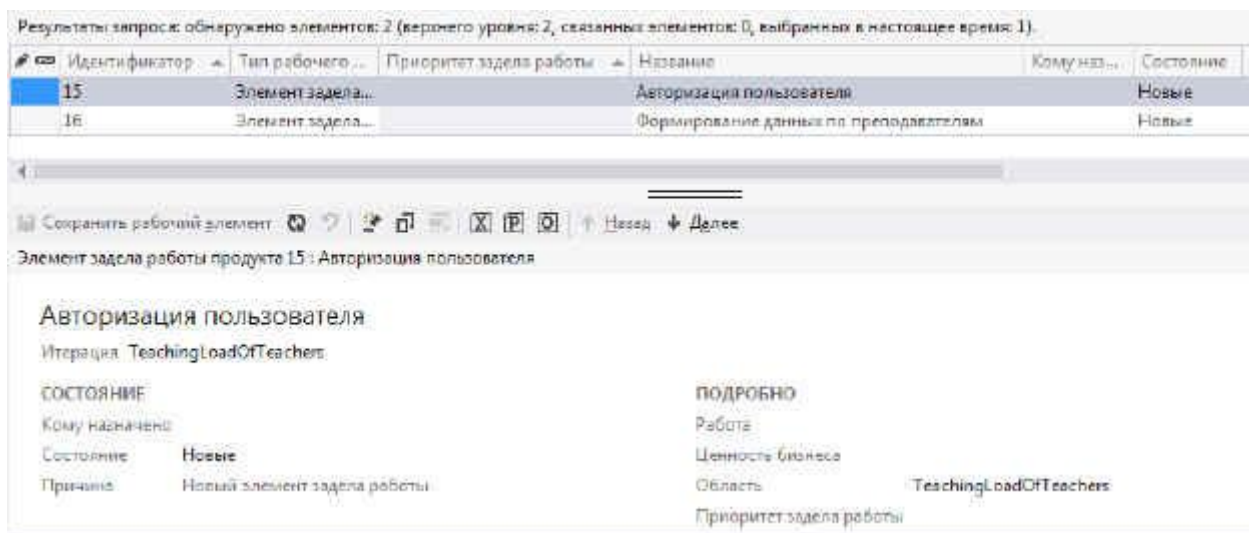


Рис. 7.6. Рабочий элемент невиконані роботи по продукту

Створення користувальницьких вимог за допомогою Team Web Access. При використанні Team Web Access для формування вимог користувача відкрийте сторінку командного проекту (<http://localhost:8080/tfs/DefaultCollection/TeachingLoadOfTeachers>), яка представлена на рис. 7.7, і клацніть на кнопці Елемент заділ роботи продукту.



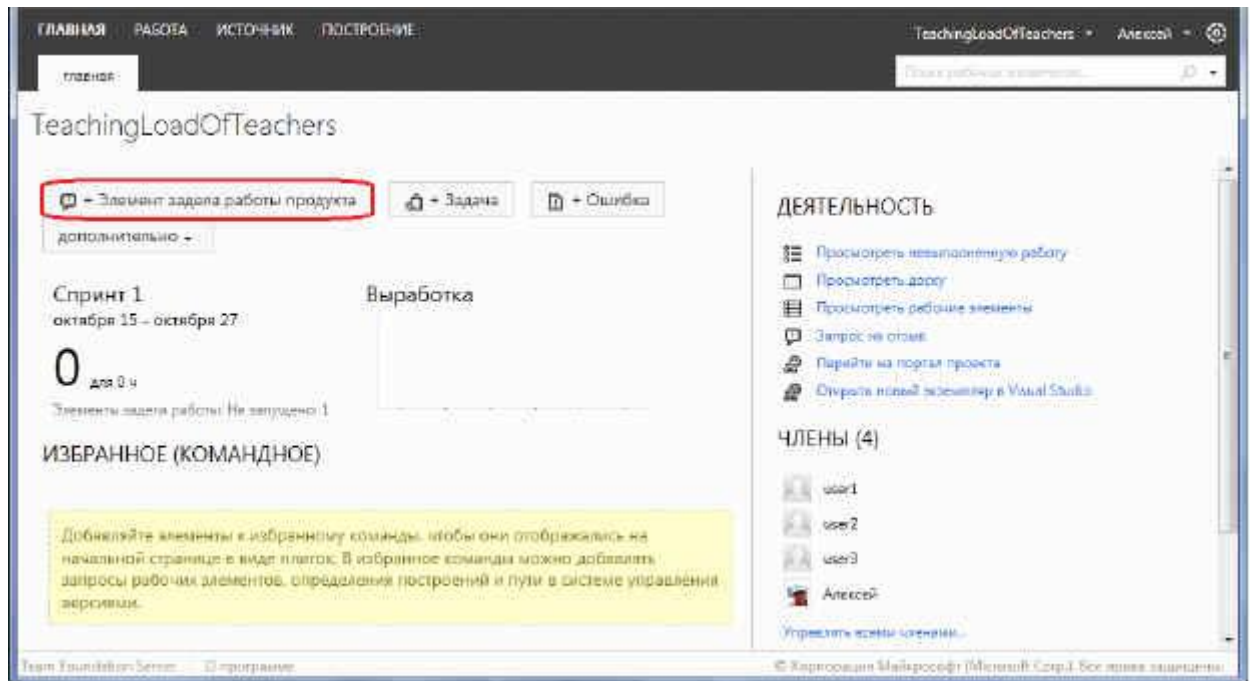


Рис. 7.7. Головна сторінка командного проекту

У діалоговому вікні Елемент заділ роботи продукту введіть найменування вимоги та короткий опис (рис. 7.8). По закінченню введення даних натисніть кнопку Зберегти й закрити.

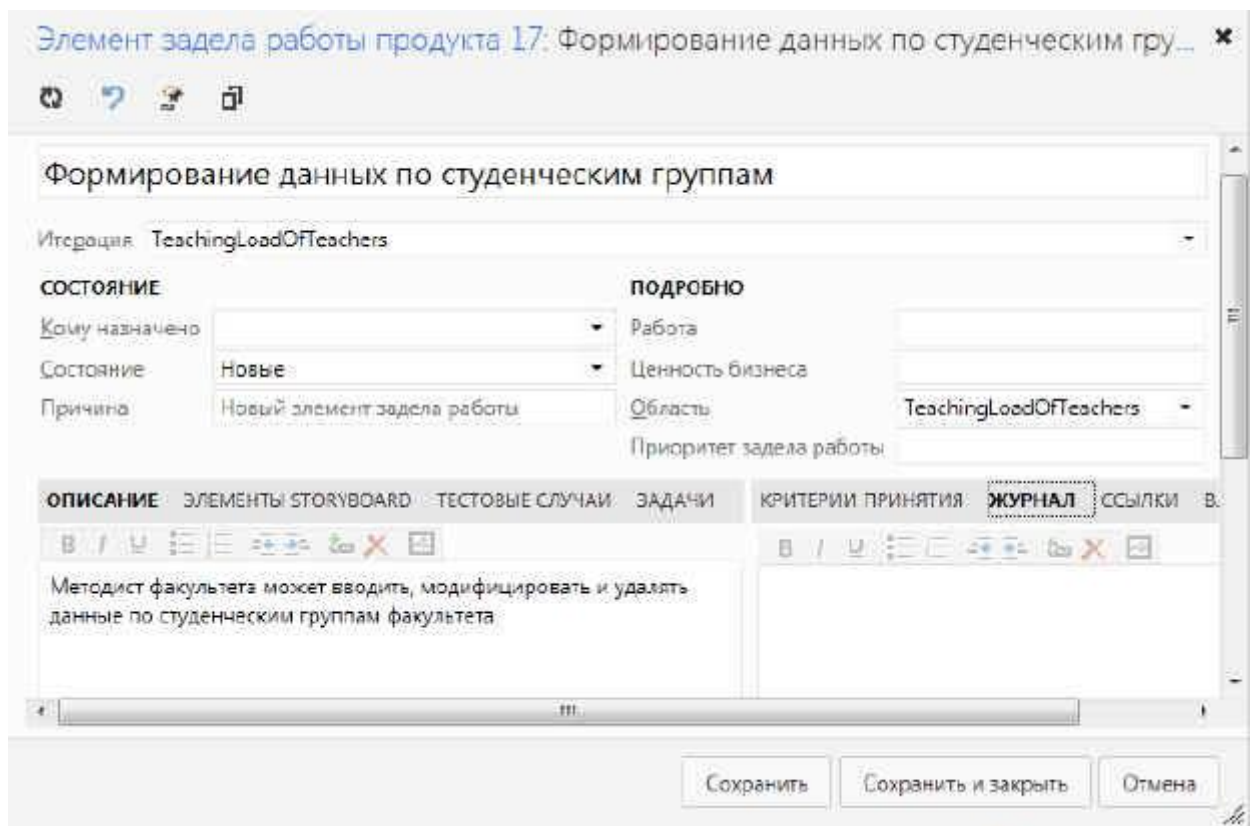


Рис. 7.8. Діалогове вікно Створення: Елемент заділ роботи продукту

Для перегляду сформованого списку користувача вимог на головній сторінці проекту виберіть посилання Переглянути невиконану роботу (рис. 7.9).

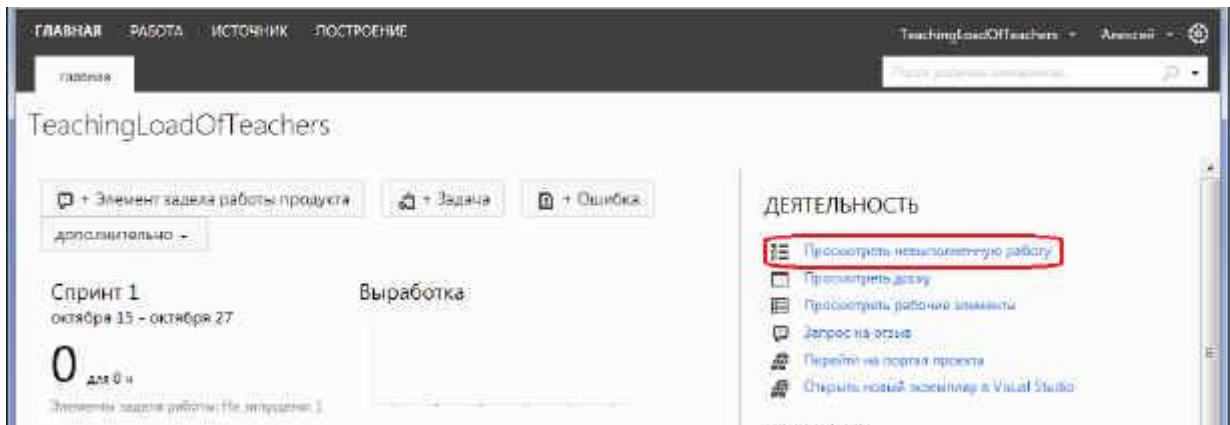


Рис. 7.9. Перехід на список користувача вимог

Список користувача вимог відображається на вкладці невыполнена работа (рис. 7.10). Дана вкладка може використовуватися для створення нових вимог користувача. Для додавання нового елемента заділ роботи продукту натисніть кнопку Додати.

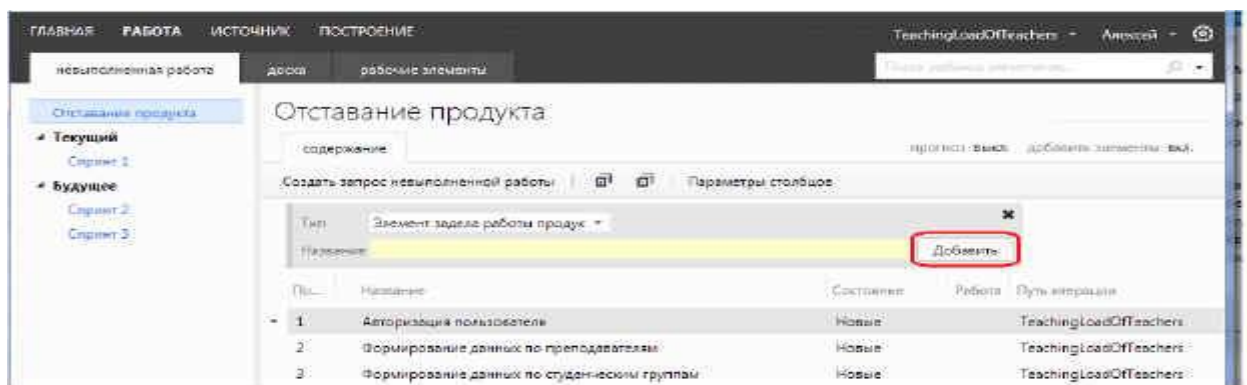


Рис. 7.10. Вкладка невыполнена работа

Користувача вимога формується в діалоговому вікні Елемент заділ роботи продукту (рис. 7.11).

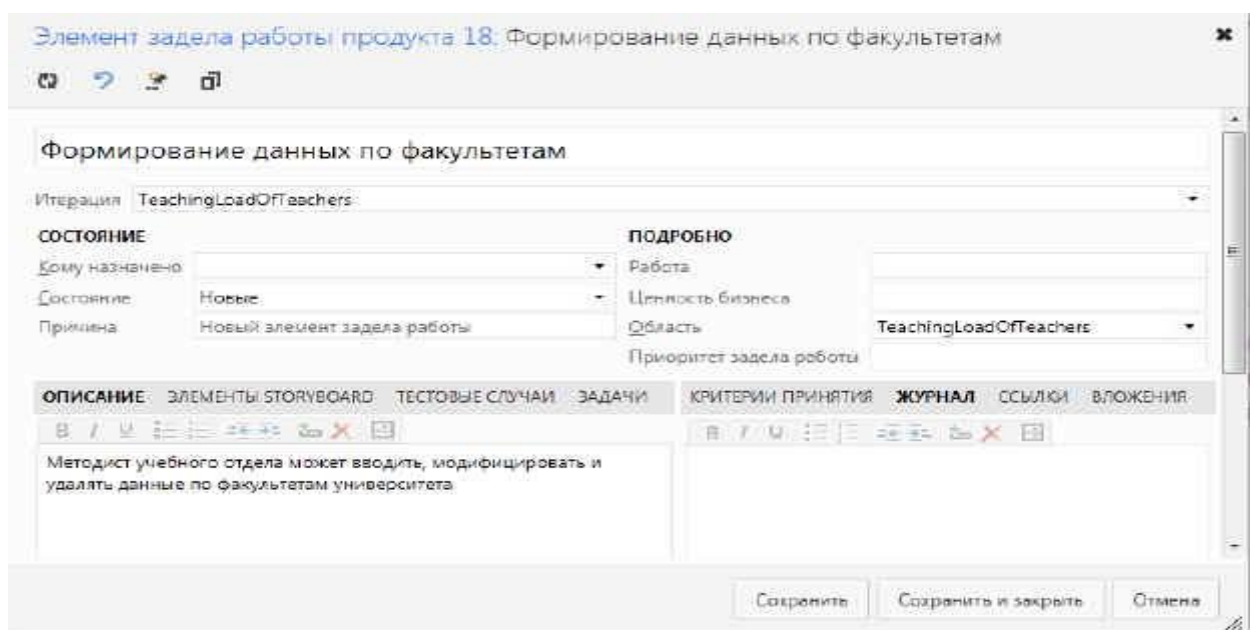


Рис. 7.11. Діалогове вікно Створення: Елемент заділ роботи продукту

Після збереження нового Елемента заділ роботи продукту він з'являється у списку Відставання продукту на закладці зміст (рис. 7.12).

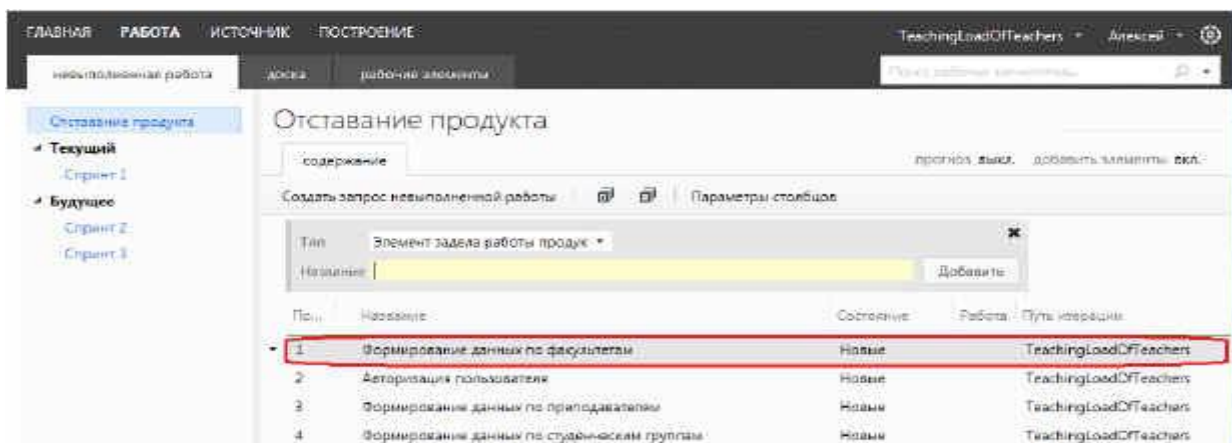


Рис. 7.12. Оновлена вкладка невиконана работа

Створені робочі елементи можна переглянути вибравши посилання Переглянути робочі елементи на головній сторінці проекту (рис. 7.13).



Рис. 7.13. Перехід на перегляд робочих елементів

На вкладці робочі елементи відображається список створених користувача вимог (рис. 7.14). На даній вкладці можна створити новий робочий елемент, клацнувши на посиланні Створення і вибравши зі списку необхідний тип робочого елемента, наприклад Елемент заділ робіт продукту.

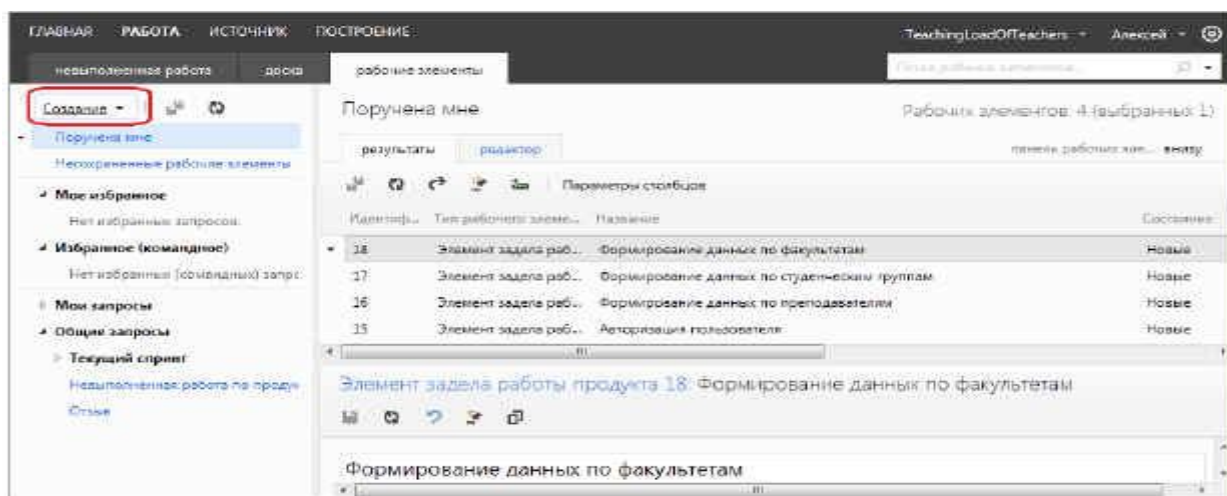


Рис. 7.14. Вкладка робочі елементи

## **Завдання**

1. Ознакомитесь с теоретическим материалом. По завданню викладача створити командний проект. Налаштувати параметри створеного командного проекту. параметри командного проекту.

Звіт з лабораторної роботи повинен складатися з:

1. Постановки завдання.
2. Документу «Протокол лабораторної роботи», що містить:
  - Назву роботи. Мету;
  - Хід виконання роботи (графічні результати виконання роботи);
  - Відповіді на контрольні питання;
  - Висновки.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.

## **Контрольні питання**

1. Назвіть основні принципи MSF.
2. Чим визначається універсальність моделі MSF?
3. Яка модель циклу програмної системи використовується в MSF?
4. У чому полягає Ітеративність методології MSF?
5. Поясніть призначення інтеграції в методології MSF?
6. Як можна масштабувати команду, що використовує методологію MSF?

## **Лабораторна робота №6**

### ***Розробка колективного проекту клієнт-серверного застосування для роботи з базами даних за індивідуальним завданнями.***

**Мета:** Отримати практичні навички в розробці користувальницьких вимог до програмного продукту за допомогою Visual Studio, Team Web Access.

### **Теоретичні відомості**

#### **Розробка застосування**

Основним засобом розробки в VisualStudio2012 є інтегрована середовище розробки (IDE). IDE-середовище інтегрована із засобами модульного тестування та забезпечує можливості виявлення неефективного, небезпечного чи погано написаного коду, управління змінами і модульне тестування як коду, так і бази даних.

Важливим інструментом розробника програмного забезпечення є модульне тестування, яке реалізується в середовищі UnitTestFramework. Призначенням модульних тестів є перевірка того, що код працює правильно з точки зору програміста. Модульні тести формуються на більш низькому рівні, ніж інші види тестування, і перевіряють чи працюють лежать в їх основі функції так, як очікується. Для модульного тестування використовується метод прозорого ящика, для якого потрібно знання внутрішніх структур коду.

Модульні тести допомагають виявити проблеми проектування та реалізації. Крім того, модульний тест є хорошою документацією з використання проекрованої системи. Хоча модульне тестування вимагає додаткового програмування, але його застосування окупається за рахунок скорочення витрат на налагодження програми.

Модульні тести є важливим елементом регресійного тестування. Регресійне тестування являє собою повторне тестування частини програми після внесення до неї змін або доповнень. Мета регресійного тестування - виявлення помилок, які можуть з'явитися при внесенні змін до програми

У VisualStudio 2012 є функція "Аналіз покриття коду", яка проводить моніторинг того, які рядки коду виконувалися в ході модульного тестування. Результатом аналізу покриття коду є виявлення областей коду, які не покриті тестами.

Важливим аспектом створення якісного програмного продукту є дотримання розробниками правил і стандартів організації в написання коду. У VisualStudio 2012 є функції аналізу коду, які дозволяють проаналізувати код, знайти типові помилки, порушення стандартів і запропонувати заходи щодо усунення помилок і порушень. Набори правил аналізу коду поставляються з VisualStudio 2012. Розробники можуть налаштувати свої проекти на певний набір правил, а також додати свої специфічні правила аналізу коду.

У процесі аналізу коду використовуються метрики коду, які дають кількісні оцінки різних характеристик коду. Метрики дозволяють визначити складність коду і його ізольовані області, які можуть призвести до проблем при супроводі програми. У VisualStudio 2012 використовуються наступні метрики коду:

- складність організації циклів - визначає число різних шляхів коду;
- глибина спадкування - визначає число рівнів в ієрархії спадкоємства об'єктів;
- об'єднання класів - визначає число класів, на які є посилання;
- рядки коду - визначає кількість рядків коду в виконуваному методі;
- індекс зручності підтримки - оцінює простоту обслуговування коду.

Для аналізу продуктивності та ефективності використання ресурсів додатком у VisualStudio 2012 є інструменти профілювання. Профілювання являє собою процес спостереження і запису показників про поведінку програми. Інструментарій профілювання (профілювальники) дозволяють виявити у додатку проблеми з продуктивністю. Такі проблеми, як правило, пов'язані з кодом, який виконується повільно, неефективно або надмірно використовує системну пам'ять. Профілювання зазвичай використовується для виявлення ділянок коду, які в ході виконання програми виконуються часто або довгий час.

Профілювальники бувають з вибіркою та інструментуванням. Профілювальники з вибіркою роблять періодичні знімки виконуються додатки і записують його стан. Профілювальники з інструментуванням додають маркери відстеження в початок і кінець кожної досліджуваної функції. У процесі роботи Профілювальники маркери активізуються, коли потік виконання програми входить у досліджувані функції і виходить з них. Профілювальник записує дані про програму і про те, які маркери були порушені в ході виконання програми. У VisualStudio 2012 підтримується профілювання з вибіркою і з інструментуванням. Для аналізу продуктивності необхідно:

- створити сеанс новий сеанс продуктивності;
- за допомогою Обозревателя продуктивності задати властивості сеанси;
- запустити сеанс, виконуючи програму та профілювальник;
- проаналізувати дані в звітах по продуктивності.

Більшості корпоративних додатків працює з базами даних, що визначає необхідність розробки та тестування додатків спільно з базами даних командою проекту. У VisualStudio 2012 є інструментарій створення баз даних і розгортання змін до них. Для цього використовується автономна розробка схем баз даних, яка дозволяє вносити зміни до схеми без підключення до виробничій базі даних. Після внесення змін в середовище розробки VisualStudio 2012 дозволяє протестувати їх в самому середовищі розробки та / або виділеної середовищі тестування. Крім того, VisualStudio 2012 дозволяє згенерувати псевдореальном дані для проведення тестів. При позитивних результатах тестування VisualStudio 2012 дозволяє згенерувати сценарії для оновлення виробничої бази даних. Цикл розробки бази даних програми складається з наступних кроків:

- переклад схеми бази даних в автономний режим;
- ітеративна розробка додатки з базою даних;
- тестування схеми бази даних;
- побудова і розгортання бази даних і додатки.

