

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Завдання до лабораторних робіт та методичні
рекомендації до їх виконання з навчальної
дисципліни**

**"ОСНОВИ СИСТЕМНОГО АНАЛІЗУ
ОБ'ЄКТІВ ТА ПРОЦЕСІВ
КОМП'ЮТЕРИЗАЦІЇ"
(У СЕРЕДОВИЩІ RATIONAL REQUISITEPRO)**
**для студентів напряму підготовки "Комп'ютерні науки"
усіх форм навчання**

Харків. Вид. ХНЕУ, 2009

Затверджено на засіданні кафедри інформаційних систем.
Протокол №5 від 12.12.2008 р.

P78 Завдання до лабораторних робіт та методичні рекомендації до їх виконання з навчальної дисципліни "Основи системного аналізу об'єктів та процесів комп'ютеризації" (у середовищі Rational RequisitePro) для студентів напряму підготовки "Комп'ютерні науки" усіх форм навчання / Укл. І. О. Ушакова, Г. О. Плеханова. За редакцією д. е. н., проф. Пономаренка В. С. – Харків: Вид. ХНЕУ, 2009. – 92 с . (Укр. мов.)

Приведені завдання для лабораторних робіт з управління вимогами до програмних систем у середовищі Rational RequisitePro, розкриті основні теоретичні положення стосовно предмету завдання, описані методичні рекомендації та порядок виконання робіт. Для закріплення матеріалу включені контрольні запитання.

Методичні рекомендації розраховані на студентів напряму підготовки "Комп'ютерні науки" усіх форм навчання.

ВСТУП

При розробленні програмних систем головним завданням є визначення вимог до програмного забезпечення (ПЗ). Під програмною вимогою (Software Requirement) розуміється можливість, яку будь-хто чекає від даного ПЗ. У стандарті (IEEE Standard Glossary of Software Engineering Terminology, 1990) , розробленому Інститутом інженерів з електротехніки і електроніки вимога визначається таким чином:

1. Умова або можливість, необхідна користувачеві для вирішення завдань або досягнення мети.

2. Умова або можливість, яку повинна мати система або її компонент, які відповідають договору, стандарту, специфікації або іншому офіційному документу.

3. Документоване надання умови або можливості перелічені в попередніх пунктах.

Правильно визначені вимоги є гарантією того, що система задовольнятиме вимогам зацікавлених у її розробці осіб. А оскільки, звичайно, вимоги зв'язані одна з одною, то процес їх систематизації і подальшого виконання може забирати багато сил. Причому завжди існує вірогідність, що якісь з вимог користувачів будуть не реалізовані або неправильно реалізовані. Для великих систем кількість цих вимог може бути дуже значною і при цьому вимоги можуть змінюватися.

Для роботи з вимогами і документами, в яких вони відбиваються, відстеженням їх змін у методології RUP використовується програмний продукт Rational RequisitePro. Метою вивчення методології управління вимогами при розробці програмних систем з використанням IBM Rational RequisitePro є освоєння етапів планування проектів з розробки програмних систем, їх реалізації в IBM Rational RequisitePro, отримання навичок для роботи з IBM Rational RequisitePro.

Методичні рекомендації до виконання лабораторних робіт навчальної дисципліни "Основи системного аналізу об'єктів та процесів проектування" за напрямом підготовки "Комп'ютерні науки" дозволяють майбутнім фахівцям в області інформаційних систем і технологій отримати знання в сфері використання сучасних методів управління вимогами до програмних систем та навички роботи з інструментальним засобом Rational RequisitePro.

Лабораторна робота №1

Знайомство з RequisitePro

Цілі лабораторної роботи:

1. Вивчення роботи з вікном Let's Go RequisitePro і додатковими можливостями системи вбудованої допомоги Help.
2. Вивчення роботи із списком проектів RequisitePro у вікні Open Project.
3. Вивчення прикладу проекту Learning Project.

Методичні вказівки до виконання

Завдання 1.1. Використання вбудованої системи допомоги HELP, вікна Let's Go RequisitePro

У завданні 1.1 потрібно виконати наступне:

1. Відкрити RequisitePro.
2. Викликати вбудовану систему допомоги HELP.
3. Розглянути розділи довідки, зміст і індекс для пошуку, вікно Let's Go Rational RequisitePro.
4. Вивчити структуру вікна RequisitePro.

Вікно RequisitePro Let's Go RequisitePro (рис. 1) забезпечує простий доступ до інформації про програму і прийомів роботи з нею з використанням системи вбудованої допомоги.