Для вдосконалення процесу налагодження додатків в VisualStudio 2012 є функція інтелектуального відстеження роботи програми IntelliTrace. Функція IntelliTrace конфігурується за допомогою наступних розділів:

- Загальні (General);
- Додатково (Advanced);
- Події IntelliTrace (IntelliTrace Events);
- Модулі (Modules).

Розділ Загальні дозволяє включити і відключити функцію IntelliTrace і задати запис тільки подій або додаткової інформації, що включає події, дані діагностики, виклики і відстеження на рівні методів. У розділі Додатково задається розташування для генерованого файлу журналу та його максимальний розмір. У розділі Події IntelliTrace перераховуються всі події діагностики, які будуть збиратися в ході налагодження програми. Розділ Модулі визначає список модулів, для яких необхідно збирати дані в процесі налагодження програми. При записи подій, що відбуваються в додатку, відбувається їх перехоплення при роботі програми та інформація про події фіксується в журналі. При налагодженні з IntelliTrace можна призупинити інтерактивний сеанс налагодження і переглянути події або виклики. Також є можливість зупинення виконання



програми та покрокове рух назад і вперед в інтерактивному сеансі налагодження, а також відтворення записаного сеансу налагодження.

## Виконання роботи

Створення користувальницьких вимог за допомогою Microsoft Excel В якості клієнтського інтерфейсу для Team Foundation Server можна використовувати Microsoft Excel і Microsoft Project (рис. 8.1).

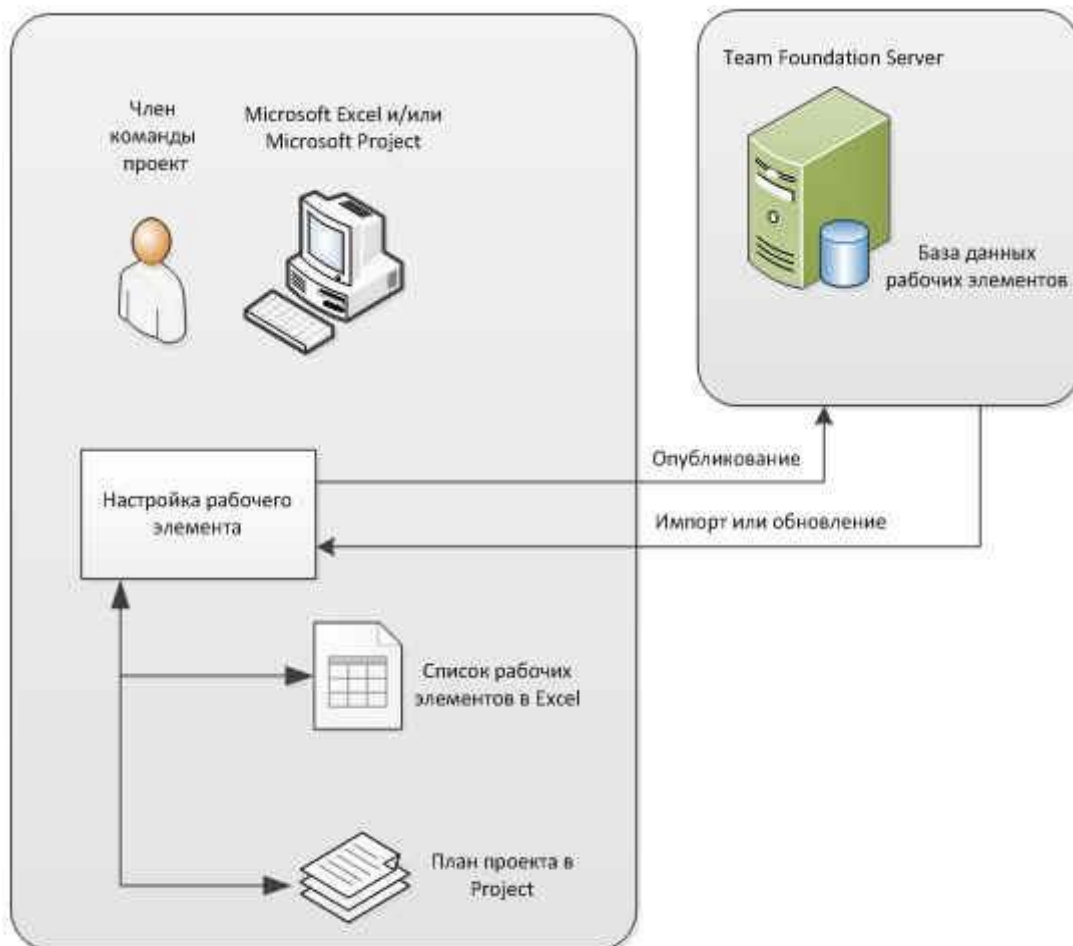


Рис. 8.1. Клієнти Microsoft Office для TFS

Microsoft Excel і Microsoft Project можуть взаємодіяти з Team Foundation Server і працювати як в режимі підключення до сервера, так і локально. При імпорті робочих елементів в документи Microsoft Excel або Microsoft Project, створюються їх локальні копії. Користувач може в автономному режимі обробляти робочі елементи проекту, а для синхронізації з базою даних TFS необхідно опублікувати дані з таблиць Microsoft Excel або документів Microsoft Project. При виконанні публікації документ, що включає зміни користувача, передається в базу даних робочих елементів TFS. Якщо користувачеві необхідно переглянути останні зміни в базі даних, то необхідно провести оновлення документів. При виконанні поновлення документ оновлюється у відповідності з поточними значеннями робочих елементів в базі даних TFS. Для використання Microsoft Excel запустіть цей додаток. Відкрийте вкладку Робоча група і натисніть кнопку Новий список (рис. 8.2)

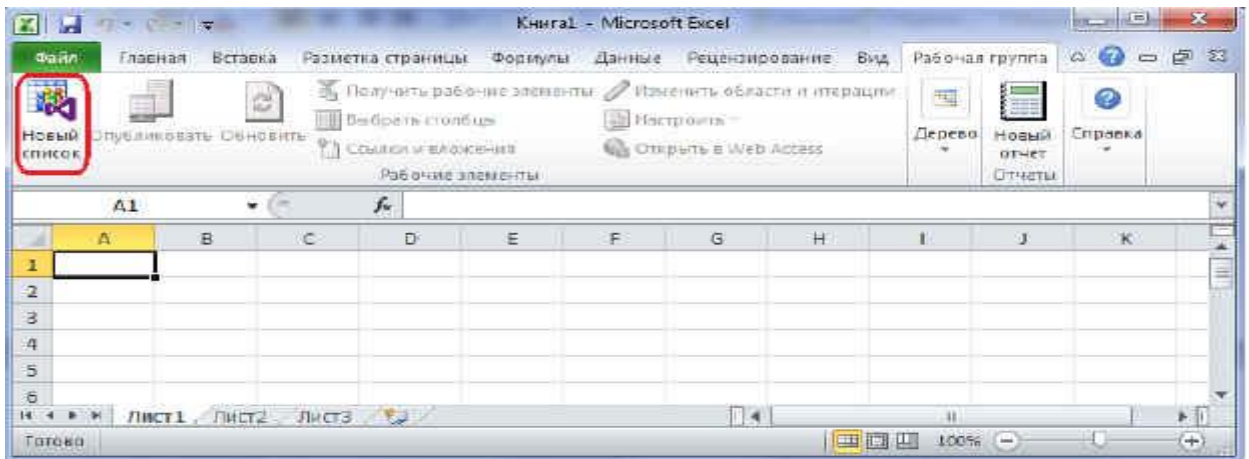


Рис. 8.2. Книга Microsoft Excel

У діалоговому вікні Підключення до Team Foundation Server виберіть потрібний сервер (на рис. 8.3 це сервер TeachingLoadOfTeachers) і натисніть кнопку Підключитися.

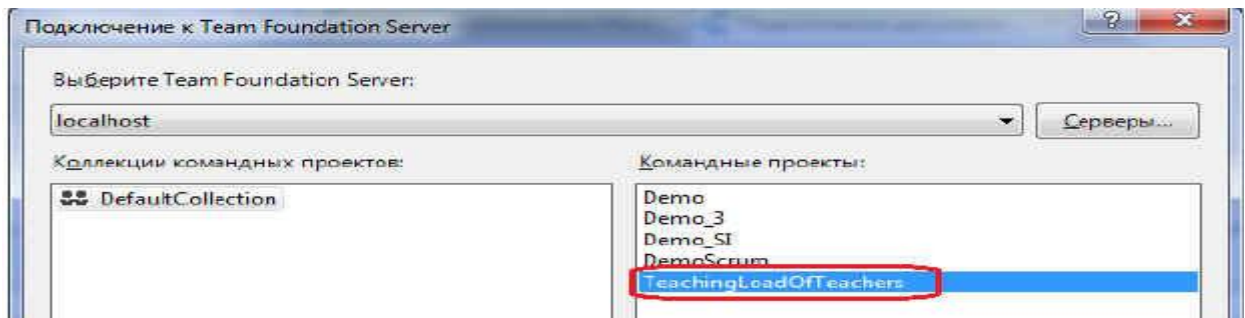


Рис. 8.3. Підключення до TFS

У діалоговому вікні Новий список включіть перемикач перед пунктом Список запитів і натисніть кнопку (рис. 8.4).

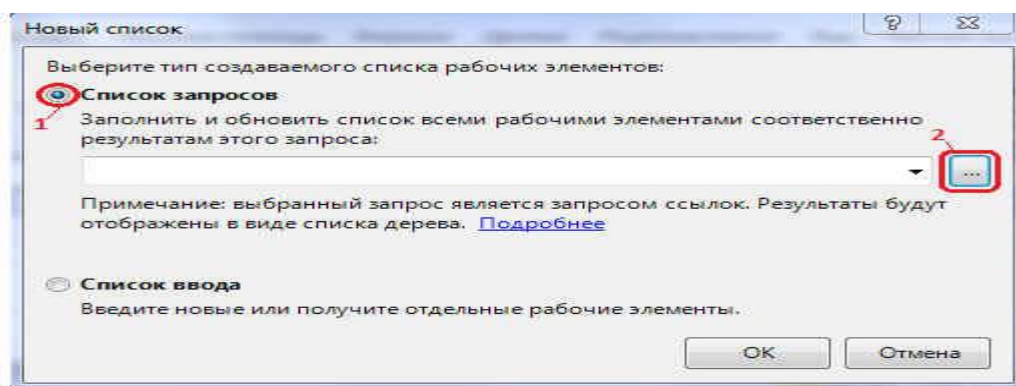


Рис. 8.4. Діалогове вікно Новий список

У діалоговому вікні Вибрати запит виберіть пункт невиконані роботи по продукту і двічі натисніть кнопку ОК (рис. 8.5).



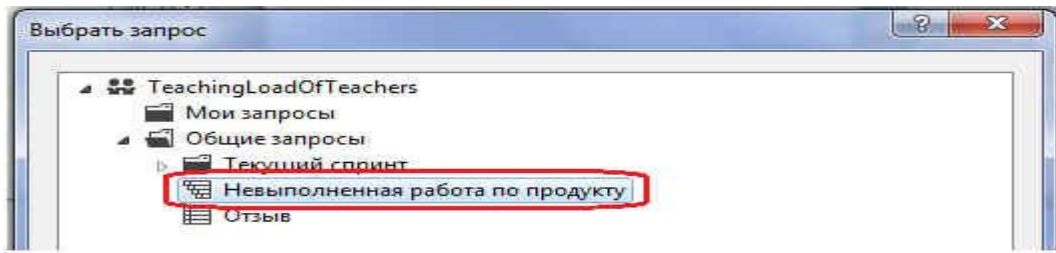


Рис. 8.5. Діалогове вікно Вибрати запит У книзі Excel представлена інформація про Елементах заділ робіт продукту у вигляді списку невиконаних робіт по продукту (рис. 8.6).

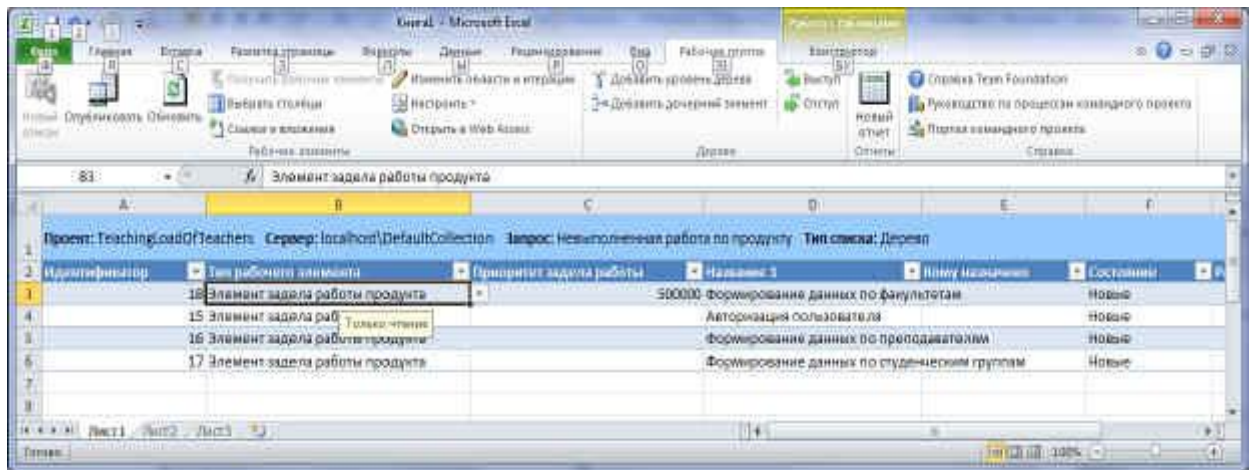


Рис. 8.6. Книга Excel до вимог користувачів

Введіть новий робочий елемент, тип якого задайте - Елемент заділ робіт продукту (рис. 8.7), а назва - Формування навчальних робочих планів (1) і для синхронізації з базою даних TFS натисніть кнопку Опублікувати (2).

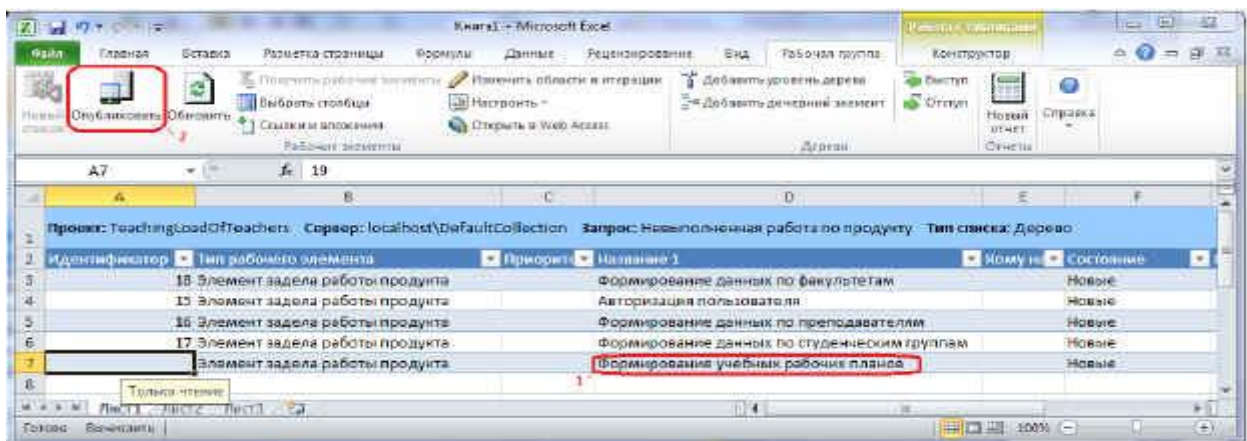


Рис. 8.7. Введення нового Елемента заділ робіт продукту

У Microsoft Excel при роботі з робочими елементами можна додати посилання і вкладення, відкрити проект в Web Access, відкрити портал проекту. Microsoft Excel дозволяє редагувати існуючі у проекті робочі елементи.

Створення користувальницьких вимог за допомогою Microsoft Project Для використання Microsoft Project запустіть цей додаток. Відкрийте вкладку Робоча група і натисніть кнопку Вибір командного проекту (мітка 1 на рис. 8.8).

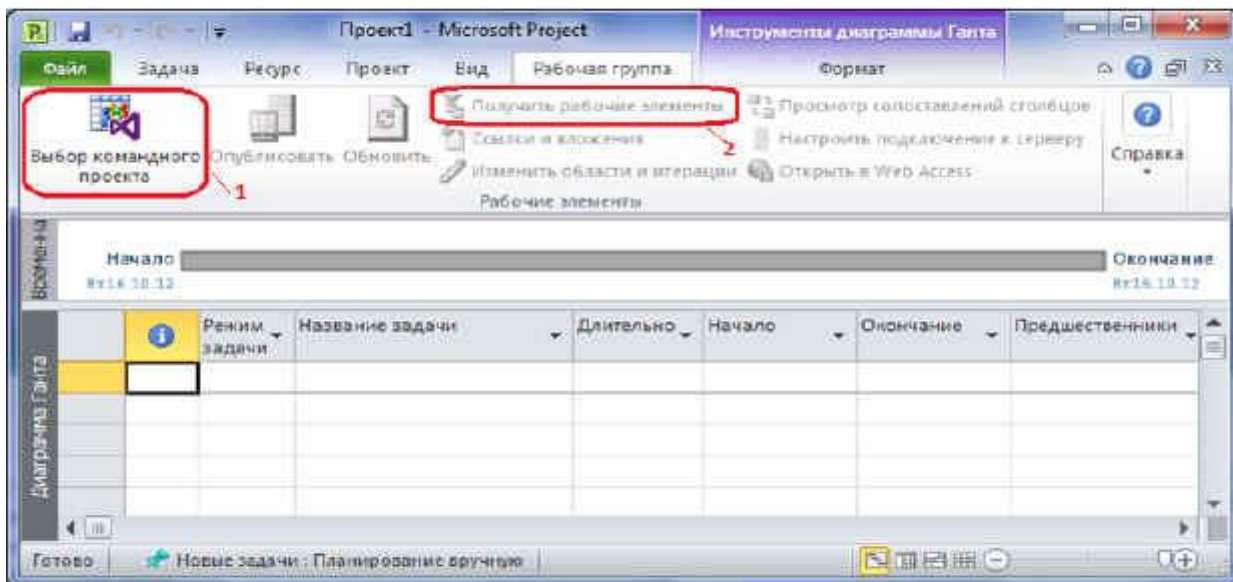


Рис. 8.8. Додаток Microsoft Project

Підключення до командного проекту TFS здійснюється аналогічно підключенню для Microsoft Excel. Після підключення до командного проекту активізується кнопка Отримати робочі елементи. Для імпорту робочих елементів і бази даних TFS в Microsoft Project натисніть кнопку Отримати робочі елементи (мітка 2 на рис. 8.8).

У діалоговому вікні Отримати робочі елементи включите перемикач перед пунктом Збережений запит і зі списку виберіть пункт невиконані роботи по продукту (рис. 8.9).

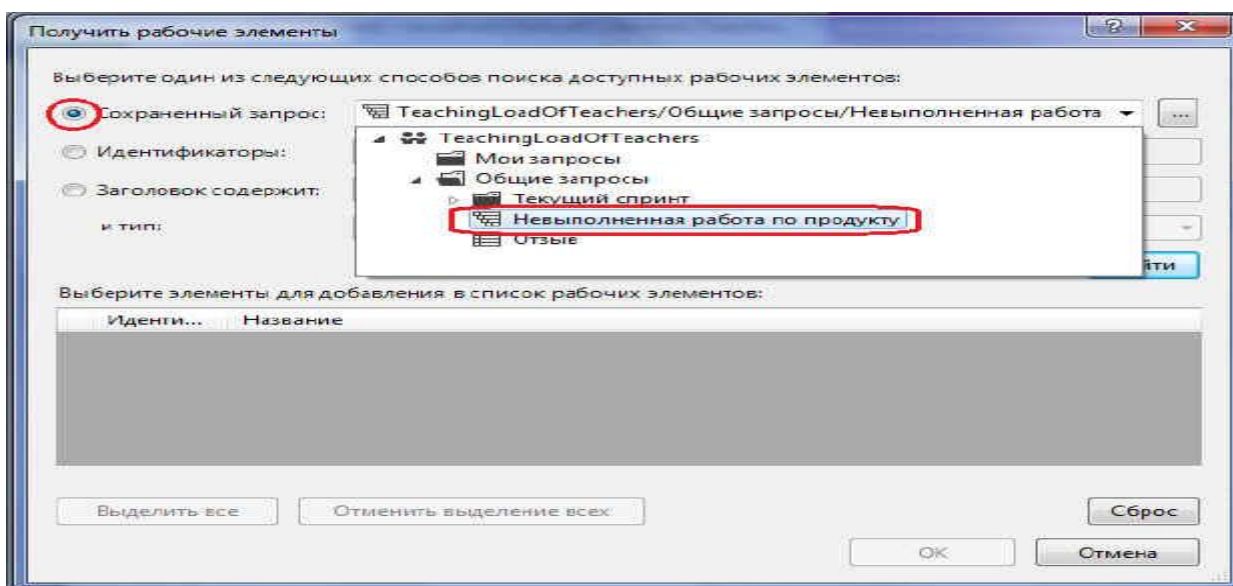


Рис. 8.9. Діалогове вікно Отримати робочі елементи Після формування запити натисніть кнопку Знайти (рис. 8.10).

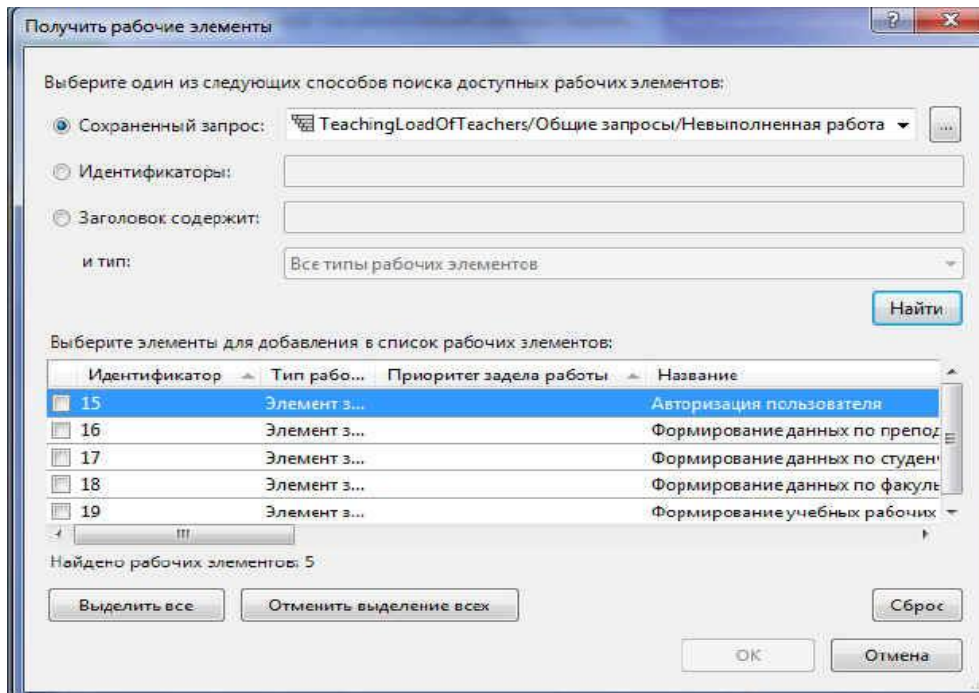


Рис. 8.10. Диалоговое окно Отримати робочі елементи

Для виділення всіх елементів заділ роботи проекту натисніть кнопку Виділити все. Після імпорту робочих елементів в Microsoft Project в діаграмі Ганта будуть представлені назви всіх вимог користувача (рис. 8.11).

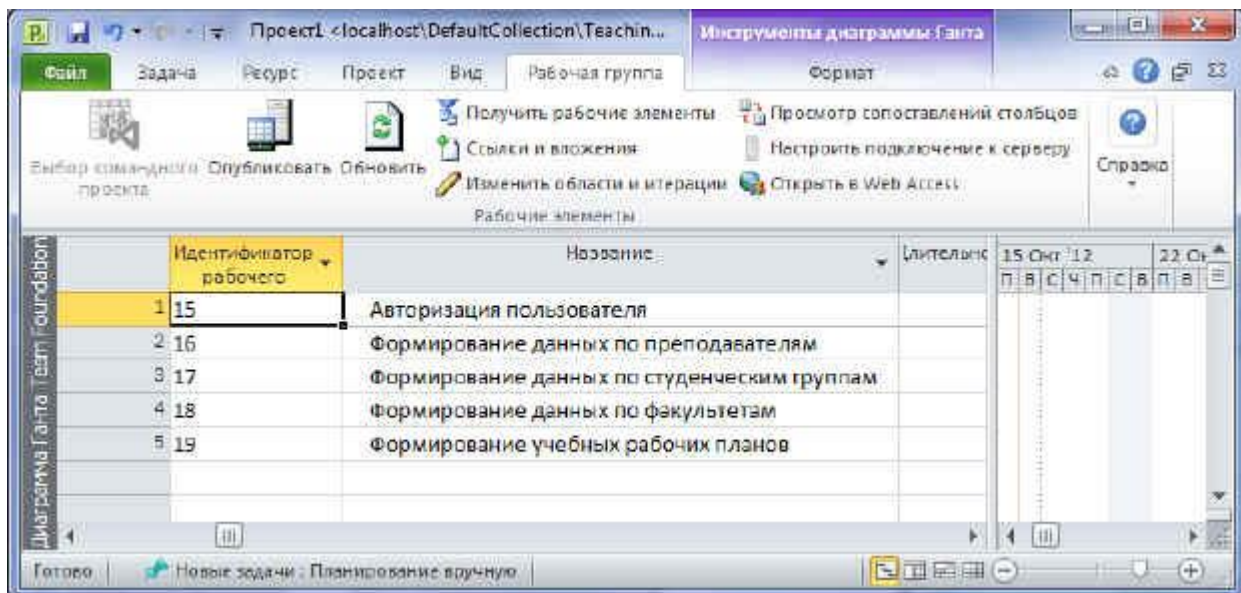


Рис. 8.11. Робочі елементи проекту в Microsoft Project

Введіть новий робочий елемент - Елемент заділ робіт продукту (рис. 8.12). У графу Найменування (1) введіть Формування навчального навантаження кафедри, зі списку Тип робочого елементу виберіть Елемент заділ роботи продукту і для синхронізації з базою даних TFS натисніть кнопку Опублікувати (3).



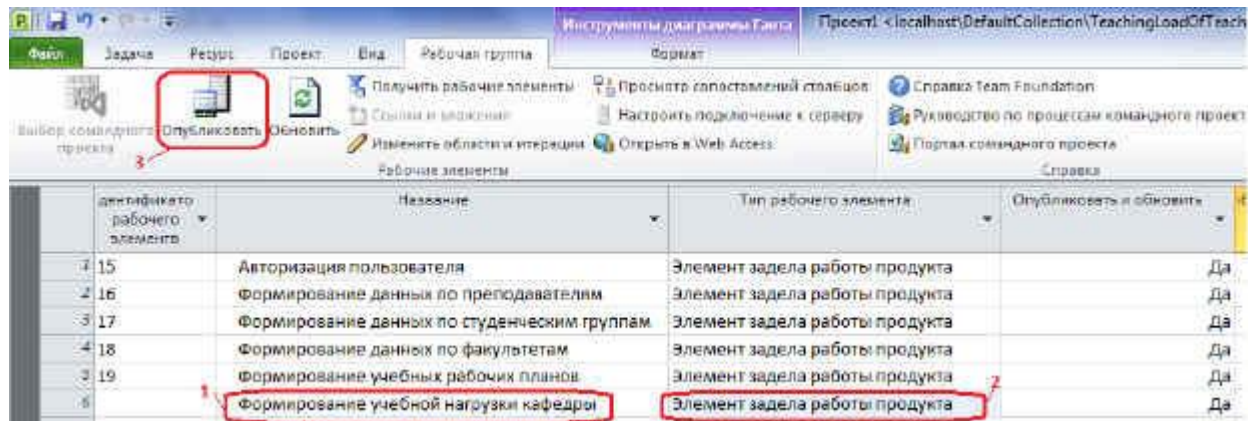


Рис. 8.12. Додавання робочого елементу проекту в Microsoft Project

При роботі з робочими елементами Microsoft Project, також як і Microsoft Excel дозволяє додавати посилання і вкладення, відкривати проект в Web Access, відкривати портал проекту, а також редагувати існуючі у проекті робочі елементи.

## Завдання

1. Ознайомитися з теоретичним матеріалом. Для розроблюваного проекту створити користувальницькі вимоги, використовуючи Microsoft Excel і Microsoft Project. По завданню викладача створити командний проект. Налаштувати параметри створеного командного проекту. параметри командного проекту.

Звіт з лабораторної роботи повинен складатися з:

1. Постановки завдання.
2. Документу «Протокол лабораторної роботи», що містить:
  - Назву роботи. Мету;
  - Хід виконання роботи (графічні результати виконання роботи);
  - Відповіді на контрольні питання;
  - Висновки.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.

## Контрольні питання

1. Назвіть основні принципи MSF.
2. Чим визначається універсальність моделі MSF?
3. Яка модель циклу програмної системи використовується в MSF?
4. У чому полягає Ітеративність методології MSF?
5. Поясніть призначення інтеграції в методології MSF?
6. Як можна масштабувати команду, що використовує методологію MSF?

## Лабораторна робота №7

*Аналіз вимог на розробку, проектування архітектури ПЗ. Вибір архітектурних шаблонів. Формування команди та розподіл ролей. Проектування схеми БД. Розробка діаграм прецедентів, ієрархії класів, послідовності, розгортання.*

**Мета:** Отримати практичні навички архітектурного проектування програмних додатків при розробці схем варіантів використання UML і схем класів UML.

### Теоретичні відомості

При командній розробці програмного забезпечення важливими питаннями є планування робіт, складання розкладу, управління областю проекту, комунікації, складання звітів, аналіз і постійне вдосконалення процесу. Для вирішення цих питань TeamFoundationServer пропонує наступні інструменти:

- шаблон процесу, який визначає процес, який використовується командним проектом;
- керівництво по процесу, яке містить опис шаблонів процесу;
- колекція командних проектів, яка представляє собою контейнер для декількох командних проектів;
- командний проект, який зберігає і організує дані про всьому життєвому циклі розробки програмного забезпечення;
- відстеження робочих елементів, яке дозволяє відстежувати стан робочих елементів та іншу інформацію, пов'язану з ними;
- портал проекту / панелі моніторингу, на яких надається інформація по проекту для всіх членів команди;
- елементи планування для управління списками вимог як на рівні проекту, так і на рівні ітерації;
- звітність, яка дозволяє формувати звіти в ході життєвого циклу проекту;
- інтеграція з MicrosoftProject і MicrosoftExcel для управління проектами з середовища MicrosoftOffice.

Розробка програмного забезпечення починається зі створення командного проекту. Командний проект містить інформацію про кожен крок життєвого циклу розробки програмного забезпечення, включаючи вимоги користувачів, завдання, тестові випадки, помилки, перешкоди, побудови.

При створенні командного проекту необхідно задати його ім'я, опис, визначитися з шаблоном процесу, вимогами до системи контролю версій і необхідністю створення порталу для проекту.

Після створення командного проекту керівник формує команду. Для кожного члена команди керівник проекту визначає доступ до проекту в цілому і окремим артефактів, які важливі для роботи кожного члена команди. Для структурування проекту використовуються області та ітерації. Області можуть визначатися виходячи з певного

функціонального призначення етапу робіт, а ітерації - у вигляді набору робіт на заданому часовому інтервалі. робочі елементи

При плануванні командного проекту ключовими сутностями є робочі елементи. Залежно від шаблону командного проекту набір і найменування робочих елементів дещо відрізняються. Так для гнучкої методології Agile робочими елементами є :

- Користувача опис функціональності (UserStory);
- Завдання (Task);
- Помилка (Bug);
- Перешкода (Issue);
- Тестовий випадок.

Робочий елемент користувача "опис функціональності" представляє собою користувальницьке вимога, яку необхідно виконати при реалізації проекту.

Робочий елемент "Завдання" створюється в проекті для призначення та виконання роботи. Завдання надають деталі реалізації для користувача вимог.

Робочий елемент "Помилка" використовується для відстеження та моніторингу проблем в програмному продукті.

Робочий елемент "Перешкода" використовується для фіксації в проекті подій або об'єктів, які створюють проблеми у виконанні проекту і повинні бути усунені в ході поточної або майбутньої ітерації.

Робочий елемент "Тестовий випадок" описує умови перевірки правильності виконання програмним продуктом вимог користувача.

Робочі елементи можуть включати найменування, опис призначення, стан, кому призначено, цінність для бізнесу, приналежність до робочої області.

При плануванні командного проекту користувальницькі вимоги записуються в Елемент невиконана робота по продукту. Дану роботу можна проводити з використанням:

- Командного оглядача в VisualStudio;
- Microsoft Project;
- Microsoft Excel.

Користувальницькі опису функціональності можуть бути пов'язані між собою, а також з дочірніми або батьківськими елементами. В якості дочірніх елементів можуть бути, наприклад, завдання або помилки.

Створення проекту моделювання програмного додатка. Після розробки початкового списку користувача вимог, які є змістом робочого елементу невиконані роботи по продукту, доцільно обговорити зафіксовані в проекті вимоги із зацікавленими особами. При проведенні обговорення користувача вимог до програмного продукту доцільно представити зацікавленим особам не тільки текстову документацію, але графічний матеріал, який більш наочно відображає користувальницькі вимоги. Для цього використовуються UML-діаграми.

Власник продукту, виконуючи роль архітектора, для підготовки графічних діаграм може використовувати можливості Visual Studio з архітектурного моделювання.

## Виконання роботи

Спочатку створимо пусте Рішення Visual Studio (1), в який будемо додавати проекти, пов'язані з програмним продуктом - ProjectTeachingLoadOfTeachers (2), зазначивши ознака Додати в систему управління версіями (3). На рис. 12.1 наведено вікно створення проекту.

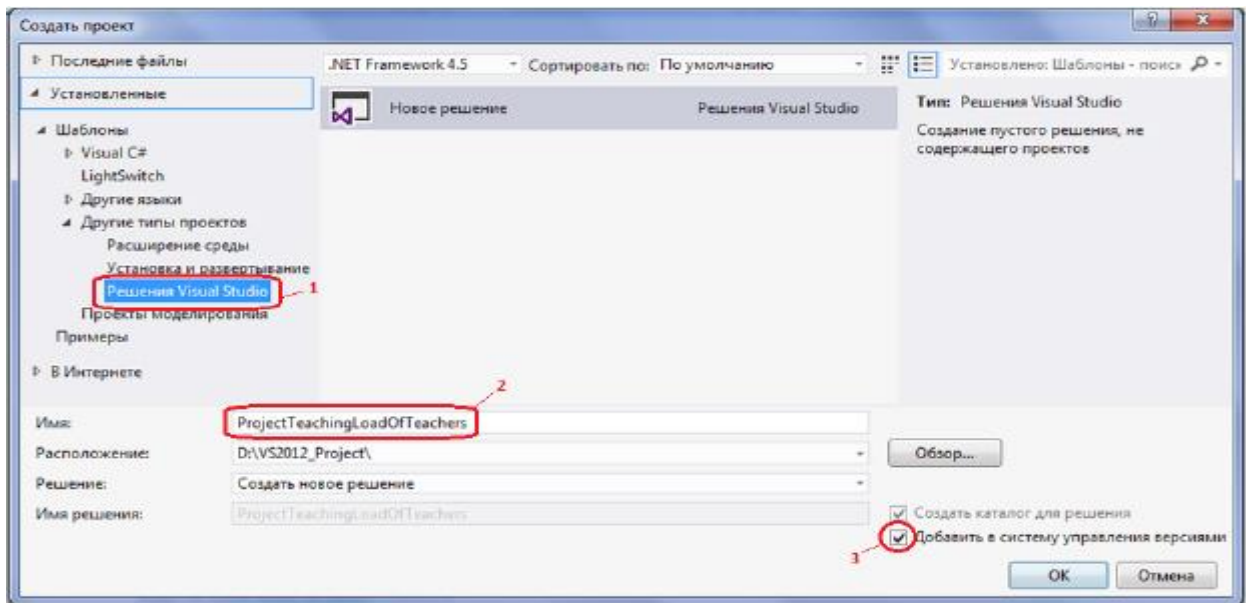


Рис. 12.1. Створення порожнього рішення Visual Studio

При додаванні рішення в систему управління версіями необхідно вказати розташування командного проекту (рис. 12.2).

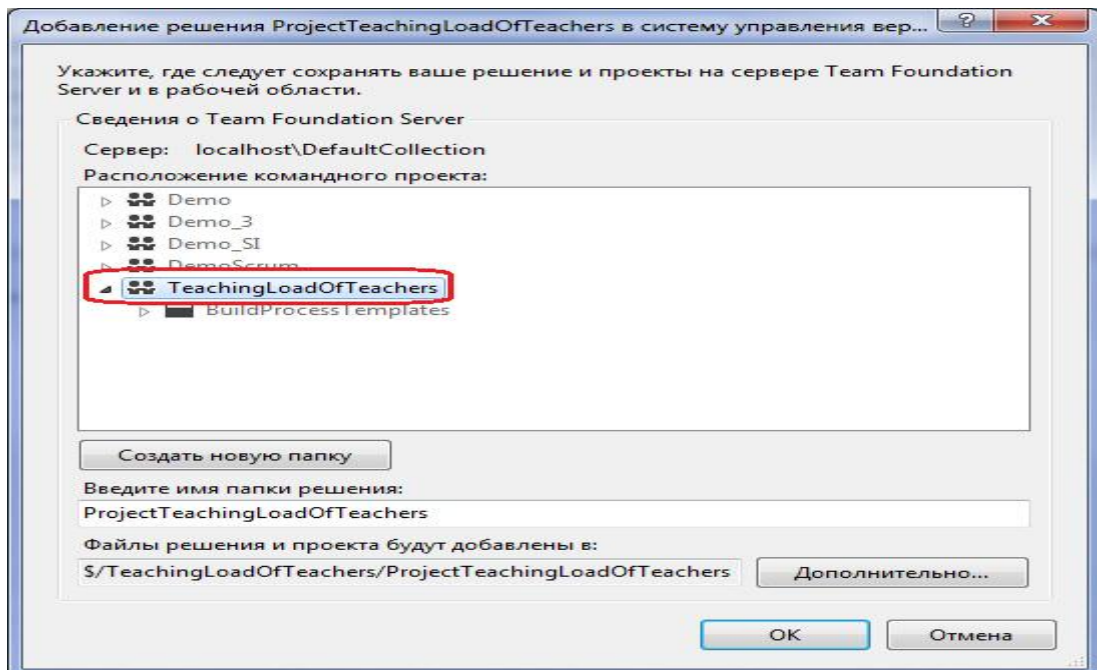


Рис. 12.2. Діалогове вікно Додавання рішення в систему управління версіями

У створенні рішення ProjectTeachingLoadOfTeachers додамо проект моделювання ModelingProjectTeachingLoadOfTeachers на основі шаблону Проект моделювання (рис. 12.3).

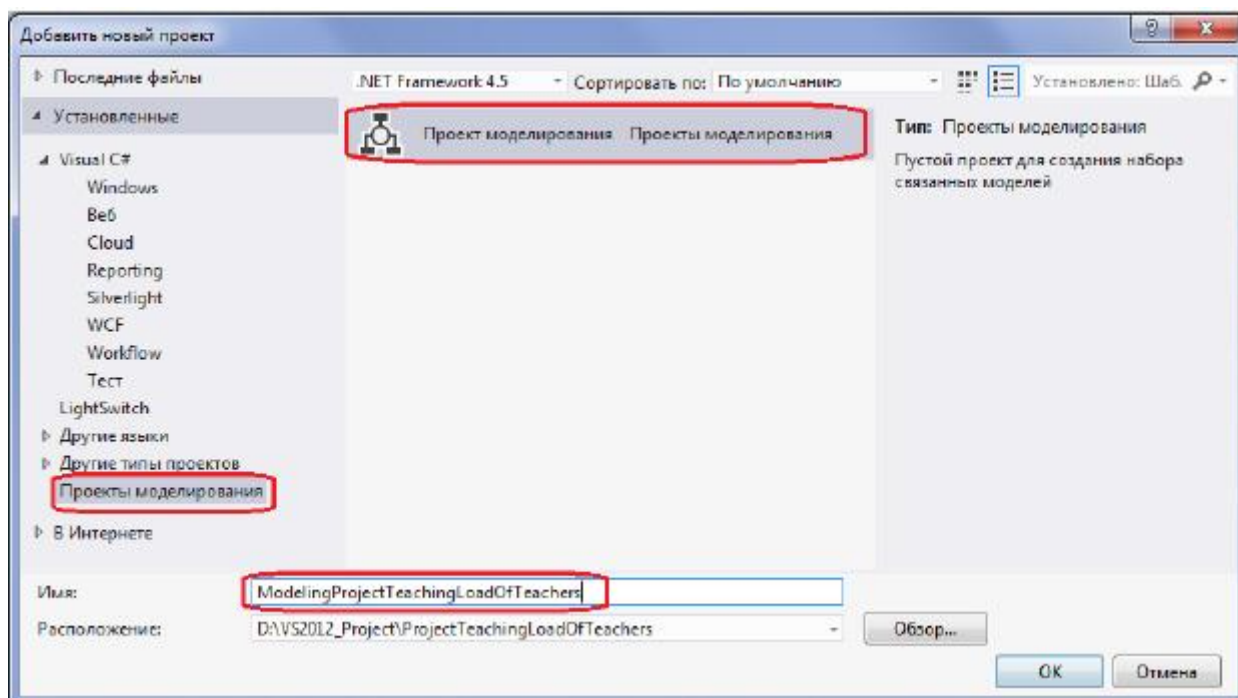


Рис. 12.3. Додавання проекту моделювання

У браузері рішень буде додано проект моделювання ModelingProjectTeachingLoadOfTeachers.modelproj (рис. 12.4).

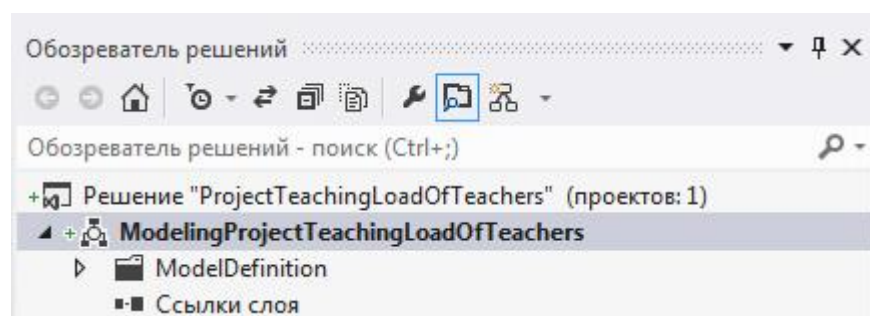


Рис. 12.4. Представлення проекту моделювання в оглядачі рішень

Інструментальні засоби Visual Studio включають шаблони для створення наступних UML-діаграм:

- Схема класів UML;
- Схема послідовностей UML;
- Схема варіантів використання UML;
- Схема активності UML;
- Схема компонентів UML;
- Схема шарів.



### Розробка схеми варіантів використання

Для моделювання функціональних вимог до програмного продукту використовують схему варіантів використання, яка представляє потреби користувача до системи. Застосуємо шаблон Схема варіантів використання UML для моделювання користувача вимог (рис. 12.5).

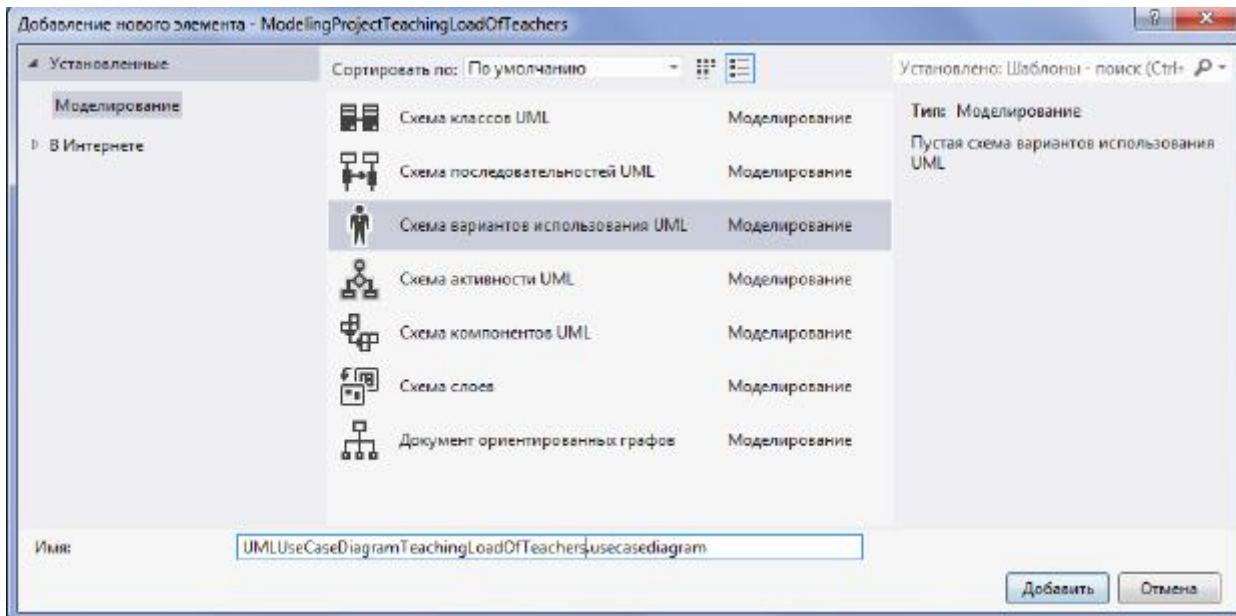


Рис. 12.5. Додавання Схеми варіантів використання UML

При додаванні в проект моделювання схеми варіантів використання буде відображений дизайнер схем (рис. 12.6).

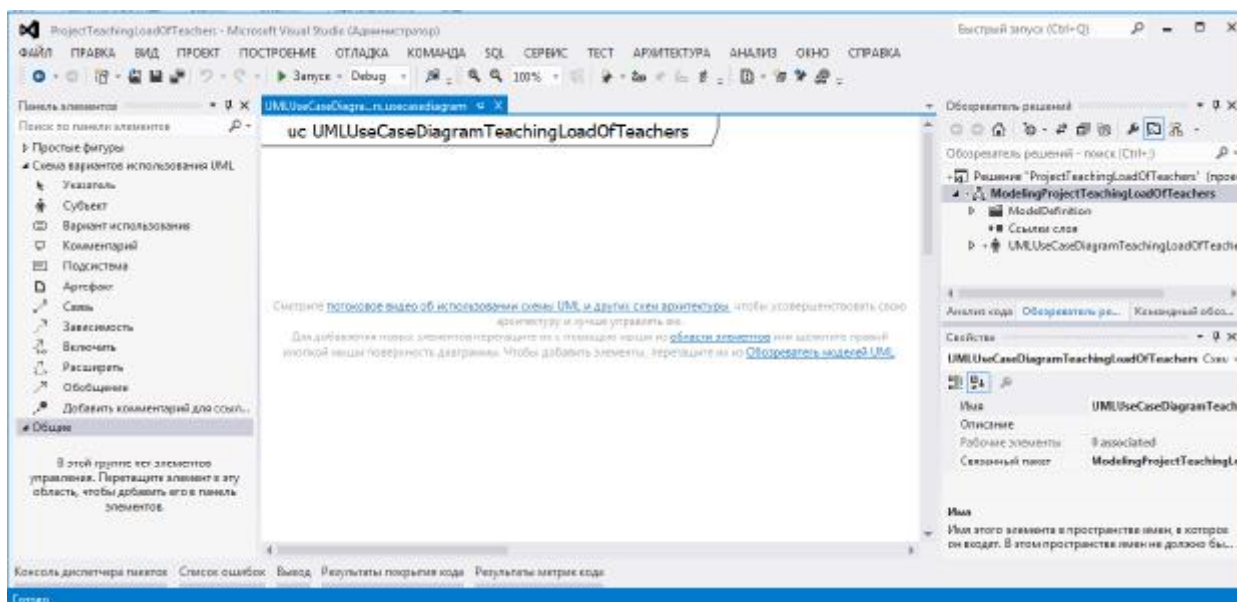


Рис. 12.6. Дизайнер побудови схеми варіантів використання

Дизайнер побудови схеми варіантів використання має набір елементів для створення схеми:

- Показчик, для виділення елементів схеми;

- Суб'єкт, для додавання користувачів або зовнішньої системи, яка взаємодіє з розробляється системою;
- Варіант використання, для додавання специфікації дій, які виконує користувач системи;
  - Коментар, для введення пояснень у схему;
  - Підсистема, для об'єднання варіантів використання в підсистемі;
  - Артефакт, для створення посилання на іншу схему або документ;
  - Зв'язок, для з'єднання суб'єкта з варіантом використання;
  - Залежність, для визначення залежності варіантів використання;
  - Включити, для визначення того, що один варіант використання викликає інший;
  - Розширити, для визначення того, що один варіант використання розширює визначення іншого варіанту використання за певних умов;
  - Узагальнення, для визначення того, що один варіант використання є уточненням іншого і успадковує його функції і обмеження;
  - Додати коментар для посилання - для з'єднання коментаря з елементом схеми.

*Для розроблюваного програмного продукту в навчальному проекті визначені три суб'єкти:*

- методист навчального відділу;
- методист факультету;
- методист кафедри.

*Функціональність проектованої системи задана наступними призначеними для користувача вимог, визначених у робочому елементі невиконані роботи по продукту:*

- авторизація користувача;
- формування даних по викладачам;
- формування даних по факультетах;
- формування даних по студентським групам;
- формування навчальних планів;
- формування навчального навантаження кафедри;
- розподіл навчального навантаження кафедри.