У вікні Let's Go RequisitePro можна вибрати п'ять варіантів доступу до інформації про програму RequisitePro і прийомів роботи з нею.

При натисненні іконок з наступними найменуваннями з'являються:

1. RequisitePro Quick Tour – рекомендації щодо швидкого освоєння RequisitePro;
2. RequisitePro Tutorial – підручник з RequisitePro;
3. Requirements Management Tour – опис управління вимогами;
4. Project Administration Tips – рекомендації щодо адміністрування проекту;
5. Online Help – онлайн-довідка з Rational RequisitePro.



Рис. 1. Вікно Let's Go RequisitePro

З вікна Let's Go RequisitePro можна викликати:

1) розділ Release Notes and Help (Опис і допомога):

опис версії RequisitePro при натисненні кнопки RequisitePro Release Notes;

опис інтерфейсу користувача при натисненні кнопки Exploring RequisitePro;

довідку про Rational RequisitePro при натисненні кнопки RequisitePro Help;

систему вбудованої допомоги з розширення інтерфейсу RequisitePro, при натисненні кнопки RequisitePro Extensibility Interface Online Help;

інформацію з технічної підтримки RequisitePro, при натисненні кнопки Contacting Technical Support.

2) розділ Online Resources (онлайн ресурси):

технічну підтримку в мережі при натисненні кнопки Technical Support on the Web;

сайт розробників Rational при натисненні кнопки RequisitePro tools, technical notes, whitepapers, scripts, documentation, downloads, training, discussions: Rational Developer Network;

інформацію про RequisitePro на сайті фірми Rational Software Corporation, при натисненні кнопки RequisitePro Product page on the Web;

інформацію про продукти Rational, при натисненні кнопки Rational on the Web.

3) розділ Process (Процес):

методологію розробки програмного забезпечення Rational Unified Process, при натисненні кнопки Software Engineering Process – Rational Unified Process.

Система вбудованої допомоги HELP забезпечує підтримку отримання різної інформації, включаючи допомогу з розширення інтерфейсу RequisitePro, розширену довідку, пошук інформації.

Порядок виконання

1. Для запуску RequisitePro клацніть двічі по іконці із зображенням продукту на робочому столі або виберіть RequisitePro через меню "Пуск". На екрані з'являється вікно допомоги Let's Go Rational RequisitePro (див. рис. 1). Виберіть кнопку Close для продовження роботи. RequisitePro відкриє вікно Open Project (рис. 2). Закрийте вікно для продовження роботи, натиснув кнопку Cancel. На екрані відобразиться вікно RequisitePro з меню команд і панеллю інструментів у верхній частині (рис. 3).

2. Використовуючи меню команд RequisitePro, виберіть пункт меню Help. З'явиться меню розділів вбудованої довідкової системи HELP RequisitePro.

3. Виберіть розділ меню Contents and Index і прогляньте вкладки Help. Ця опція – типовий довідковий файл допомоги Windows з вкладками. Після закінчення перегляду закрийте це вікно.

4. Виберіть розділ меню Extensibility Interface Reference і прогляньте цей розділ Help. З'явиться інформація щодо інтерфейсу доступу до даних RequisitePro.

5. Виберіть команду Help знову і виберіть розділ меню Let's Go Rational RequisitePro. На екрані з'явиться вікно Let's Go Rational RequisitePro.

6. Натисніть на кожну з чотирьох іконок Help у вікні Let's Go Rational RequisitePro. З'являються різні розділи. Прогляньте ці розділи.