На основі призначених для користувача вимог, які описані в робочих елементах проекту Елементи заділ робіт продукту, розробляється схема варіантів використання проектованої програмної системи (рис. 12.7). Схема створюється шляхом перетягування елементів Панелі інструментів у поле дизайнера і з'єднання вибраних елементів. Для елементів схеми задаються властивості, наприклад ім'я, опис, зв'язку з робочими елементами.

Варіанти використання доцільно пов'язати з Елементами заділ робіт продукту для відстеження взаємозв'язку робочих елементів проекту і схем архітектурного моделювання. Для створення зв'язку варіанту використання з робочим елементом необхідно виділити варіант використання в дизайнера і або натиснути кнопку в панелі інструментів, або викликати контекстне меню, клацнувши правою кнопкою миші на зображенні варіанту використання, і вибрати пункт меню Зв'язати із робочим елементом.

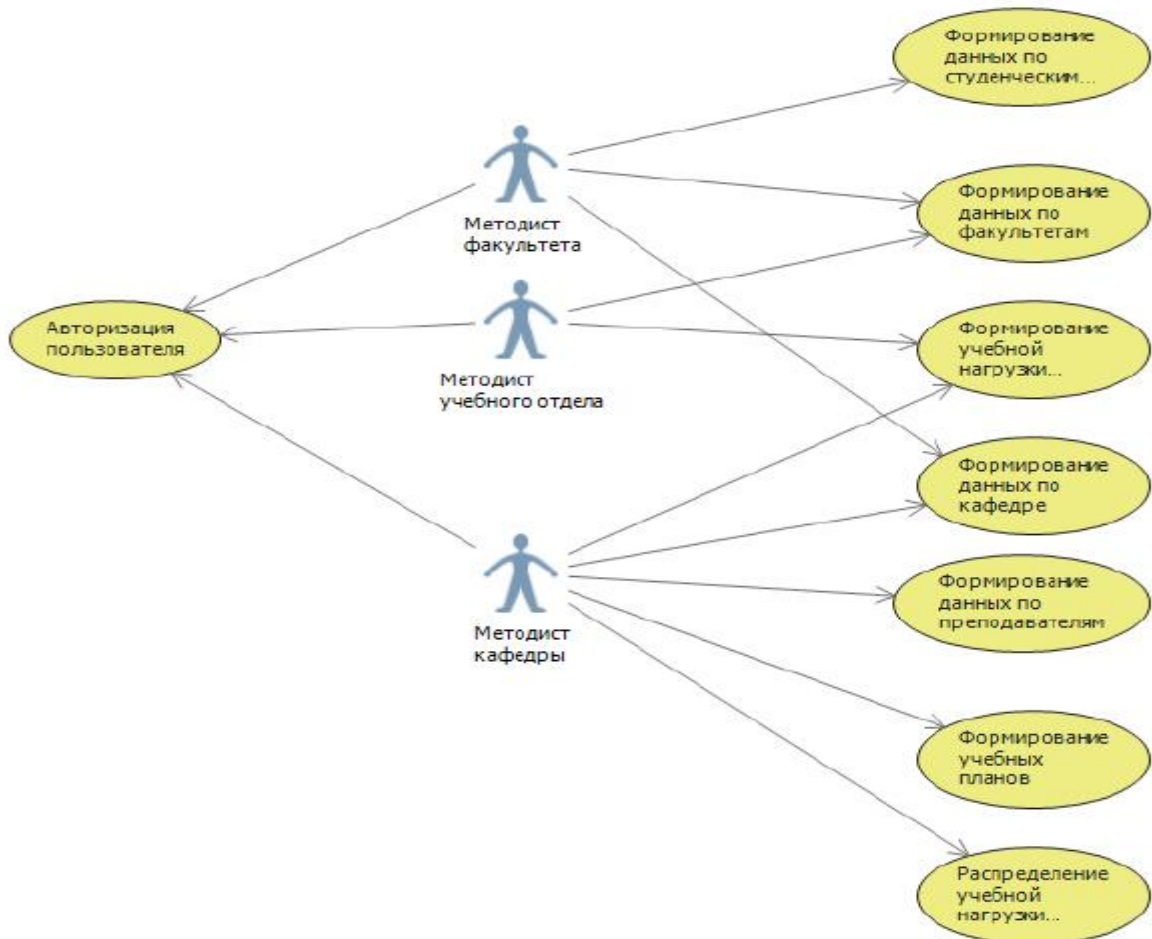


Рис. 12.7. Схема вариантов использования проектной системы

У диалоговому вікні Зв'язок з робочими елементами необхідно сформувати запит на вибірку робочих елементів, відзначити необхідний робочий елемент і натиснути кнопку ОК (рис. 12.8).

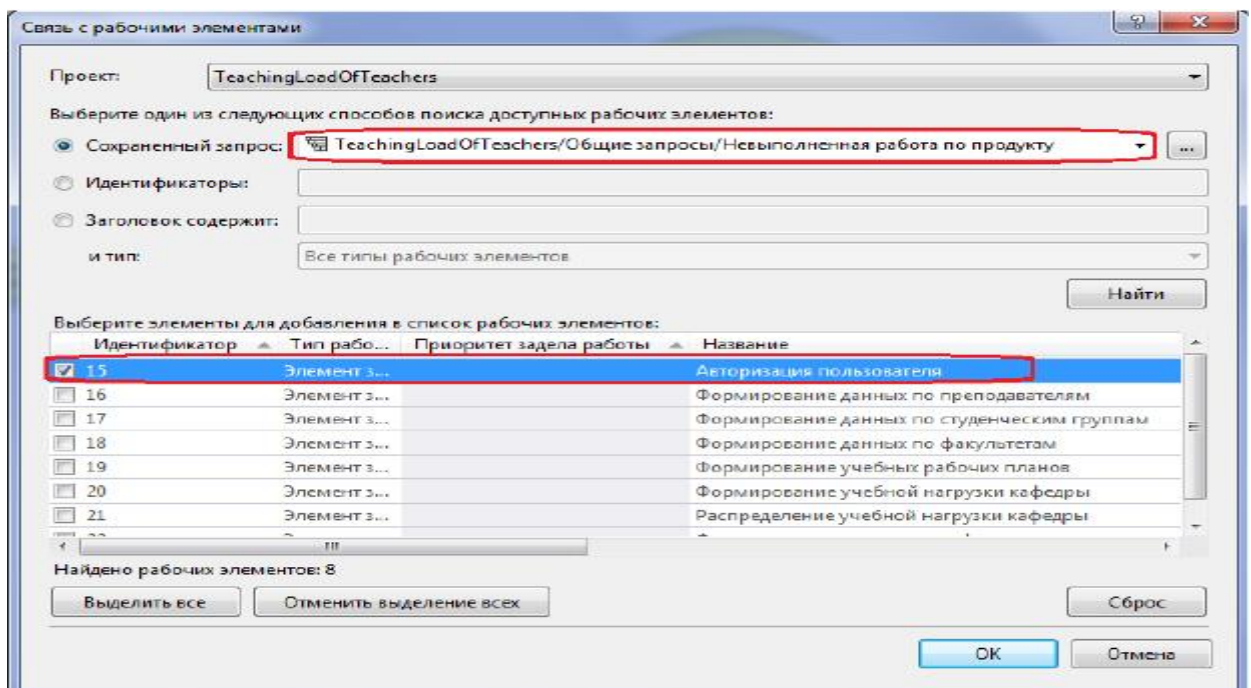


Рис. 12.8. Диалоговое окно Зв'язок з робочими елементами

При зв'язуванні з робочими елементами на схемі поряд з варіантами використання з'являються значки (рис. 12.9).

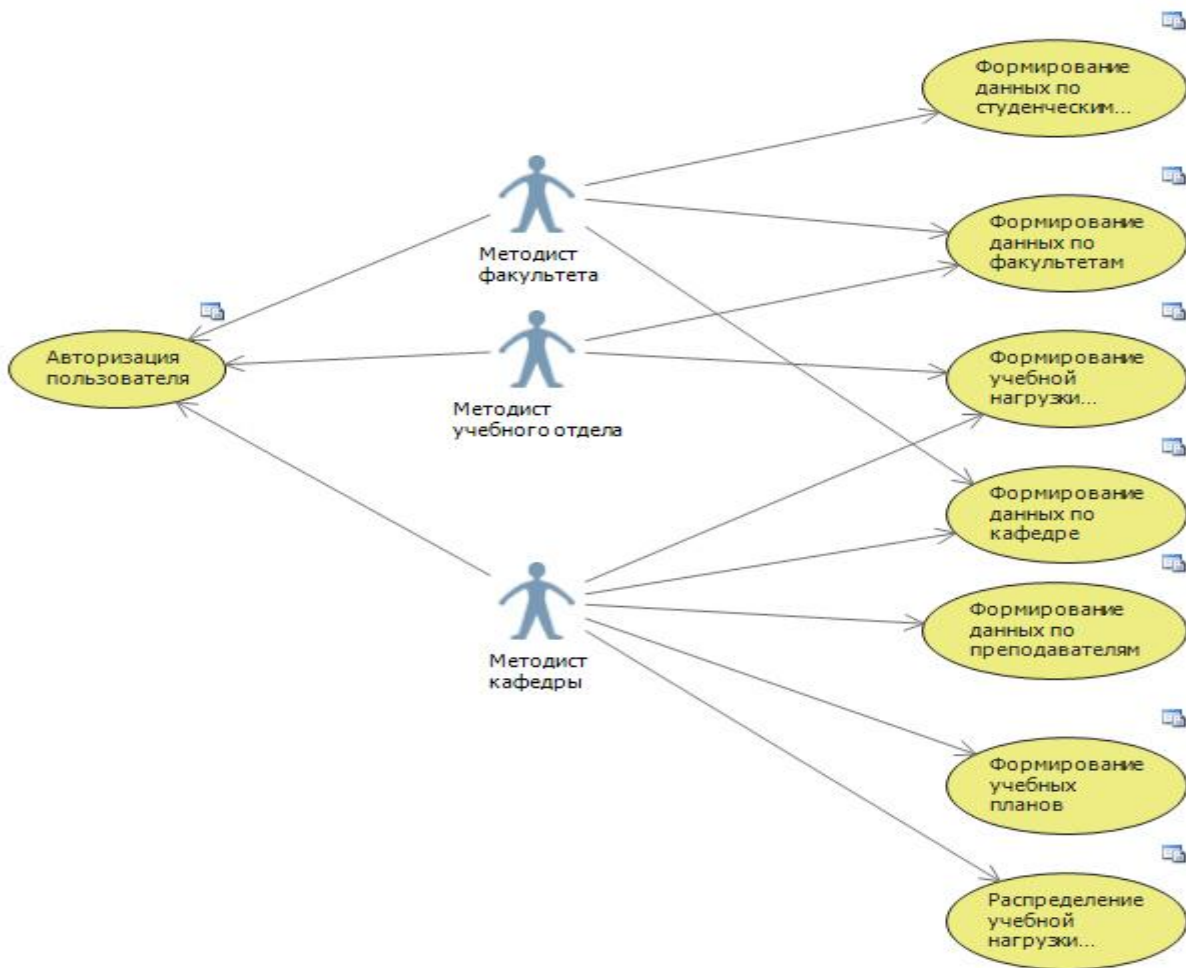


Рис. 12.9. Схема варіантів використання зі зв'язками з робочими елементами

Якщо клацнути на значок поруч з варіантом використання, то відкривається пов'язаний з ним робочий елемент, наприклад Формування даних по факультету (рис. 12.10). При наявності посилання на модель з опису робочого елемента також можна перейти до відповідної моделі, клацнувши на посиланні Зв'язок моделі.

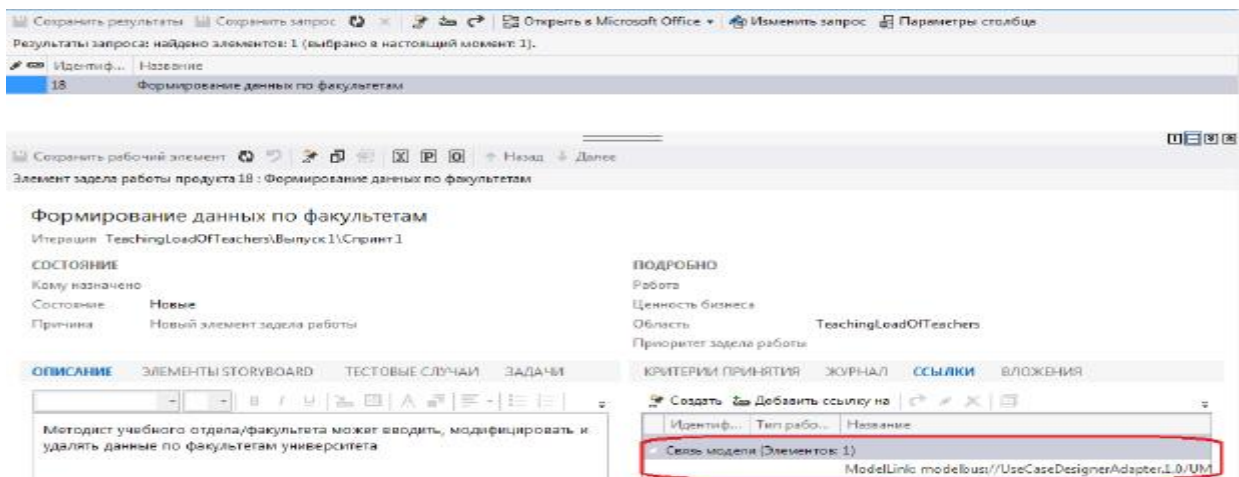


Рис. 12.10. Робочий елемент з посиланням на модель

### Розробка схеми класів

Моделювання сутностей програмної системи проводять за допомогою схеми класів UML. Схема класів, створювана на початкових етапах проектування програмного продукту, дозволяє попередньо проаналізувати сутності програмної системи та їх зв'язку на досить високому рівні абстракції. Створимо схему класів для проекрованої програмної системи. Додамо в проект моделювання схему класів UML (рис. 12.11).

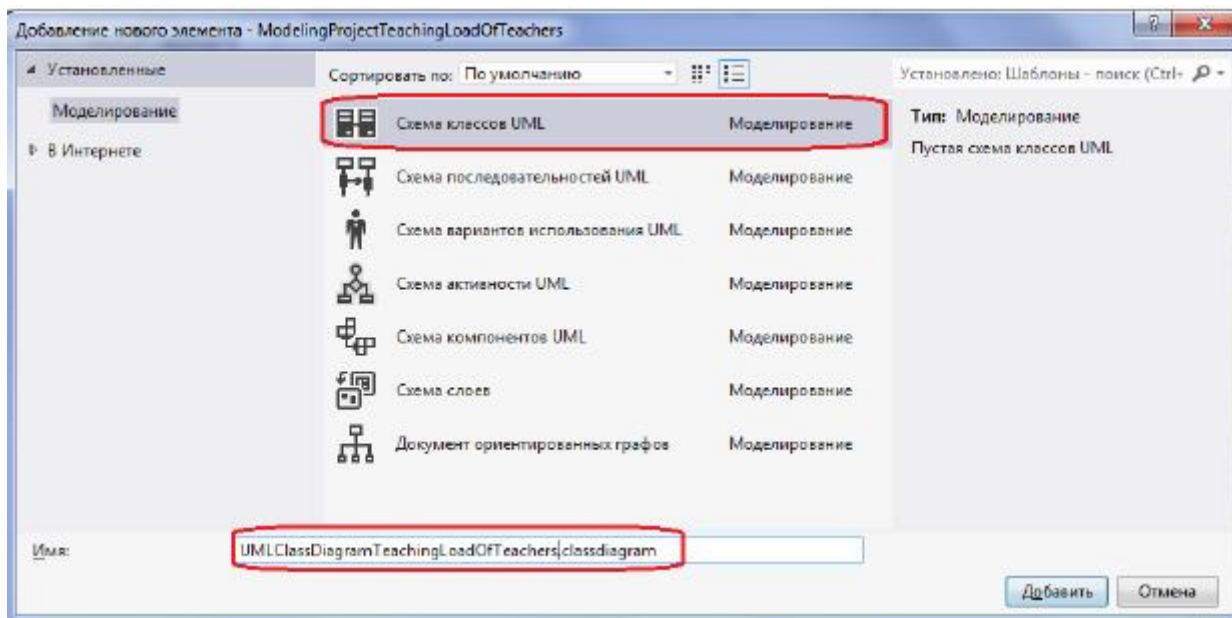


Рис. 12.11. Додавання схеми класів UML

В панелі інструментів для побудови схеми класів (рис. 12.12) є такі елементи як Клас, Інтерфейс, Перерахування, Пакет і Коментар, які можна додавати на схему, а також різні типи зв'язків класів: Асоціація, Агрегат (Агрегація), Композиція, Залежність і успадкування.

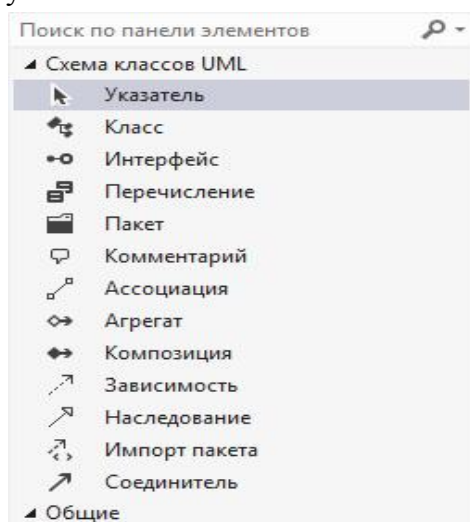


Рис. 12.12. Панель інструментів для побудови схеми класів

При обговоренні з зацікавленими особами власник продукту з'ясував, що для проекрованої програмної системи важливими є такі класи:

- Університет;



- Факультет;
- Кафедра;
- Викладач;
- Навчальне навантаження кафедри;
- Навчальне навантаження викладача;
- Навчальний робочий план напряму підготовки;
- Студентська група;
- Норми часу для розрахунку обсягу навчальної роботи.

Власник продукту, виконуючи роль архітектора, розробляє попередню схему класів програмної системи (рис. 12.13).

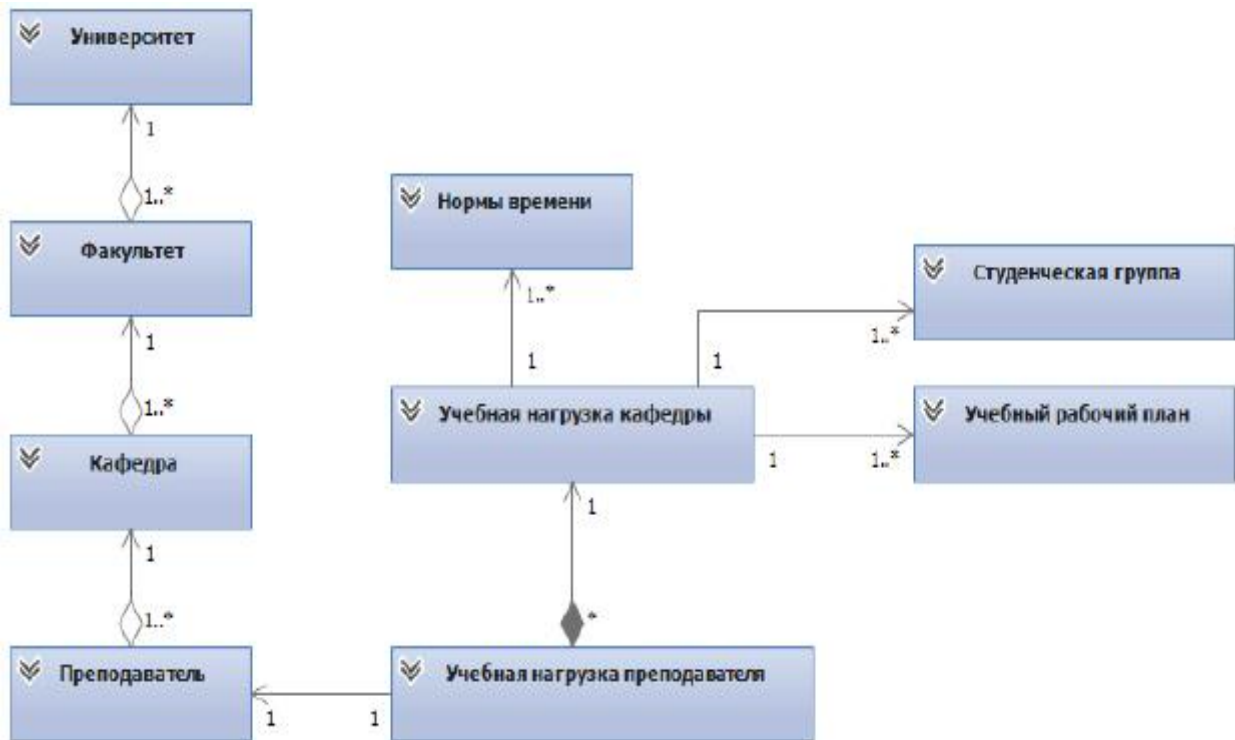


Рис. 12.13. Схема класів проектової системи

Проект моделювання створювався в локальній папці. У браузері рішень проект та створені схеми позначені знаком, що відзначає не збережені в базі даних TFS елементи рішень (рис. 12.14).

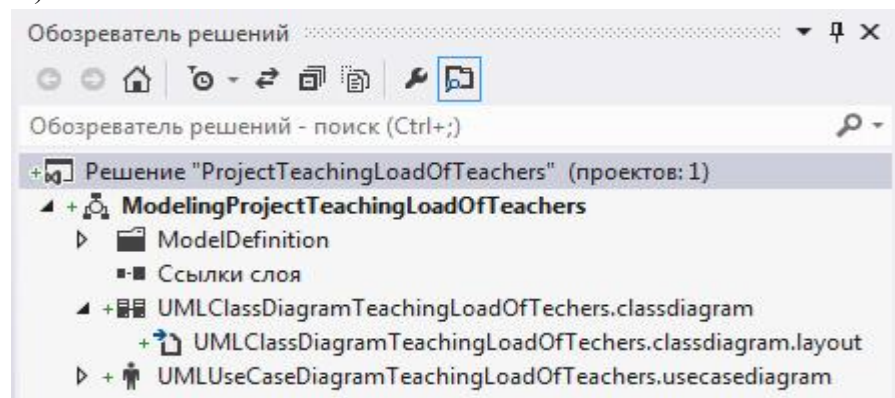


Рис. 12.14. Схема класів проектової системи

Для збереження в базі даних TFS елементів рішень необхідно натиснути правою кнопкою миші на рядку рішення і з випадаючого меню вибрати пункт Повернути. У діалоговому вікні необхідно підтвердити збереження елементів рішення і натиснути кнопку Так (рис. 12.15).

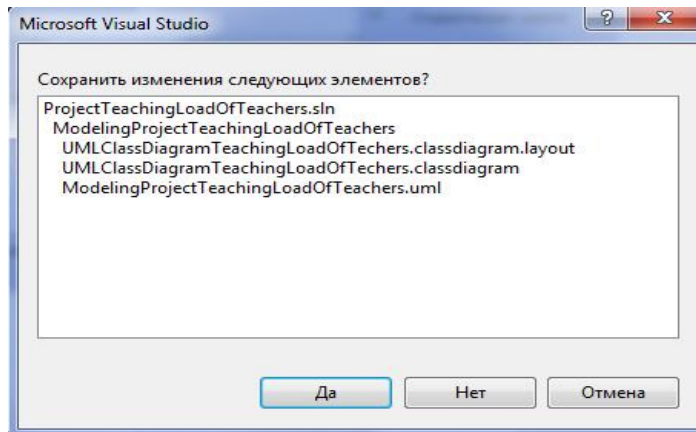


Рис. 12.15. Діалогове вікно підтвердження збереження елементів рішення

Після підготовки повернення змін до бази даних сервера TFS в командному браузері на вкладці Очікують зміни потрібно натиснути кнопку Повернення і при необхідності додати коментарі (рис. 12.16).

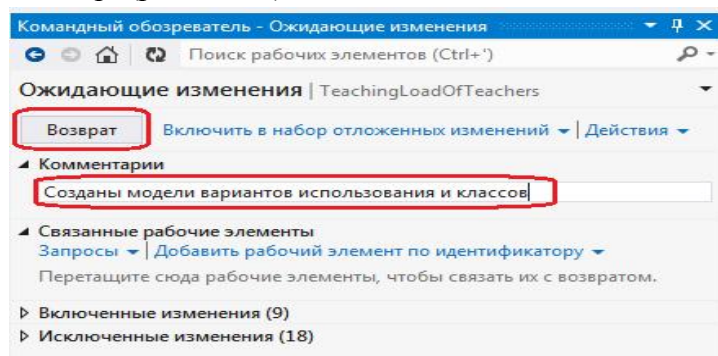


Рис. 12.16. Вкладка Очікують зміни командного оглядача

При виконанні повернення змін виводиться повідомлення про успішно виконану операцію (рис. 12.17).