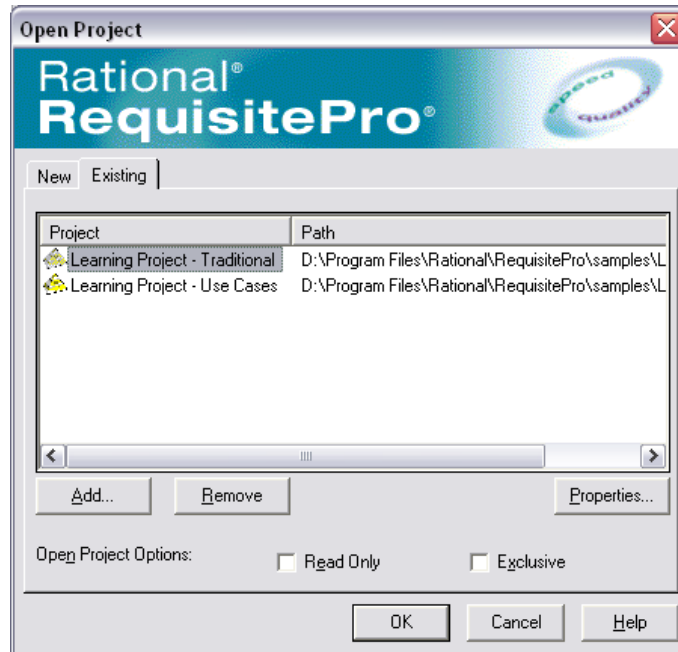


Рис. 2. Вікно Open Project

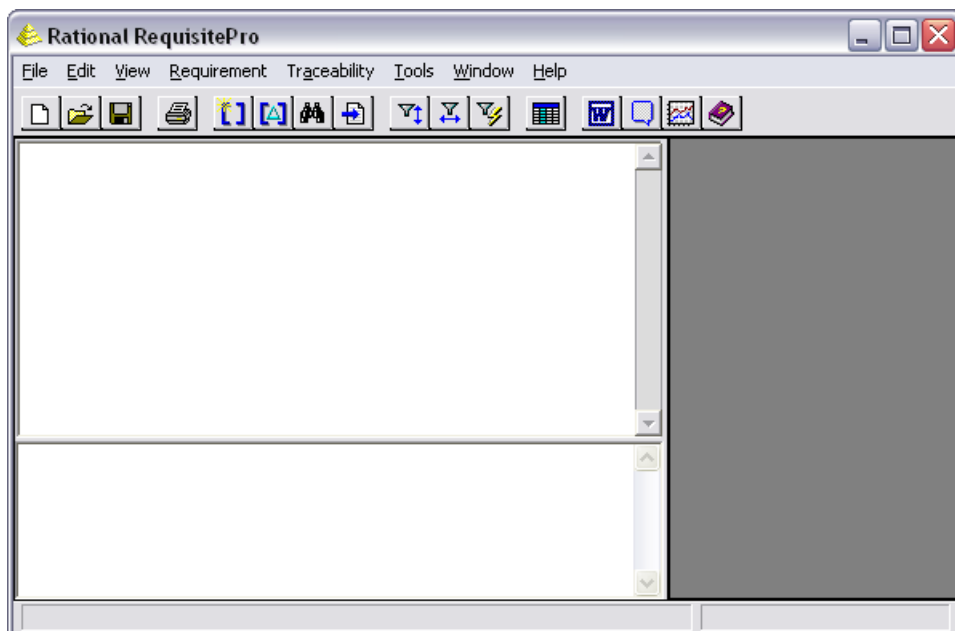


Рис. 3. Вікно RequisitePro

7. Натисніть на іконки в розділах Release Notes and Help, Online Resources, Process для вивчення додаткових можливостей. Кнопки, розміщені в нижній частині вікна Let's Go Rational RequisitePro

забезпечують доступ до Rational Unified Process, до статей, технічної підтримки і інших ресурсів мережі Інтернет.

8. Вивчіть інтерфейс користувача у вікні RequisitePro, використовуючи розділ Exploring RequisitePro вікна Let's Go RequisitePro.

9. Після закінчення роботи закрийте вікно Let's Go RequisitePro.

Завдання 1.2. Додавання і видалення проекту в списку проектів

У завданні 1.2 потрібно виконати наступне:

1. Переглянути список проектів RequisitePro у вікні Open Project.
2. Визначити розташування файлу RequisitePro у форматі *.rqs.
3. Видалити проект із списку проектів.
4. Додати проект до списку проектів.

Порядок виконання.

1. Виконайте команду File→Open Project. На екрані з'являється вікно із списком проектів (див. рис. 2).

2. Перегляньте список проектів. Переконайтеся, що в списку проектів існують навчальні проекти Learning Project. Learning Project – є типовим проектом і включає інформацію, яку можна використовувати при створенні власних проектів.

3. Виберіть із списку проект Learning - Traditional. Learning - Traditional - міні-проект, який можна використовувати при створенні звичайного проекту.

4. Клацніть по кнопці Remove. Проект Learning - Traditional видалиться із списку проектів. Проект більше не показаний у списку проектів. Проте, проект зберігається у файловій системі. Проект можна додати до списку проектів при натисненні кнопки "Add" і вибору директорії, у якій зберігається проект.

5. Натисніть кнопку "Add".

6. Проглянете вікно Add Project і знайдіть папку з прикладами проектів RequisitePro, в якій зберігається проект Learning - Traditional.rqs. Проект розміщений у директорії C:\program files\Rational\RequisitePro\samples\Learning_Project-Traditional\LEARNING-TRADITIONAL.rqs

7. Виберіть файл LEARNING - TRADITIONAL.rqs.