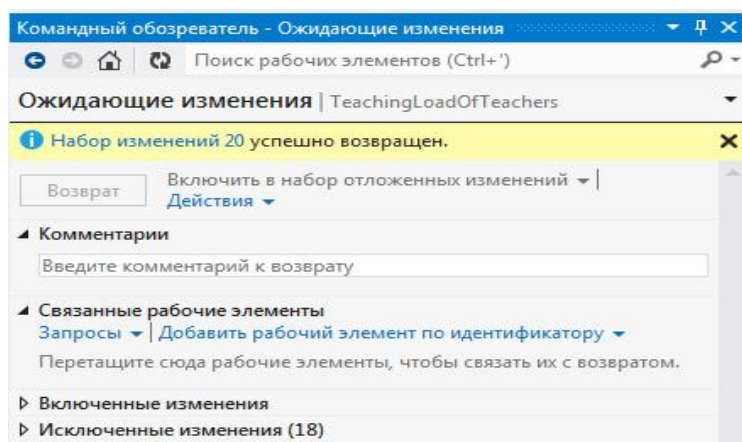


Рис. 12.17. Повідомлення про успішне повернення змін

Слід мати на увазі, що після обговорення із зацікавленими особами користувача вимог до програмної системи та основних класів можуть бути внесені зміни до схеми

варіантів використання, схеми класів, робочі елементи невиконані роботи по продукту та Елементи заділ робіт продукту.

### **Контрольні питання**

1. Які інструменти маються на TeamFoundationServer для управління командними проектами?
2. Які папки генеруються при створенні командного проекту?
3. Для чого використовуються у проекті області та ітерації?
4. Які робочі елементи наявні в шаблоні Agile?
5. За допомогою яких клієнтських інструментів можна проводити планування командного проекту?

Поясніть процес створення коду розробником командного проекту в середовищі VisualStudio.

### **Завдання**

Ознайомитися з теоретичним матеріалом. Для розроблюваного проекту, згідно з варіантом завдання здійснити:

- моделювання функціональності і класів
- схему варіантів використання
- схему класів
- аналіз вимог на розробку,
- проектування архітектури ПЗ,
- проектування схеми БД.
- розробку діаграм прецедентів,
- ієрархію класів, послідовності, розгортання.

Звіт з лабораторної роботи повинен складатися з:

1. Постановки завдання.
2. Документу «Протокол лабораторної роботи», що містить:
  - Назву роботи. Мету;
  - Хід виконання роботи (графічні результати виконання роботи);
  - Відповіді на контрольні питання;
  - Висновки.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.



## Варіанти завдань

### Варіант 1. Передплата на періодичні видання

Видання	Видавництва	Передплатник	Передплата
Індекс	код видавництва	код передплатника	код квитанції
Назва	Назва	назва	код передплатника
вартість номеру	Адреса	адреса	Індекс видання
періодичність	e-mail		термін підписки
Код видавництва	ПІБ головного редактора		дата підписки

### Варіант 2. Футбольна ліга

Гравці	Команди	Персонал	Посади
Код гравця	Код команди	код	код посади
Прізвище	Назва	прізвище	назва
Ім'я	Стадіон	Ім'я	
Дата народження	Місто	По батькові	
Зріст		дата народження	
Амплуа		код команди	
код команди		код посади	
Номер			

### Варіант 3. Вироби народної творчості

Вироби	Кваліфікація	Майстри	Промисли
Код виробу	код кваліфікації	код майстра	код промислу
назва виробу	назва	прізвище	Назва промислу
код промислу		Ім'я	
Код майстра		По батькові	
		адреса	
		дата народження	
		код кваліфікації	

### Варіант 4. Розклад занять викладачів

Предмети	Заняття	Викладачі	Кафедри
Код предмету	Код	код викладача	код кафедри
			назва
Назва	код предмету	Прізвище	
Скорочена назва	код викладача	Ім'я	
	день тижня	По батькові	
	номер пари	посада	
	№ аудиторії	код кафедри	
	Група		

### Варіант 5. Вироби кондитерських фабрик

Вироби	Види виробів	Виробники	Форми власності
код виробу	код виду	код виробника	код форми власності
Назва	назва виду	Назва	Назва
код виду		Адреса	
одиниця вимірювання		код форми власності	
ціна за одиницю		відповідальна особа	
Код виробника		телефон	
		e-mail	

### Варіант 6. Склад

Споживач	Товари	Накладні	Поставка
код споживача	код товару	№ накладной	№ накладной
Назва	Назва	дата	код товару
Адреса	одиниця вимірювання	код споживача	Кількість
Телефон	ціна за одиницю		
Примітки	Виробник		

### Варіант 7. Будівельна організація

Об'єкти	Типи об'єктів	Матеріали	Витрата
код об'єкту	код типу об'єкту	код матеріалу	код об'єкту
Назва об'єкту	назва типу	назва матеріалу	код матеріалу
код типу об'єкту		Одиниця вимірювання	витрата матеріалу
Адреса		Ціна за одиницю	

### Варіант 8. Видавництво

Автори	Книги	Замовники	Замовлення
код автора	код книги	код замовника	код замовлення
Прізвище	Тип видання	Назва	код замовника
Ім'я	назва книги	Адреса	дата замовлення
По батькові	код автора	телефон	код книги
дата народження	вартість	примітки	Тираж
Країна			Сплачено

### Варіант 9. Автосалон

Фірми	Моделі	Автомобілі	Рахунок	Клієнти
код фірми	код моделі	код автомобіля	№ рахунку	код клієнта
Назва	Назва	код моделі	код клієнта	Назва
	код фірми	дата випуску	код автомобіля	Адреса
	макс. швидкість	колір	дата	Телефон
	кількість дверей	пробіг		Примітки
	кількість місць	ціна		

**Варіант 10. Відділ кадрів**

<b>Співробітники</b>	<b>Кваліфікація</b>	<b>Посади</b>	<b>Відділи</b>
код співробітника	код кваліфікації	код посади	№ відділу
Прізвище	назва кваліфікації	назва посади	назва відділу
Ім'я		мінімальний оклад	макс. число співробітників
По батькові		максимальний оклад	
Адреса			
код кваліфікації			
код посади			
дата прийому на роботу			
№ відділу			
Освіта			

**Варіант 11. Спортивне товариство**

<b>Спортсмени</b>	<b>Тренери</b>	<b>Види спорту</b>	<b>Тренування</b>
код спортсмена	код тренера	код виду спорту	код тренера
Прізвище	прізвище	назва	код спортсмена
Ім'я	Ім'я		
По батькові	По батькові		
дата народження	дата народження		
Звання	Звання		
	код виду спорту		

**Варіант 12. Ландшафтне озеленення**

<b>Замовники</b>	<b>Рослини</b>	<b>Співробітники</b>	<b>Договори</b>
код замовника	код рослини	Код співробітника	код договору
Назва	Назва	Прізвище	дата заключення
тип замовника	кімнатне	Ім'я	Термін виконання
Адреса	морозостійке	По батькові	код співробітника
Телефон	Ціна	Дата народження	код замовника
Примітки	щільність висадки	адреса	Площа
		дата прийому на роботу	код рослини

**Варіант 13. Замовлення на виклик лікаря**

<b>Лікарі</b>	<b>Пацієнти</b>	<b>Виклики</b>	<b>Посади</b>
№ ліцензії	№ карточки	код виклику	код посади
дата отримання ліцензії	Прізвище	№ карточки	Назва
Прізвище	Ім'я	№ ліцензії	
Ім'я	По батькові	дата/час виклику	
По батькові	Стать	відмітка про відвідування	
код посади	дата народження	діагноз	
Адреса	Адреса		
Телефон	Телефон		

## Лабораторна робота №8

### *Розробка прототипів компонентів. Реалізація базової компоненти. Модульне тестування. Тестування компонентів. Інтеграція компонентів в систему і інтеграційне тестування.*

**Мета:** Отримати практичні навички проведення оцінки складності елементів робіт та встановити пріоритети робочим елементам, включених в поточний спринт. Розподілити завдання спринту між членами команди. планування спринту

### **Теоретичні відомості**

Якість програмного продукту визначається за кількома критеріями. Якісний програмний продукт повинен відповідати функціональним і нефункціональним вимогам, відповідно до яких він створювався, мати цінність для бізнесу, відповідати очікуванням користувачів.

В життєвому циклі управління додатками якість повинна відслідковуватися на всіх етапах життєвого циклу ПЗ. Воно починає формуватися з визначення необхідних вимог. При завданні вимог необхідно вказувати бажану функціональність і способи перевірки її досягнення.

Якісний програмний продукт повинен володіти високим споживчим якістю, незалежно від галузі застосування: внутрішнє використання розробником, бізнес, наука та освіта, медицина, комерційні продажі, соціальна сфера, розваги, веб та ін Для користувача програмний продукт повинен задовольняти певному рівню його потреб.

Важливим аспектом створення якісного ПЗ є забезпечення функціональних вимог, таких як зручність в експлуатації, надійність, продуктивність, захищеність, зручність супроводу. Надійність ПО визначає здатність без збоїв виконувати задані функції в заданих умовах і протягом заданого відрізка часу. Продуктивність характеризується часом виконання заданих транзакцій або тривалих операцій. Захищеність визначає ступінь безпеки системи від пошкоджень, втрати, несанкціонованого доступу та злочинної діяльності. Зручність супроводу визначає легкість, з якою обслуговується продукт в плані простоти виправлення дефектів, внесення коректив для відповідності новим вимогам, управління зміненої середовищем.

Управління життєвим циклом програмного продукту допомагає розробникам цілеспрямовано домагатися створення якісного ПЗ, уникати втрат часу на переробку, повторне проектування та перепрограмування ПЗ.

### **Тестування програмного забезпечення**

Тестування програмного продукту дозволяє протягом усього життєвого циклу ПЗ гарантувати, що програмні проекти відповідають заданим параметрам якості. Головна мета тестування - визначити відхилення в реалізації функціональних вимог, виявити помилки у виконанні програм і виправити їх якомога раніше в процесі виконання проекту.

Протягом усього життєвого циклу розробки ПО застосовуються різні типи тестування для гарантії того, що проміжні версії відповідають заданим показникам якості. При цьому застосовуються автоматичні і ручні тести.

Для розробника програмного забезпечення в VisualStudio 2012 надана можливість створювати модульні і навантажувальні тести, а також тести для користувача інтерфейсу.

У VisualStudio 2012 є наступні шаблони тестових проектів:

- проект модульного тесту, який дозволяє створювати модульні тести в процесі розробки;
- проект з веб-тестами продуктивності і навантажувальними тестами;
- проект з закодованими тестами для користувача інтерфейсу.

Інструментарієм тестувальника в VisualStudio 2012 є MicrosoftTestManager (MTM). MTM призначений для управління життєвим циклом тестування програмного забезпечення, включаючи планування, тестування та моніторинг. MTM інтегрований з TeamFoundationServer. За допомогою Microsoft TestManager тестувальники готують плани тестування, управляють тестуванням. При створенні плану тестування в нього додаються набори тестів, тестові випадки і конфігурації, необхідні для тестування. Конфігурації використовуються для встановлення середовища, в якій будуть виконуватися набори тестів. Microsoft TestManager дозволяє виконувати ручні та автоматичні тести, а також дослідницькі тести. Результати тестування зберігаються в базі даних, що дозволяє готувати різні аналітичні звіти. Помилки, виявлені в процесі тестування, фіксуються, документуються і передаються розробникам для їх усунення. При внесенні змін в код програмної системи виникає необхідність у регресійному тестуванні, причому MTM автоматично формує план регресійного тестування, виявляючи які тести повинні бути повторно виконані.

Для тестувальників і розробників програмного забезпечення VisualStudio 2012 включає диспетчер віртуального середовища LabManagement. Інструментарій тестування LabManagement дозволяє створити інфраструктуру, яка максимально близько емулює реальне середовище планованого використання програмного продукту. Такі середовища можуть використовуватися для виконання автоматичних побудов, автоматизації тестів та виконання розробленого коду.

Модульне тестування призначено для перевірки правильності функціонування методів класів ПЗ. Модульні тести пишуться і виконуються розробниками в процесі написання коду. Модульне тестування застосовується як для перевірки якості коду програми, так і для перевірки об'єктів баз даних.

## **Виконання роботи**

### **Планування ітерацій**

Планування спринту проводять на зборах з планування. На зборах власник продукту обговорює з командою проекту користувальницькі опису функціональності (Елементи заділ роботи продукту), встановлюють пріоритети робочим елементам, формують список робіт спринту (Доробок роботи спринту), визначають завдання для реалізації користувальницьких описів функціональності, проводять оцінку складності реалізації завдань або трудовитрат і розподіляють завдання між виконавцями. Результатом зборів з планування спринту має бути визначення мети спринту, умов приймання та перелік робіт спринту. Мета спринту визначає склад тих робіт за проектом, які виконає команда по закінченню спринту, тобто які користувацькі вимоги будуть реалізовані. Умови приймання повинні визначати чітко розуміння того, що означає готовність кожного робочого елемента. Перелік робіт спринту визначає набір користувальницьких вимог і завдань, включених в поточний спринт (Доробок роботи спринту).

### **Оцінка обсягу / складності елементів роботи**

Після узгодження із зацікавленими особами користувача вимог, схем варіантів використання і класів необхідно сформувати перелік робіт для першого спринту. Власник продукту, разом з командою проекту, оцінює Елементи заділ роботи продукту і призначають їм пріоритети.

Оцінки елементів роботи необхідні для визначення того, скільки команда можемо виконати робіт в спринті. Оцінка обсягу / складності елементів роботи є абстрактною

метрикою, яка відображає думку членів команди щодо труднощів реалізації конкретних елементів роботи. Для оцінки Елементів заділ роботи продукту можна використовувати метод покеру планування. У методі покеру планування кожен член команди пропонує свою оцінку обсягу / складності робіт і в процесі обговорення команда повинна прийти до загальної, узгодженої оцінки. У покері планування кожен член команди має набір карток з числами. Числа представляють оцінку обсягу / складності роботи для робочого елемента. У більшості випадків картки покеру планування являють собою послідовність чисел Фібоначчі (1, 2, 3, 5, 8, 13 і т.д.). При оцінці конкретного елемента роботи члени команди одночасно показують обрані ними картки з певним числом. Ті члени команди, які показали найвищу і найнижчу оцінку, пояснюють свій вибір. Після поточного обговорення команда проекту проводить наступний раунд покеру планування і так далі, поки не буде виведена оцінка, з якою всі погодяться. Основна мета методики покеру планування - ініціювати обговорення обсягу / складності елементів роботи для з'ясування загального розуміння того, що означає готовність Елемента заділ роботи продукту. У разі якщо оцінка обсягу / складності елементів роботи дуже висока, то це говорить про те, що для даної роботи неясна реалізація і по ній необхідно зібрати додаткову інформацію або провести дослідження можливості реалізації.

Сформовані оцінки обсягу / складності елементів роботи фіксуються в поле Робота вкладки Елемент заділ роботи продукту (рис. 13.1).

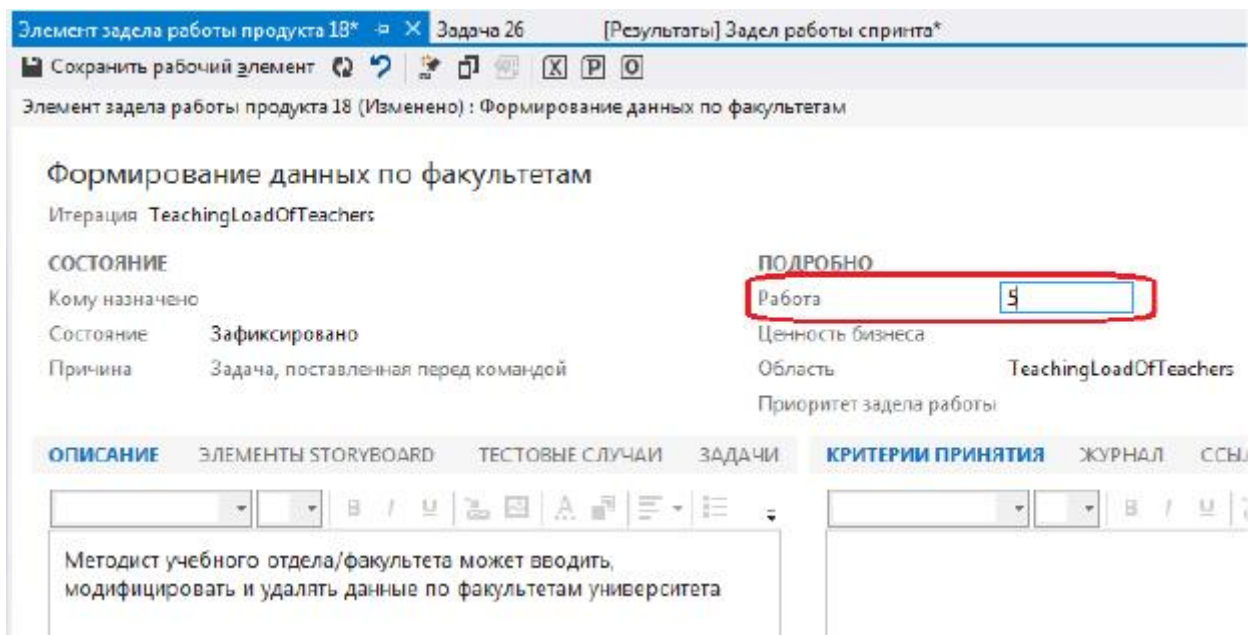


Рис. 13.1. Вкладка Елементу заділ роботи продукту

### *Призначення пріоритетів робочим елементам за допомогою Visual Studio*

Після оцінки елементів роботи доцільно встановити для них пріоритети. Це можна зробити за допомогою Visual Studio і за допомогою веб доступу до сайту проекту.

При використанні Visual Studio в командному браузері проекту на вкладці Робочі елементи виберіть пункт невиконані роботи по продукту (рис. 13.2).

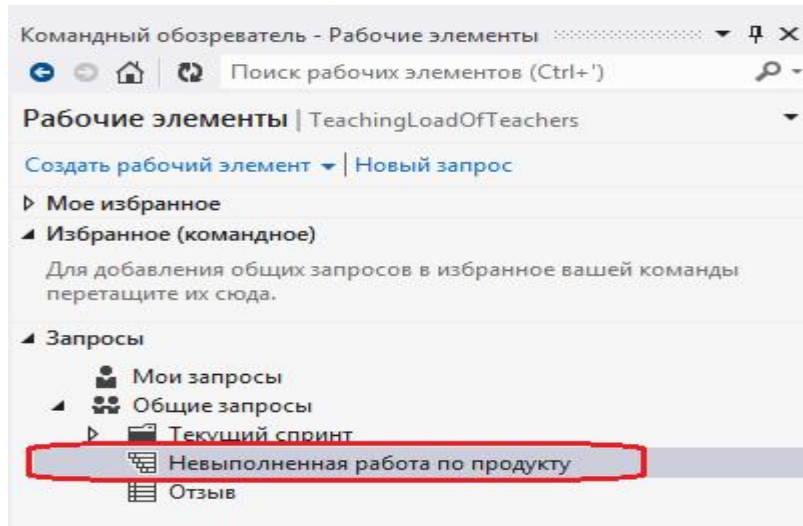


Рис. 13.2. Вкладка Рабочі елементи командного оглядача

Для обраного Елемента заділ роботи продукту призначте пріоритет (рис. 13.3).

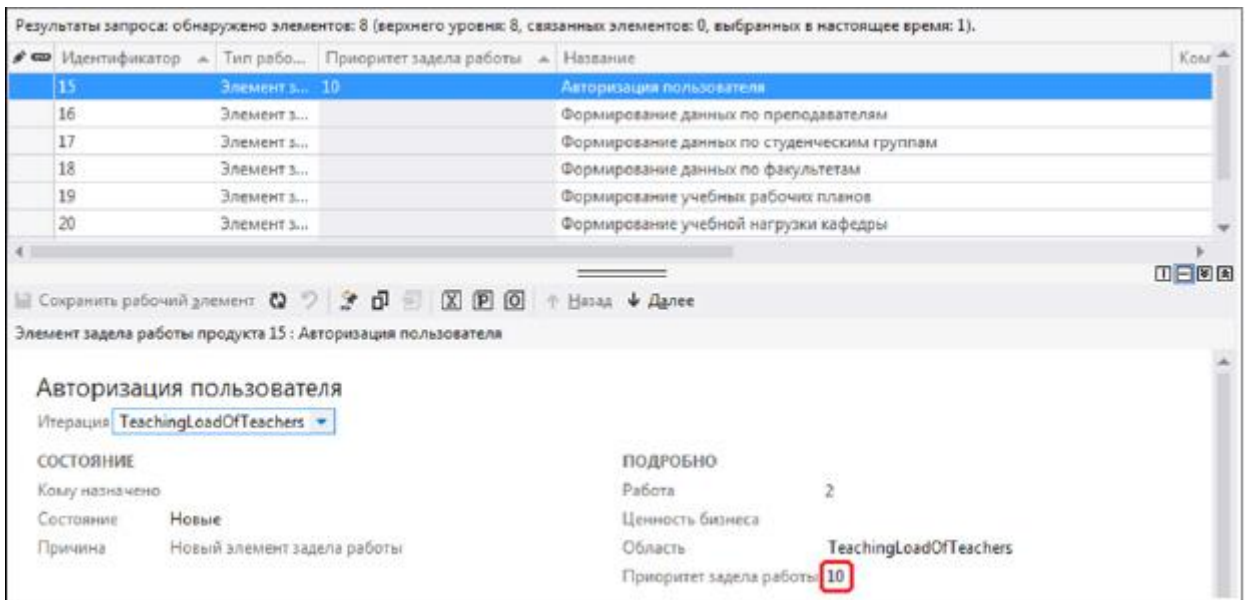


Рис. 13.3. Призначення пріоритету Еелементу заціпу роботи продукту

Пріоритети Елементам заділ роботи продукту встановлюються виходячи з міркувань доцільною послідовності етапів розробки проекту і рівня опрацьованості користувача вимог. Для Елементів заділ роботи продукту будемо встановлювати пріоритет 10, якщо їх не збираємося включати в поточний спринт. Для Елементів заділ роботи продукту, які плануємо включити до поточного спринт, встановимо пріоритет 1, також призначимо цього робочого елемента виконавця і встановимо стан Зафіксовано. Стан робочого елемента Зафіксовано означає, що даний елемент включений в поточний спринт і не підлягає зміні в спринті (рис. 13.4).

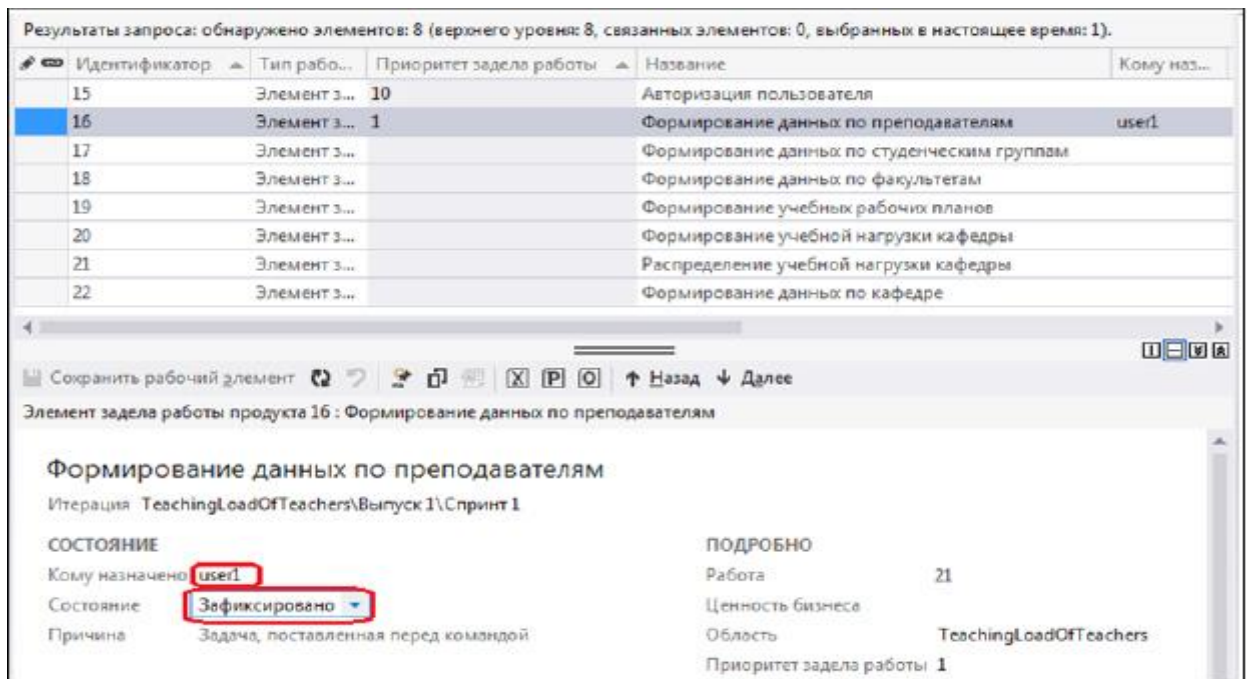


Рис. 13.4. Призначення виконавця Елементу заділу роботи продукту

Зроблені зміни в робочих елементах проекту необхідно зберегти.

### Призначення пріоритетів робочим елементам за допомогою веб доступу

Відкрийте сайт проекту, виберіть у головному меню пункт РОБОТА і на вкладці невиконані роботи виберіть Відставання продукту (рис. 13.5).

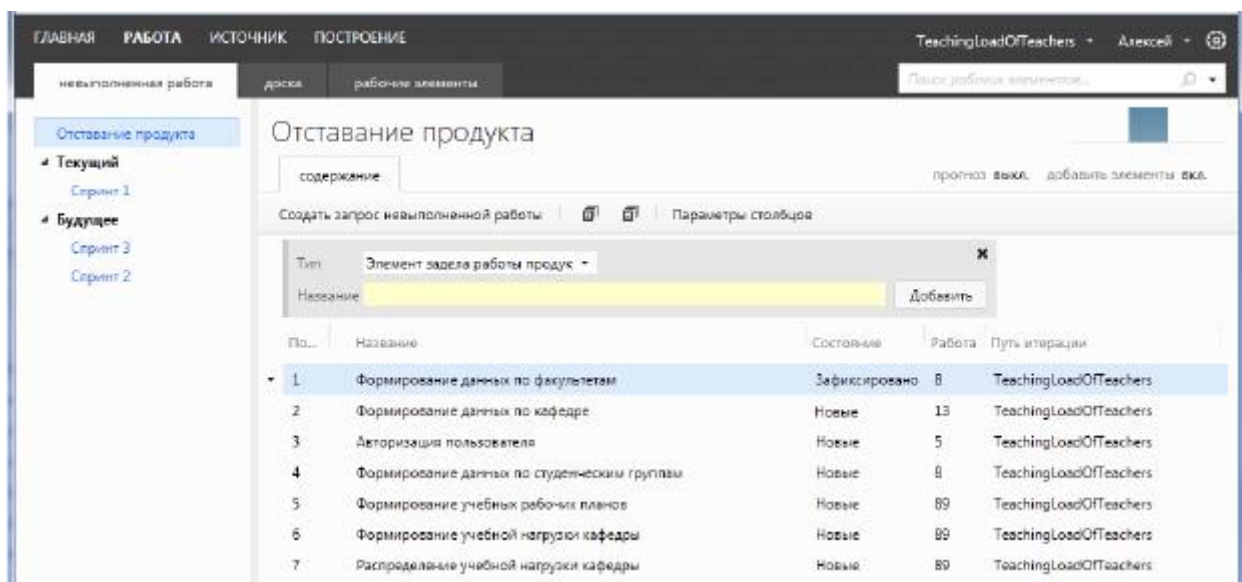


Рис. 13.5. Вкладка невиконані роботи - Відставання продукту

При подвійному натисканні на елементі заділу роботи продукту відкривається діалогове вікно (рис. 13.6), в якому можна встановити ітерацію, виконавця, стан і пріоритет (параметр Причина встановлюється відповідно до стану робочого елемента автоматично).



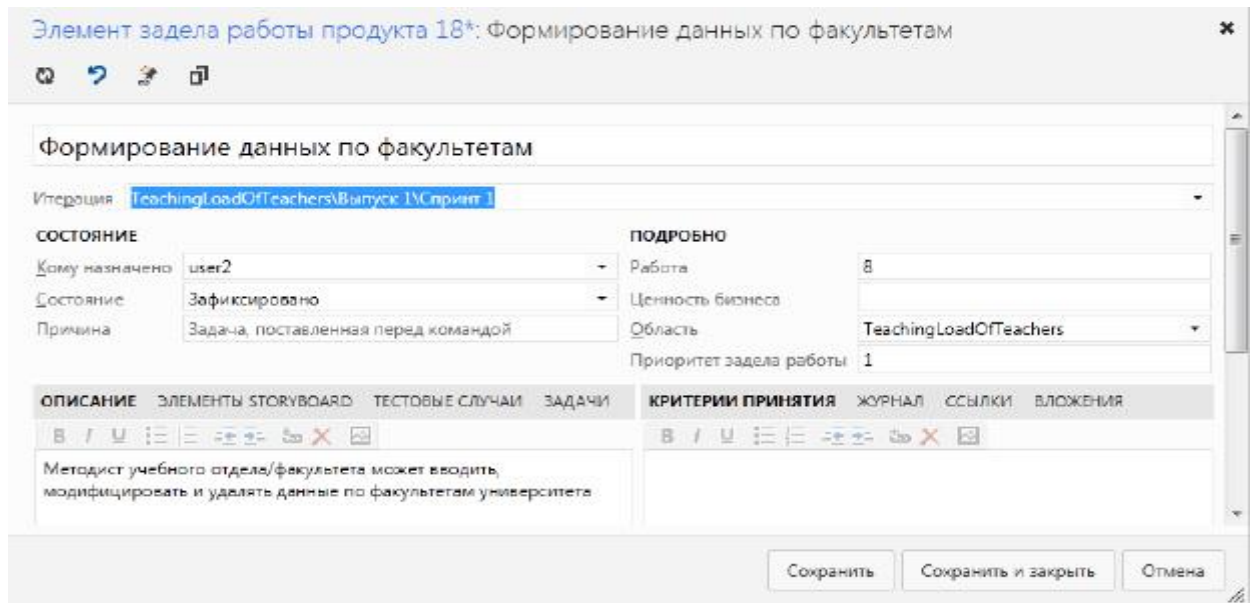


Рис. 13.6. Діалогове вікно Елемент заділу роботи

При призначенні Елементам заділу роботи продукту конкретної ітерації вони видаляються зі списку Відставання продукту, тобто з робочого елемента невиконані роботи по продукту (рис. 13.7).

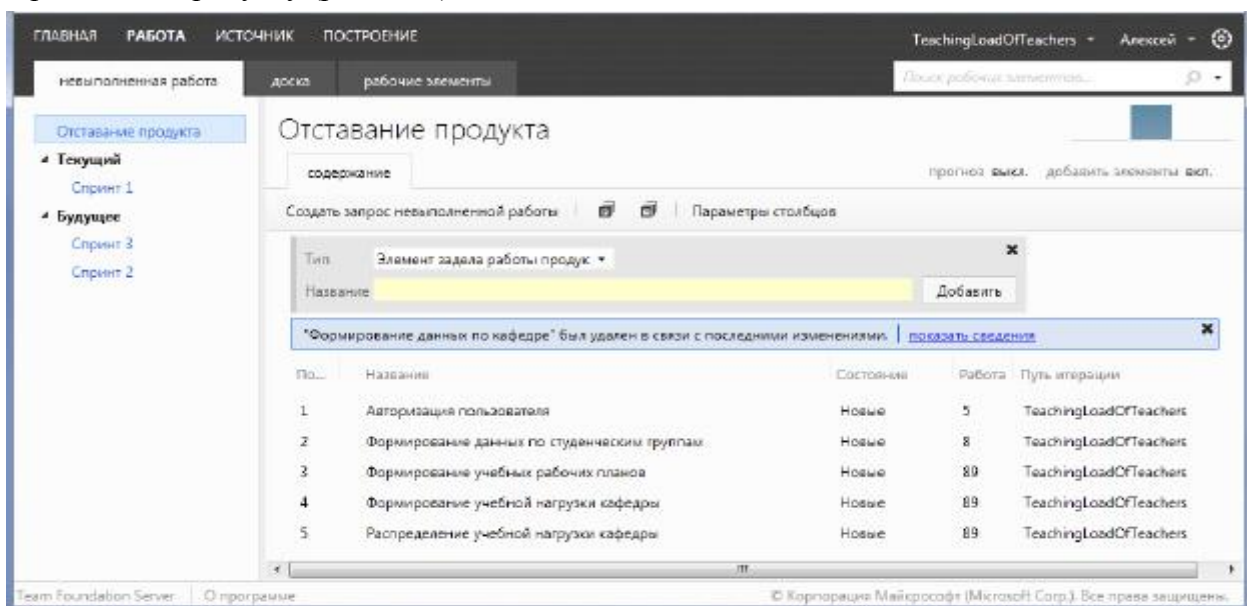


Рис. 13.7. Змінена вкладка Відставання продукту

Після призначення всім Елементом заділ роботи продукту пріоритетів та завдання ітерації для високопріоритетних елементів проводиться нарада команди проекту з планування поточного спринту.

Планування завдань спринту При плануванні спринту необхідно для кожного Елементу заділу роботи продукту визначити завдання, призначити їм виконавця та провести оцінку трудовитрат. Завдання являють собою роботу, яку необхідно виконати для реалізації користувальницьких вимог. Виберіть на вкладці невиконані роботи посилання Спринт 1 (рис. 13.8)

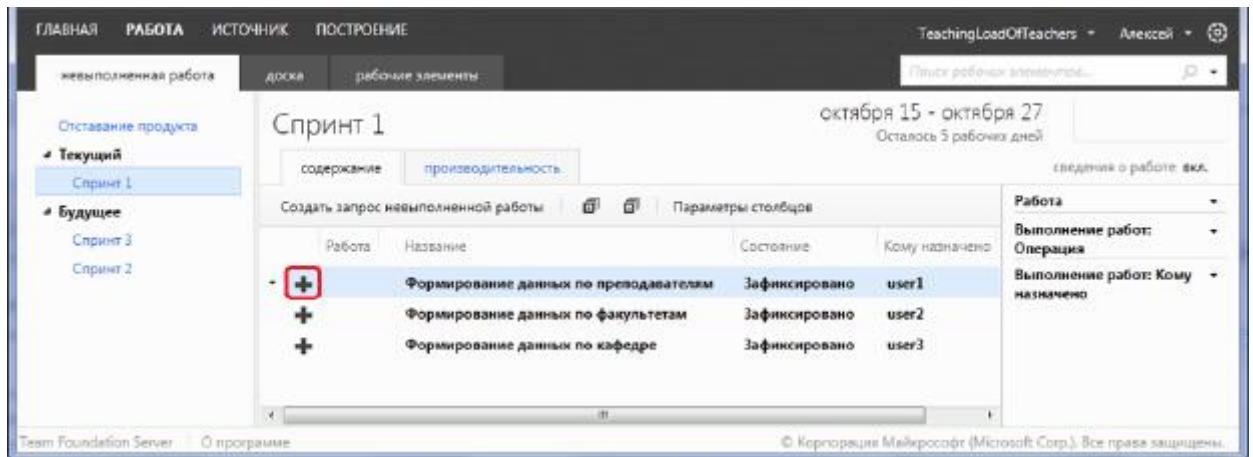


Рис. 13.8. Вкладка Спринт 1

На вкладці Спринт 1 відображаються робочі елементи, які включені в даний спринт. Для Елементів заділ роботи продукту необхідно створити завдання, які необхідні для реалізації користувальницьких вимог. Створення завдання для Елементу заділ роботи продукту можна виконати при натисканні кнопки в графі робота поточного спринту (рис. 13.8). Діалогове вікно Створення завдання (рис. 13.9) дозволяє ввести найменування завдання, визначити її стан та інші параметри.

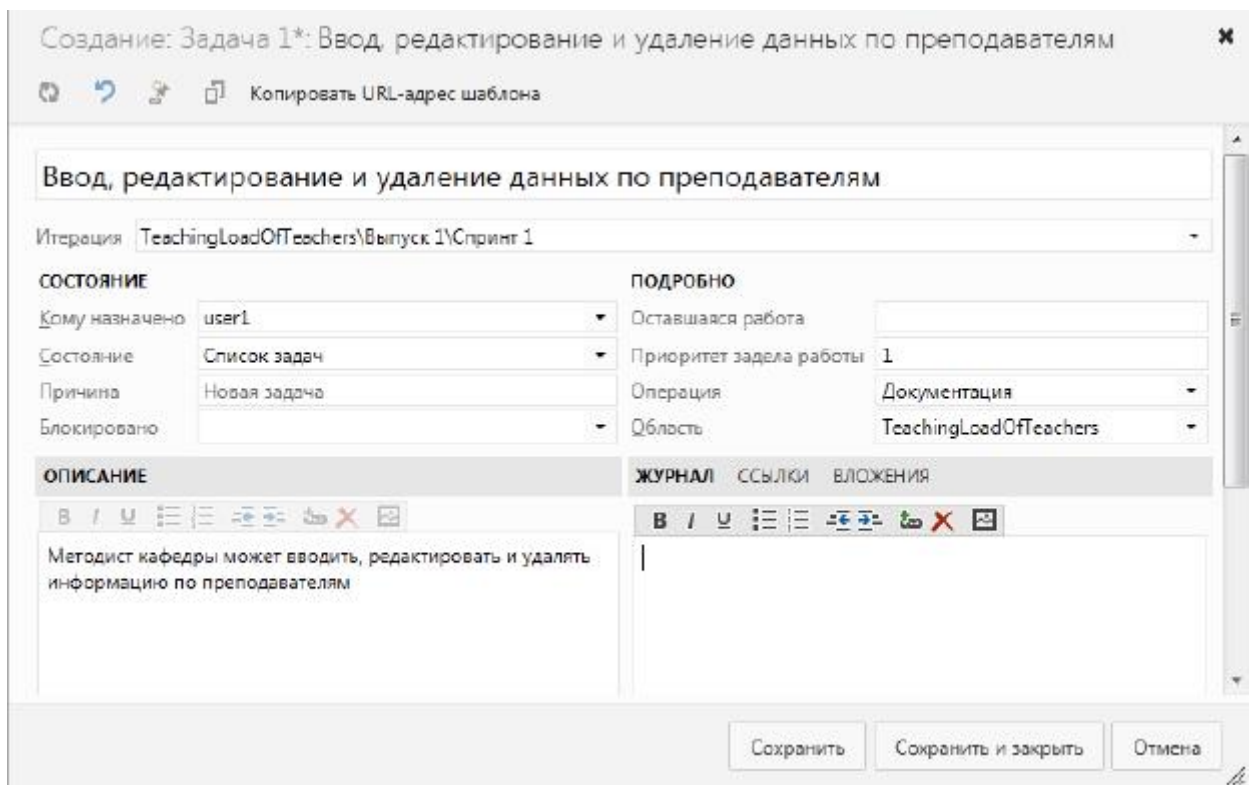


Рис. 13.9. Діалогове вікно Створення завдання

Дані по завданню відображаються у вкладці Спринт 1 після її збереження (рис. 13.10).

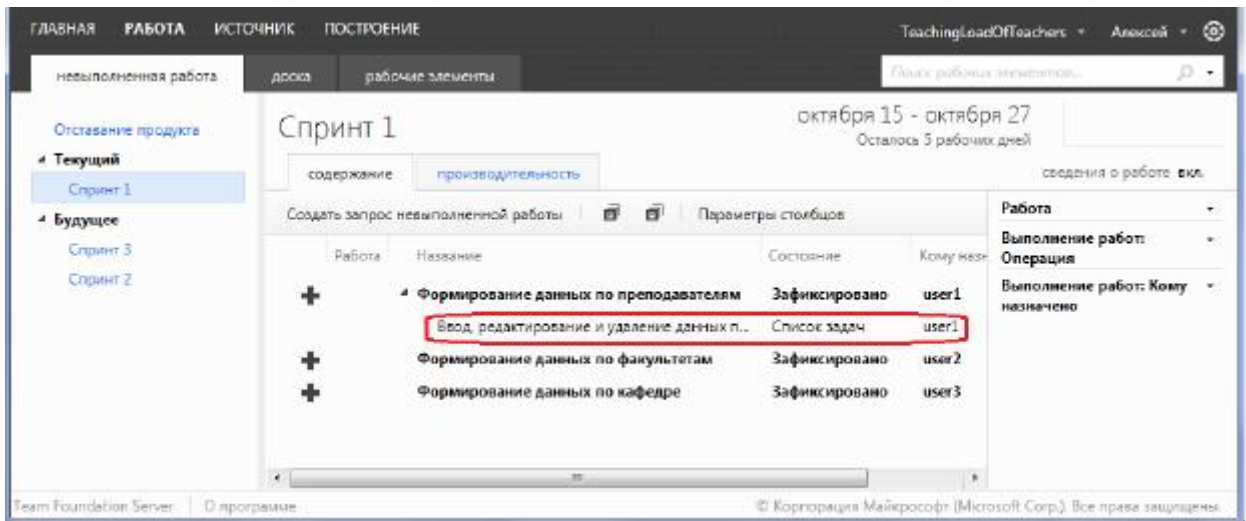


Рис. 13.10. Вкладка Спринт 1

Створення завдань для користувача вимог (Елементу заділ роботи продукту) можна проводити і за допомогою Visual Studio. Для цього в командному браузері виберіть пункт Доробок робіт спринту (рис. 13.11).

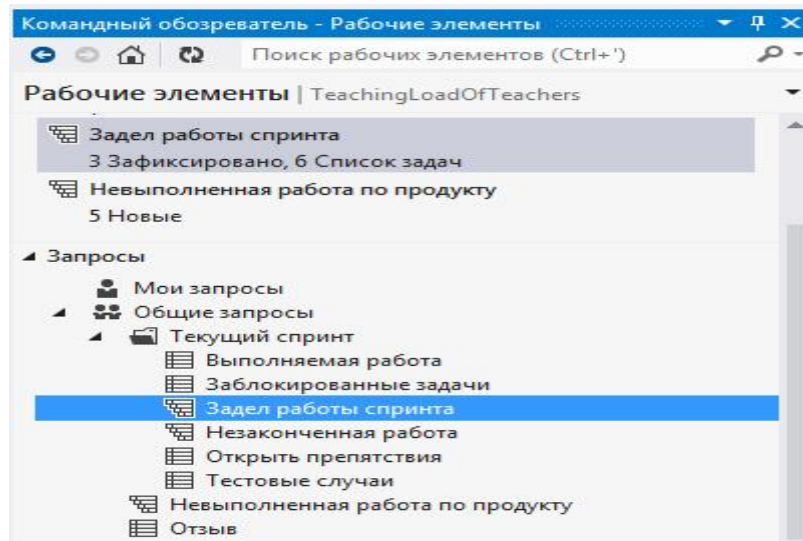


Рис. 13.11. Вибір робочого елементу Доробок робіт спринту

На панелі Доробок роботи спринту (рис. 13.12) виділіть Елемент заділ роботи продукту (1) і або натисніть кнопку Створити пов'язаний робочий елемент (2) панелі інструментів, або на вкладці ЗАВДАННЯ (3) виберіть кнопку Створити (4).

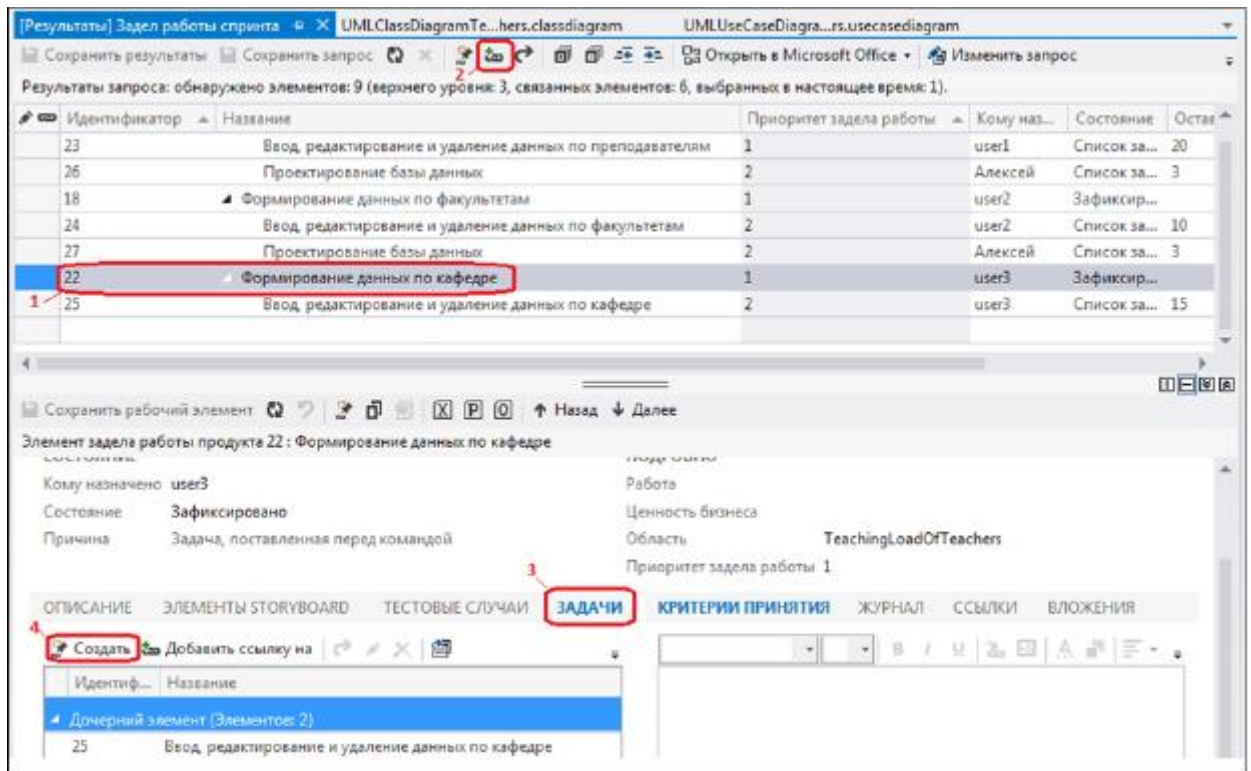


Рис. 13.12. Вкладка Доробок работи спринту

У діалоговому вікні (рис. 13.13) введіть назву (1) і коментар до створюваної задачі (2). Після натискання кнопки ОК відкриється діалогове вікно Створення завдання, аналогічне наведеному на рис. 13.9. Діалогове вікно Створення завдання дозволяє ввести найменування завдання, визначити її стан та інші параметри.

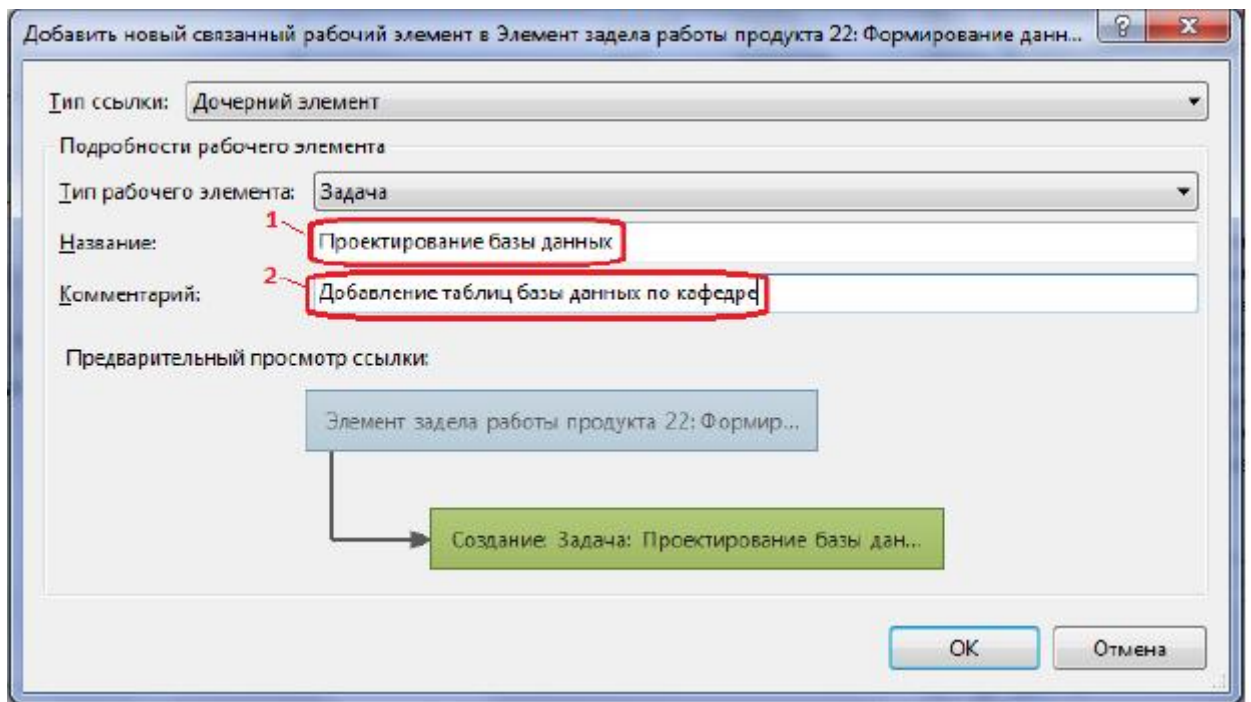


Рис. 13.13. Діалогове вікно Додати новий пов'язаний робочий елемент



Для всіх користувальницьких вимог команда проекту повинна визначити необхідні завдання. У Visual Studio робочі елементи поточного спринту представляються на вкладці Доробок робіт спринту (рис. 13.14), а на сайті проекту - на вкладці Спринт 1 (рис. 13.14).

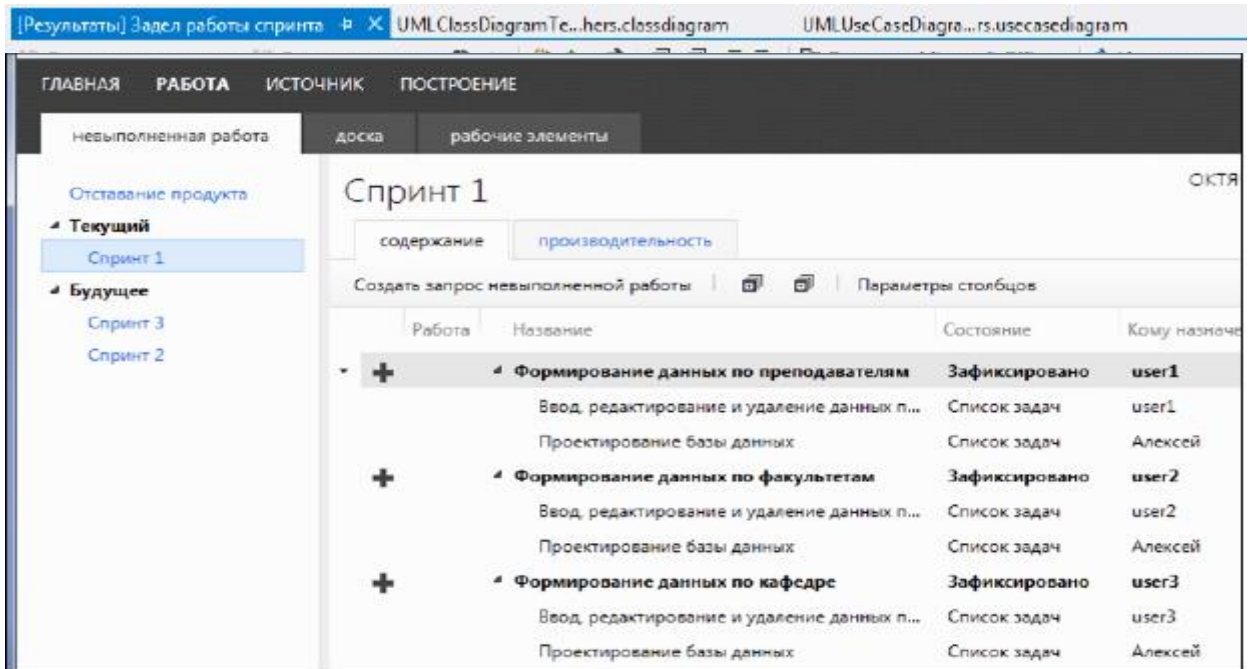


Рис. 13.14. Рабочі елементи поточного спринту

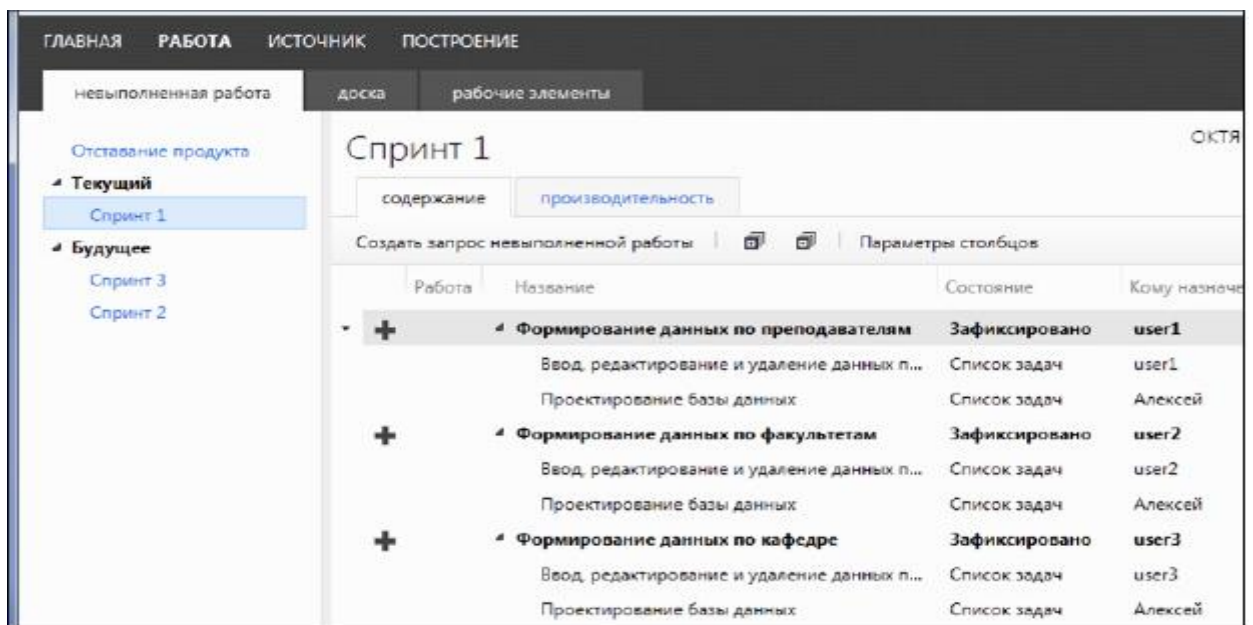


Рис. 13.15. Перелік робіт Спринт 1

Для завдань поточного спринту необхідно провести оцінку трудовитрат. При плануванні витрат часу на виконання завдань команда проекту робить оцінки необхідного часу з урахуванням обсягу / складності елементів роботи. Витрати часу плануються у годинах і додаються до опису завдань (рис. 13.16).

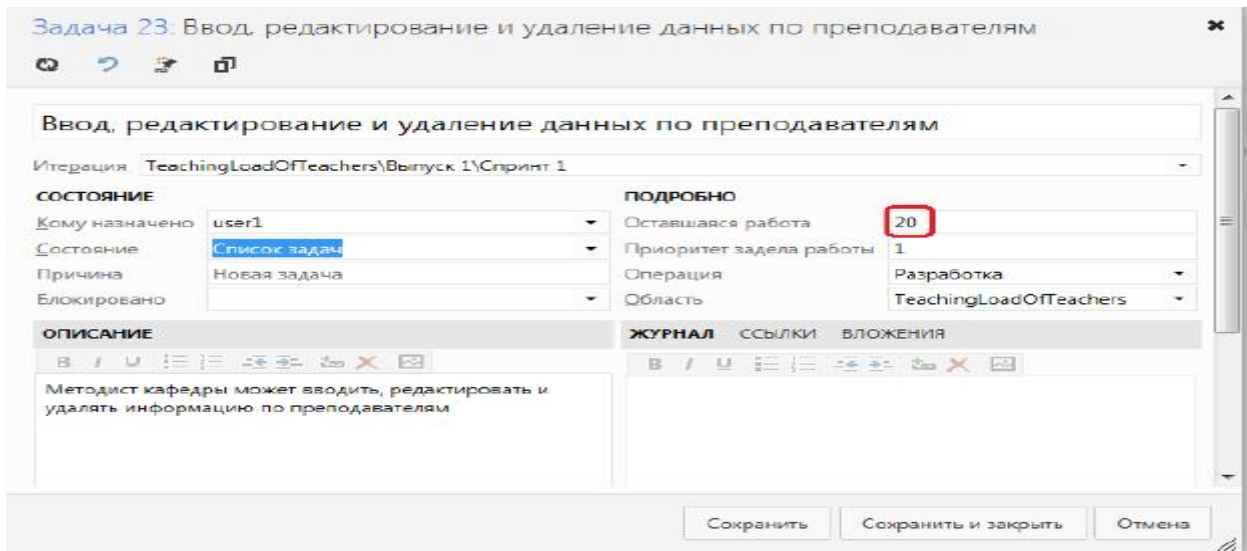


Рис. 13.16. Завдання часу виконання завдання

Результати оцінки витрат часу на реалізацію завдань спринту 1 наведено на рис. 13.17.

Работа	Название	Состояние	Кому назначено	Оставшаяся работа
21	Формирование данных по преподавателям	Зафиксировано	user1	23
	Ввод, редактирование и удаление данных п...	Список задач	user1	20
	Проектирование базы данных	Список задач	Алексей	3
8	Формирование данных по факультетам	Зафиксировано	user2	13
	Ввод, редактирование и удаление данных п...	Список задач	user2	10
	Проектирование базы данных	Список задач	Алексей	3
13	Формирование данных по кафедре	Зафиксировано	user3	19
	Ввод, редактирование и удаление данных п...	Список задач	user3	15
	Проектирование базы данных	Список задач	Алексей	4

Рис. 13.17. Вкладка Спринт 1 - результаты оцінки витрат часу

Далі плануємо завантаження членів команди (рис. 13.18). Для цього задається час роботи в день над проектом та робочі дні для ітерації (фактично задаються як вихідні, ті дні, коли член команди не працює над проектом).

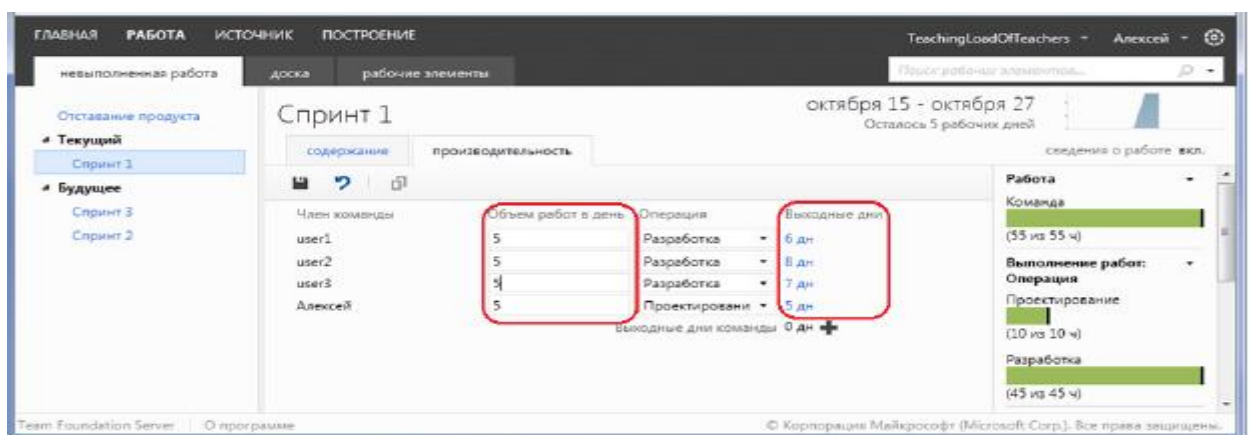


Рис. 13.18. Вкладка продуктивність для Спринту 1

На дощці проекту можна подивитися результати планування в розрізі елементів заділу роботи (рис. 13.19) і в розрізі членів команди (рис. 13.20).

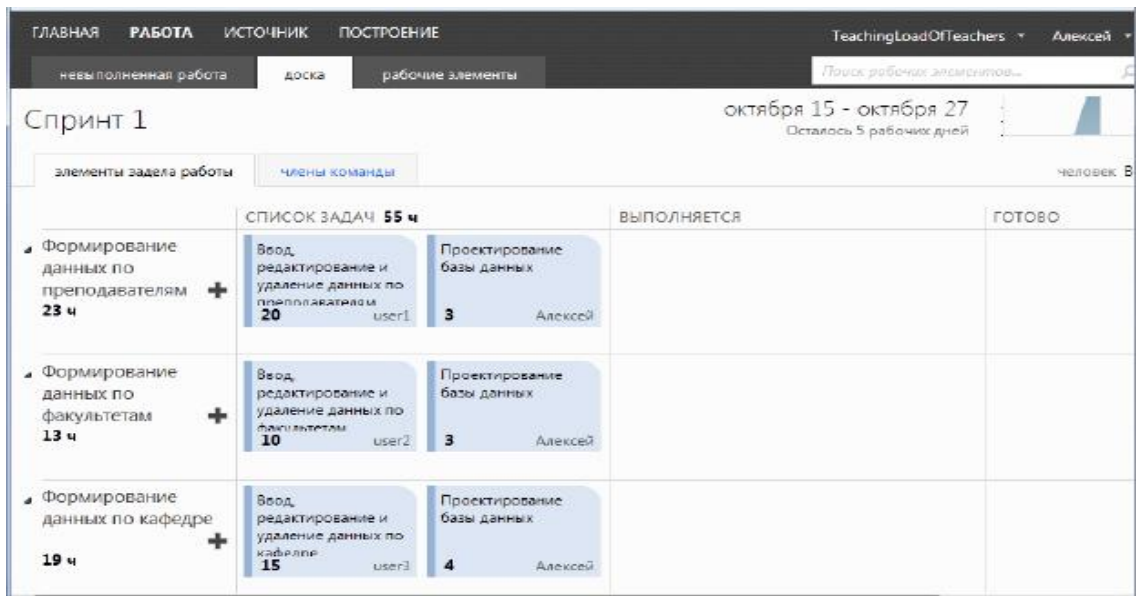


Рис. 13.19. Дощка - элементы заділ робіт Спринт 1

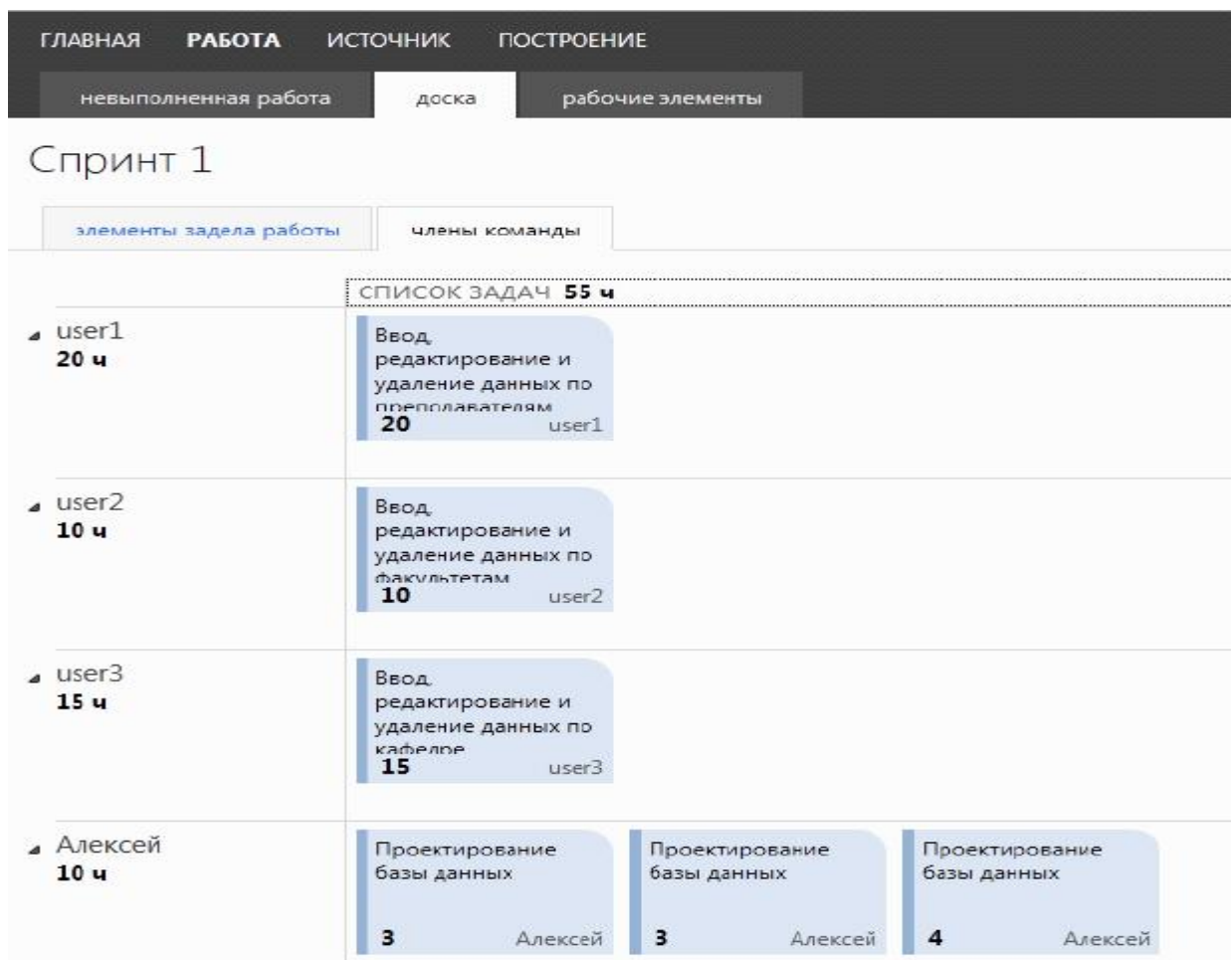


Рис. 13.20. Дощка - члены команды Спринт 1

Слід зазначити, що члени команди добровільно беруть на себе зобов'язання виконати роботу протягом спринту. Перш ніж це зробити команда повинна проаналізувати необхідні часові витрати і часовий ресурс команди. Команда ніколи не повинна брати на себе зобов'язання виконати роботу, яку, на думку її членів, вона виконати не може.

## **Контрольні питання**

1. Які інструменти маються на TeamFoundationServer для управління командними проектами?
2. Які папки генеруються при створенні командного проекту?
3. Для чого використовуються у проекті області та ітерації?
4. Які робочі елементи наявні в шаблоні Agile?
5. За допомогою яких клієнтських інструментів можна проводити планування командного проекту?
6. Поясніть процес створення коду розробником командного проекту в середовищі VisualStudio.

## **Завдання**

Ознайомитися з теоретичним матеріалом. Для розроблюваного проекту, згідно з варіантом завдання провести оцінку складності елементів робіт, встановити пріоритети робочим елементам. Для робочих елементів, включених в поточний спринт, визначити завдання. Провести оцінку трудовитрат для задач спринту. Розподілити завдання спринту між членами команди.

Звіт з лабораторної роботи повинен складатися з:

1. Постановки завдання.
2. Документу «Протокол лабораторної роботи», що містить:
  - Назву роботи. Мету;
  - Хід виконання роботи (графічні результати виконання роботи);
  - Відповіді на контрольні питання;
  - Висновки.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.



## Лабораторна робота №9

### Розробка документації. Підготовка інсталяційного пакету

**Мета:** Отримати практичні навички розробки презентацій для представлення користувача інтерфейсу всім зацікавленим особам. Створити розкадрування для додатку, що розробляється.

### Теоретичні відомості

Зручним інструментом макетування користувача інтерфейсу реалізації вимог користувача є PowerPoint, який інтегрований з Visual Studio 2012. Для Елементів заділ роботи продукту є можливість створити розкадрування подання екранних форм або веб-сторінок додатку. Розкадрування призначена для попереднього обговорення основних рішень для користувача інтерфейсу.

### Виконання роботи

Для створення розкадровки необхідно відкрити Елементи заділ роботи продукту. Це можна зробити на веб-сайті проекту (рис. 14.1) або з дизайнера командних проектів Visual Studio (рис. 14.2)

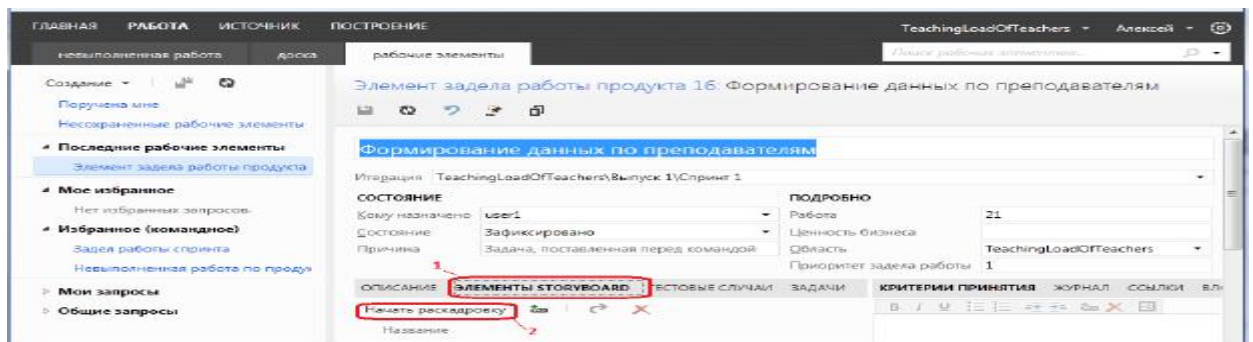


Рис. 14.1. Веб-сайт проекта - Элемент заділ роботи продукту

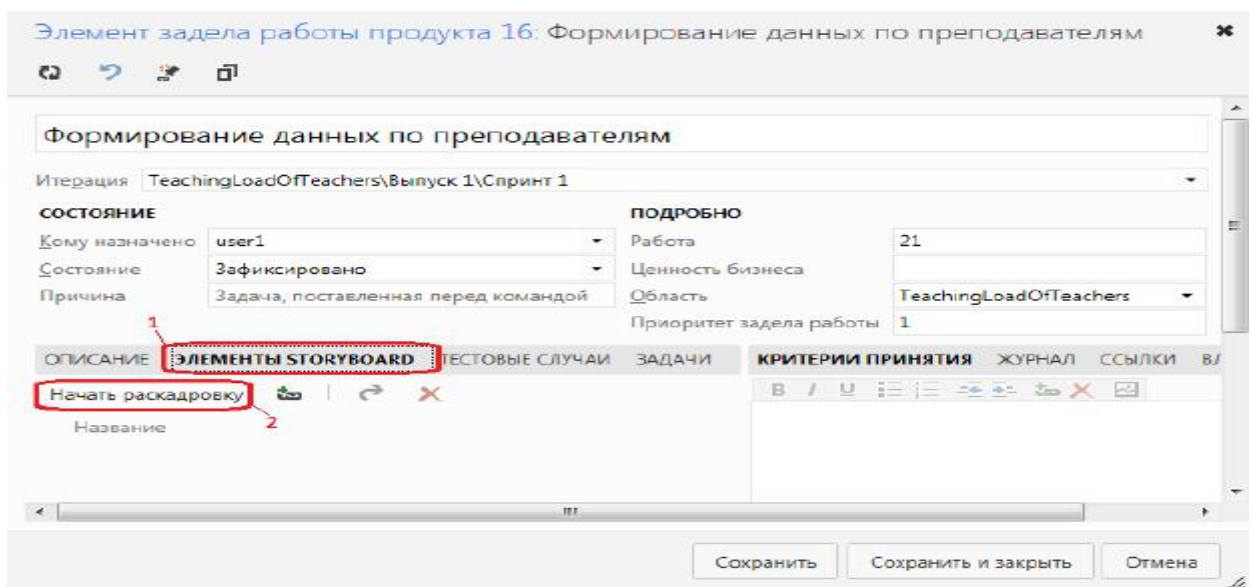


Рис. 14.2. Visual Studio - Элемент заділ роботи продукту

У вікні Елементів заділ роботи продукту (рис. 14.1, рис. 14.2) виділіть вкладку ЕЛЕМЕНТИ STORYBOARD (1) і виберіть посилання Почати розкадрування (2). презентація інтерфейсу Початкова сторінка PowerPoint наведена на рис. 14.3. Використовуючи фігури розкадровки, створіть слайд макета необхідних Windows-вікон або веб-сторінок розробляється. На рис. 14.4 наведено приклад веб-сторінки Викладачі для проекту Розрахунок навчального навантаження викладачів. Створену презентацію користувальницького інтерфейсу додатку, необхідно зберегти для забезпечення до неї доступу всіх зацікавлених осіб. Збереження файлу розкадровки необхідно провести в мережевій папці або на сайті SharePoint.

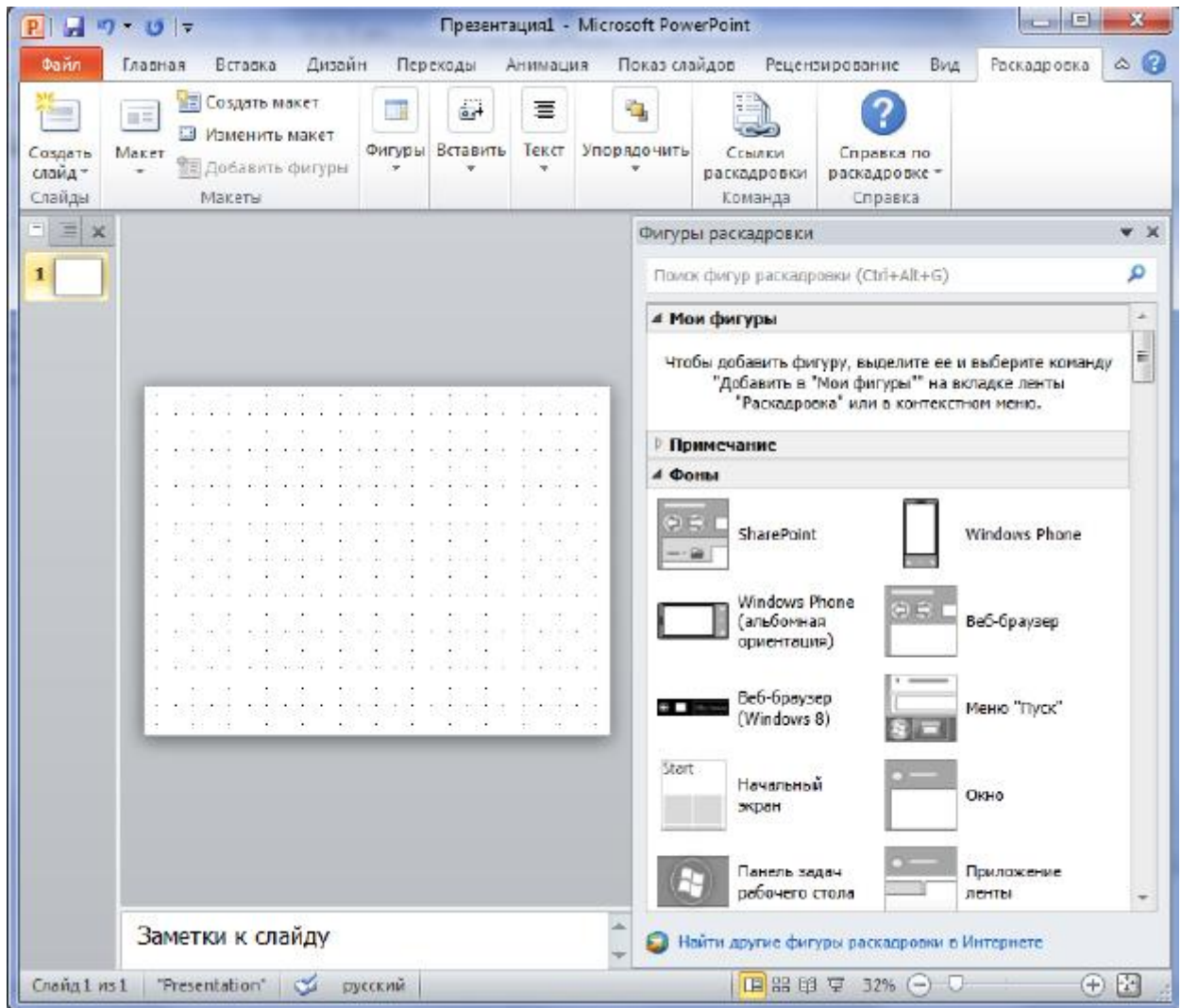


Рис. 14.3. Початкова сторінка PowerPoint

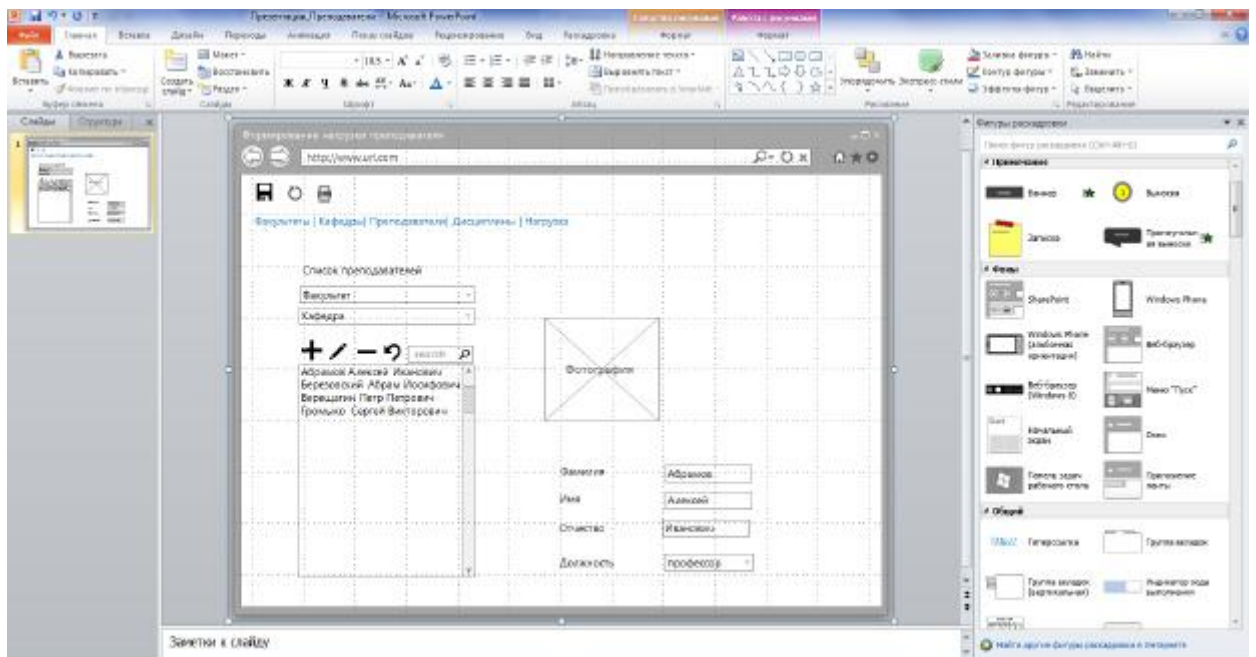


Рис. 14.4. Сторінка - Викладачі

Для зв'язку створеної презентації розкладовки з командним проектом в TFS необхідно на панелі інструментів PowerPoint вибрати команду Посилання розкладовки (рис. 14.5).

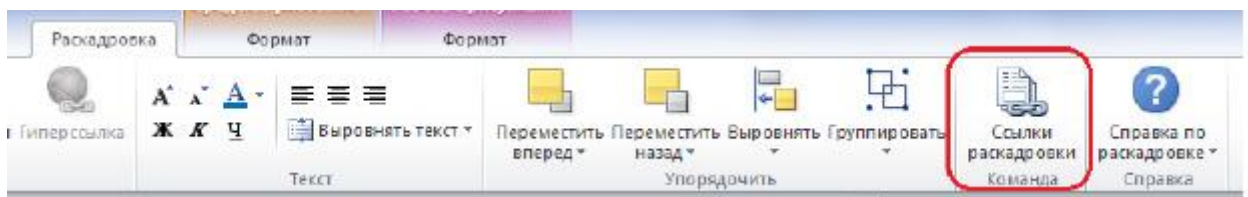


Рис. 14.5. Панель команд PowerPoint

Якщо презентація PowerPoint ще не пов'язана з командним проектом, то в діалоговому вікні Посилання розкладовки в проекті необхідно натиснути кнопку Підключити (1) для підключення до командного проекту (рис. 14.6).

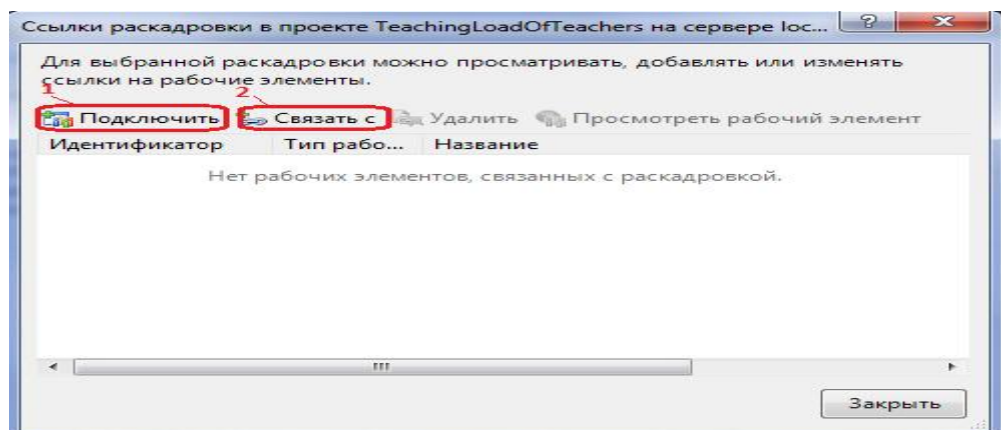


Рис. 14.6. Діалогове вікно Посилання розкладовки в проекті

У діалоговому вікні Підключення до командного проекту слід вибрати ім'я командного проекту (рис. 14.7).

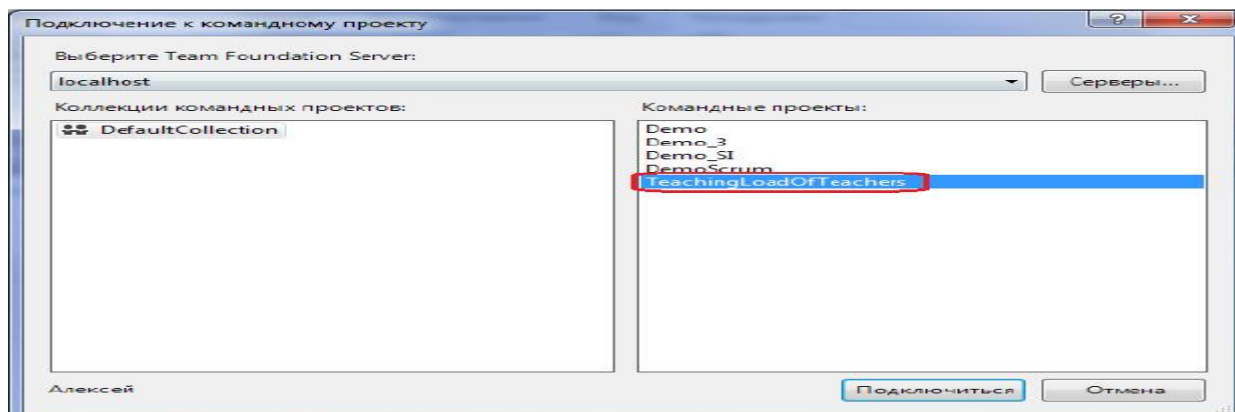


Рис. 14.7. Підключення до командного проекту

Після підключення до командного проекту вдруге буде виведено діалогове вікно Посилання розкладовки в проекті (рис. 14.6), в якому потрібно натиснути кнопку Зв'язати (2). У діалоговому вікні Вибір пов'язаних робочих елементів необхідно сформулювати запит не вибір робочого елемента, який буде пов'язаний з розкладовкою, і відзначити його галочкою у списку (рис. 14.8).

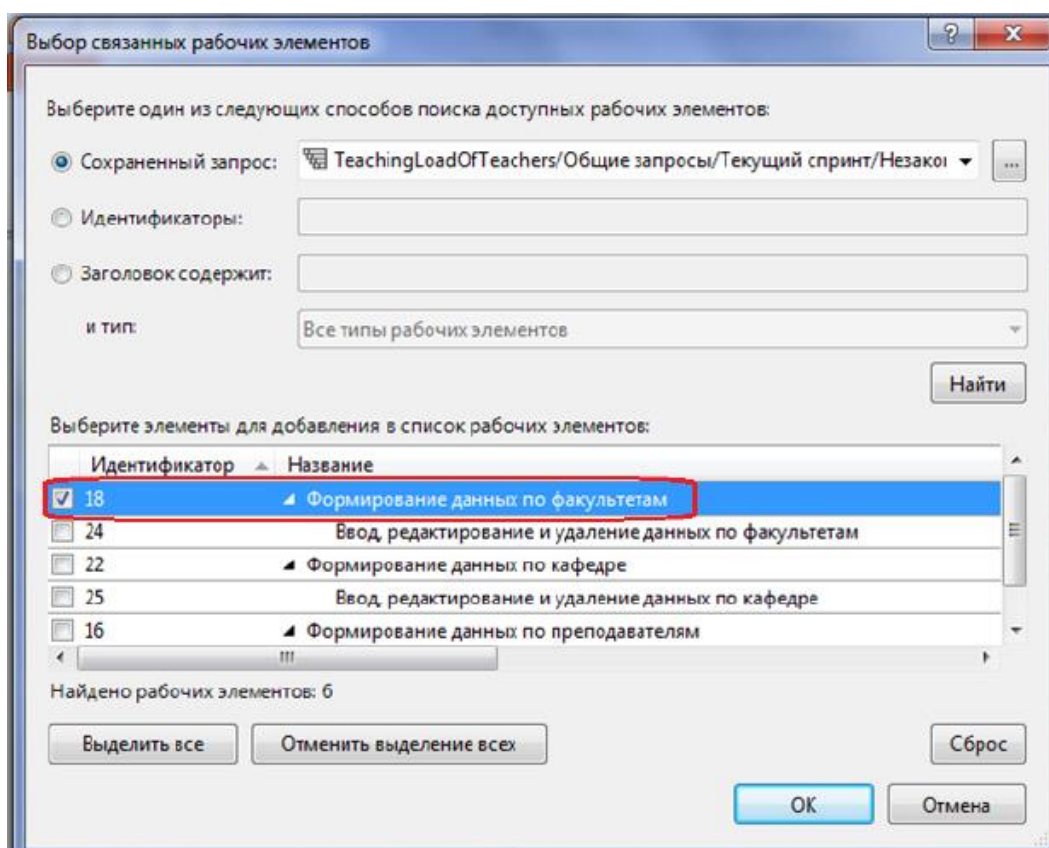


Рис. 14.8. Діалогове вікно Вибір пов'язаних робочих елементів

Після підтвердження вибраних елементів в діалоговому вікні Посилання розкладовки в проекті буде додано рядок з даними пов'язаного з розкладовкою робочого елемента (рис. 14.9).



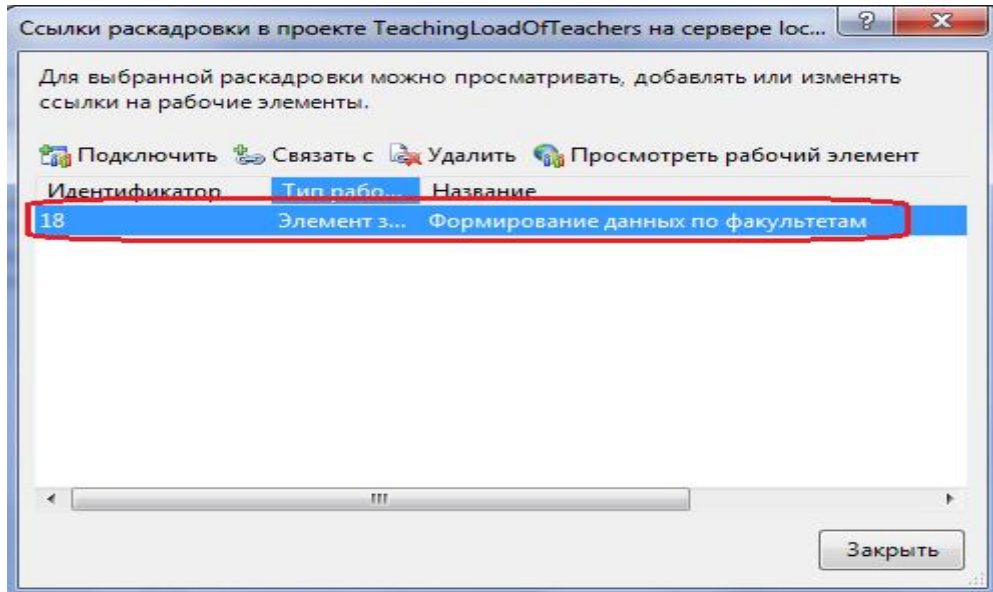


Рис. 14.9. Диалоговое вікно Посилання розкадровки в проектї з доданим елементом

Якщо робочий елемент пов'язаний з розкадровою, то це можна побачити на сайті проекту (рис. 14.10) або в оглядачі Visual Studio (мал. 14.11).

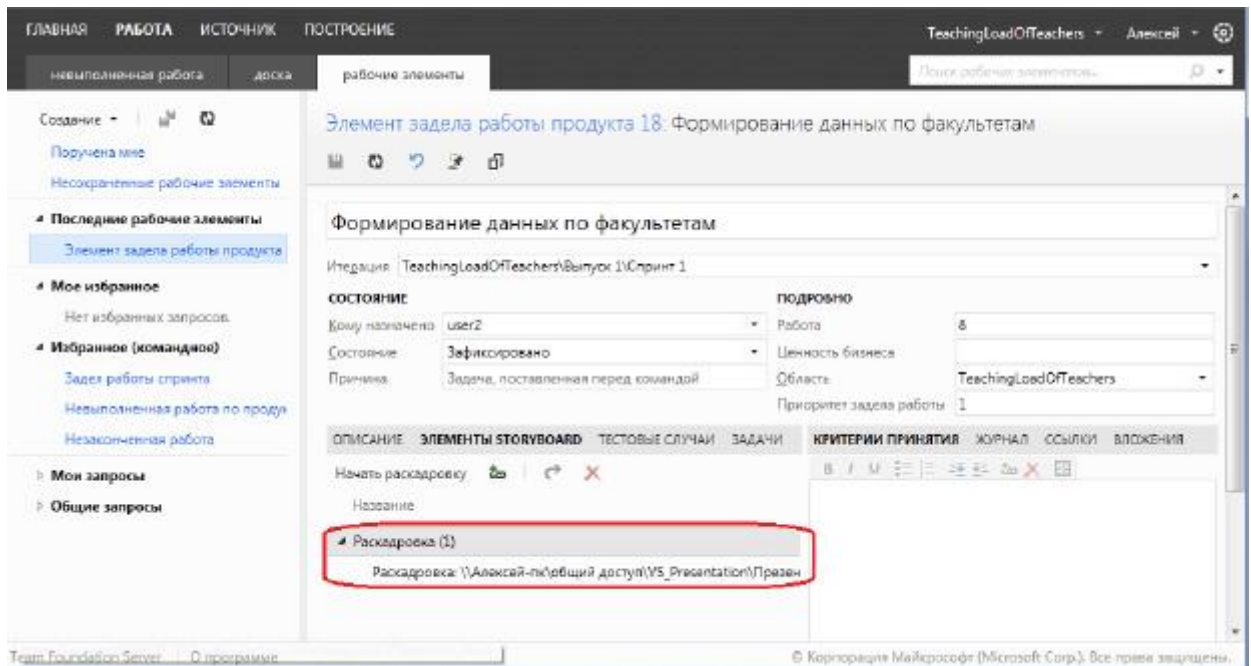


Рис. 14.10. Сайт проекту - закладка ЕЛЕМЕНТЫ STORYBOARD

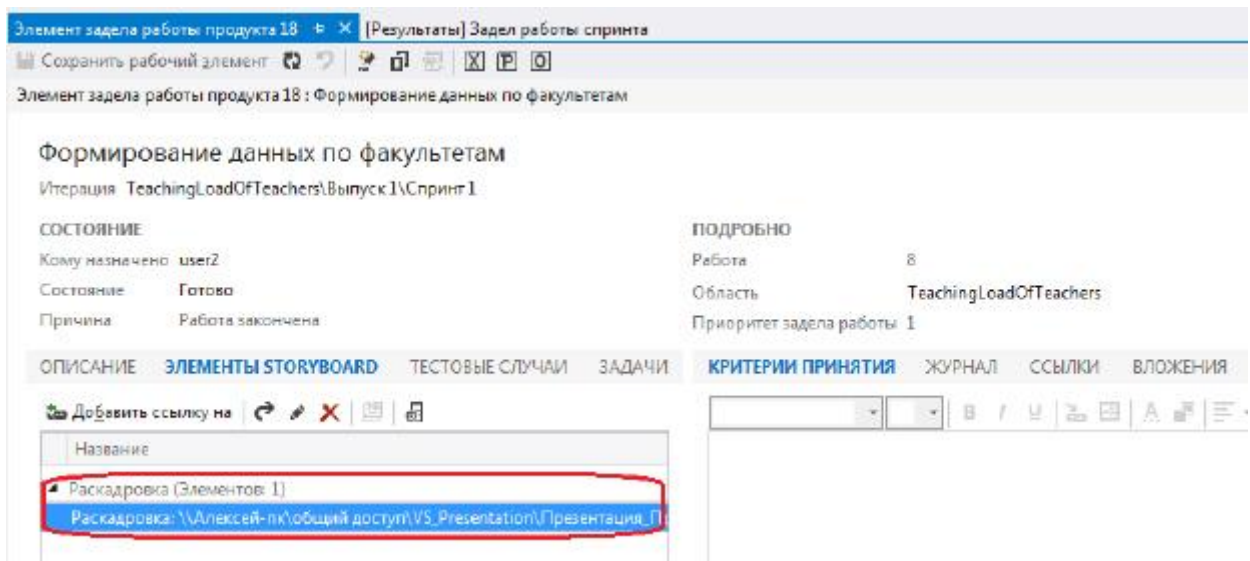


Рис. 14.11. Элемент задел работ продукта - закладка ЭЛЕМЕНТЫ STORYBOARD

Для просмотра раскадровки необходимо нажать кнопку - Открыть пов'язаний элемент на закладці ЭЛЕМЕНТЫ STORYBOARD (рис. 14.12).

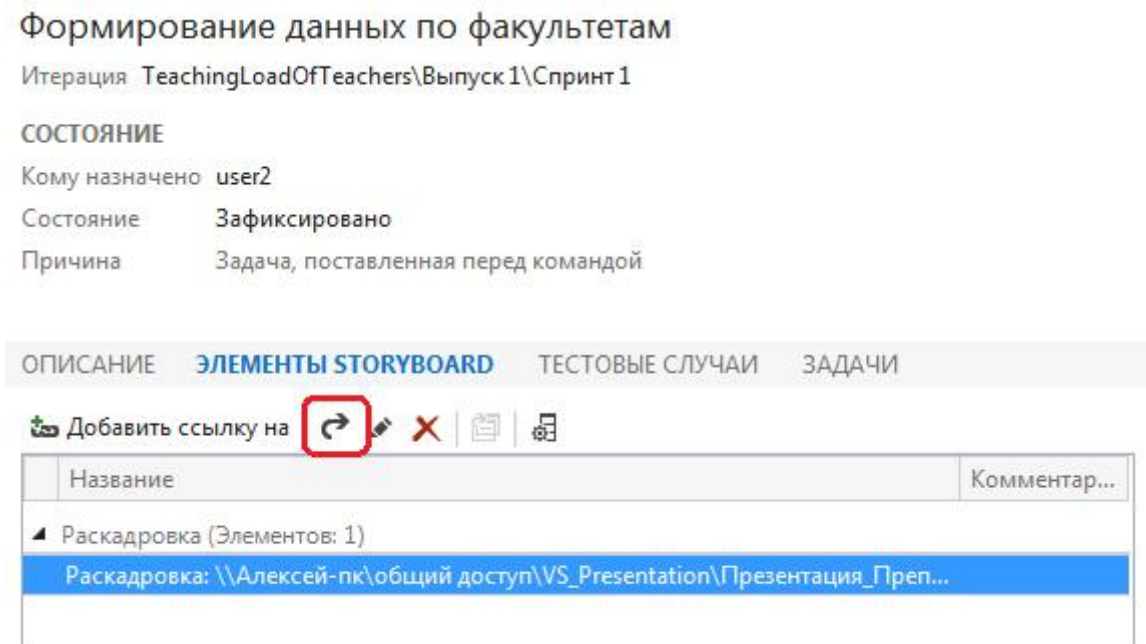


Рис. 14.12. Закладка ЭЛЕМЕНТЫ STORYBOARD

## **Контрольні питання**

1. Що являє собою командний проект в TFS.
2. Які шаблони проектів маються на TFS 2012?
3. Які робочі елементи визначені в TeamFoundationServer.
4. Які можливості надає система контролю версій TFS?
5. Які можливості має розробник для взаємодії з ключовими службами TeamFoundationServer?

## **Завдання**

Ознайомитися з теоретичним матеріалом. Для розроблюваного проекту, згідно з варіантом завдання провести, інструментом макетування користувача інтерфейсу PowerPoint, який інтегрований з Visual Studio 2012, розкадрування подання екранних форм або веб-сторінок додатку. Розкадрування призначена для попереднього обговорення основних рішень для користувача інтерфейсу.

Звіт з лабораторної роботи повинен складатися з:

1. Постановки завдання.
2. Документу «Протокол лабораторної роботи», що містить:
  - Назву роботи. Мету;
  - Хід виконання роботи (графічні результати виконання роботи);
  - Відповіді на контрольні питання;
  - Висновки.

Захист звіту з лабораторної роботи полягає в пред'явленні викладачеві отриманих результатів (на екрані монітора), демонстрації отриманих навичок і відповідях на питання викладача.



## Література

### Основна

1. Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул «Технология разработки программного обеспечения: учебное пособие» / под ред. Л.Г. Гагариной. – М.: ИД «Форум»: Инфра–М, 2008. – 400с.
2. Иванова Г.С. «Технология программирования: Учебник для ВУЗов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320с.
3. Брауде Эрик Дж. Технология разработки программного обеспечения. М.: Computer Science, 2004. – с. 655.
4. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева В.Ю. Суслов / Под ред. И.В. Сергиенко. – К.: Академперіодика. (изд. 2-е, перераб. и дополн.) К.: Академперіодика. – 2007 р. – 672 с.
5. С.А. Орлов Технологии разработки программного обеспечения. Разработка сложных программных систем. //ВНВ ПИТЕР. – 2004 г. – 527с.

### Додаткова

1. Иан Соммервилл. Инженерия программного обеспечения. Изд. дом «Вильямс». – 2002 г. – 624 с.
2. Якунин, Ю. Ю. Технологии разработки программного обеспечения. Версия 1.0 [Электронный ресурс] : лаб. практикум / Ю. Ю. Якунин, И. С. Почекутов, С. В. Якунина. – Электрон. дан. (3 Мб). – Красноярск : ИПК СФУ, 2008. – (Технологии разработки программного обеспечения : УМКД № 183-2007 / рук. творч. коллектива Ю. Ю. Якунин). – 1 электрон. опт. диск (DVD).
3. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения / Спб.: Питер, 2002. - 496 с.
4. Крачтен Ф. Введение в Rational Unified Process. — Изд. 2-е: Пер. с англ. // М.: Издательский дом “Вильямс”, 2002. — 240 с
5. Трофимов С.А. Rational XDE для Visual Studio .NET // М.: ООО «Бином-Пресс», 2004 г. — 304 с
6. Благодатских, В. А. Стандартизация разработки программных средств : учеб. пособие / В. А. Благодатских, В. А. Волнин, К. Ф. Посакалов ; под ред. О. С. Разумова. – М. : Финансы и статистика, 2003. – 288 с.