8. Натисніть кнопку Open для додавання Learning Project Traditional назад в список проектів.

9. Аналогічним чином додайте до списку проект Global Requirements.rqs. Проект розміщений у директорії C:\ProgramFiles\ Rational\RequisitePro \samples\Learning_Project-Use_Cases \ Global Requirements.rqs.

10. У списку проектів виберіть Learning Project - Traditional. Натисніть кнопку Properties. У вікні Database Properties, що з'явилося, відображаються основні властивості бази даних проекту, наприклад, тип використовуваної бази даних, директорії розташування файлів RequisitePro і тому подібне. Перегляньте властивості бази даних проекту (рис. 4).

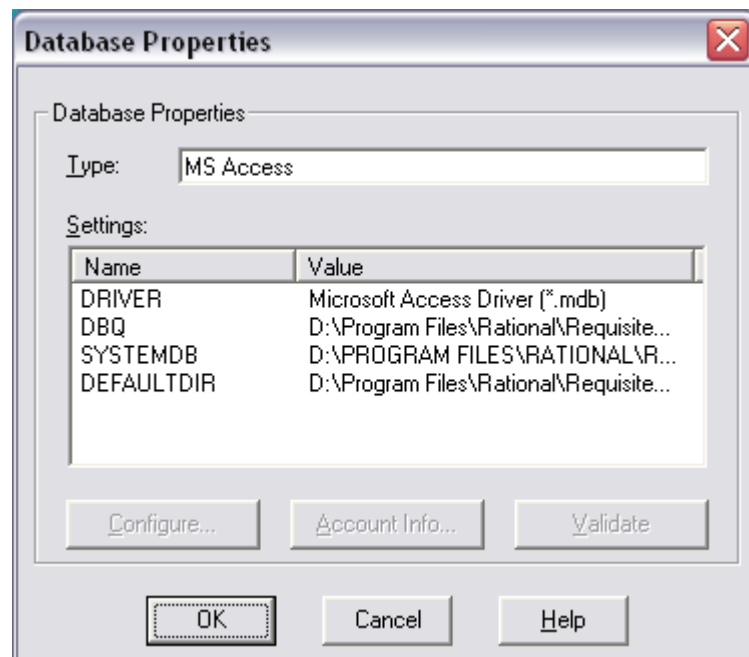


Рис. 4. Властивості бази даних проекту

11. Для повернення у вікно із списком проектів клацніть по кнопці Cancel, потім у вікні, що з'явилося, ще раз клацніть по кнопці Cancel. Вікно із списком проектів буде закрито

Завдання 1.3. Вивчення учбового проекту Learning Project

У завданні 1.3 потрібно виконати наступне:

1. Відкрити учбовий проект Learning Project.
2. Проглянути властивості типового проекту.
3. Відкрити і переглянути пов'язані з проектом документи.

Порядок виконання

1. У списку існуючих проектів виберіть проект Learning Project-Traditional. Натисніть кнопку ОК. Відкриється вікно Project Logon.

2. У випадному списку виберіть ім'я користувача. Натисніть кнопку ОК. Відкриється проект Learning Project із списком папок і документів.

3. Відкрити пов'язані з проектом документи можна по черзі, вибираючи папку і необхідний документ із зображенням іконки редактора Word. Перегляньте всі документи Word. Вибраний документ стає активним і відображається в Word, якщо його активізувати подвійним клацанням миші на іконці документа. Перегляньте зміст документа в Word.

4. У меню File виберіть пункт Close Project. Learning Project-Traditional буде закритий зі всіма активними вікнами.

5. У меню File виберіть пункт Open Project. У списку проектів, що з'явилися, виберіть проект Learning Project-Use Cases і натисніть кнопку ОК. Відкриється вікно Project Logon. У вікні відображається ім'я користувача. Знову натисніть кнопку ОК. Відкриється проект Learning Project-Use Cases із списком папок.

5. Перегляньте типовий проект. Для перегляду його властивостей в меню File виберіть Properties. Вивчіть документи і властивості типового проекту.

6. Закрийте проект Learning Project-Use Cases. Вікно проекту закриється.

7. Аналогічні дії виконаєте з проектом Global Requirements.rqs.

Зміст звіту з лабораторної роботи № 1

1. Цілі роботи.

2. Структура вікна Let's Go RequisitePro.

Навести екранну форму вікна Let's Go RequisitePro з коротким описом всіх її іконок і розділів.

3. Структура призначеного для користувача інтерфейсу вікна RequisitePro.

Навести екранну форму основного вікна RequisitePro для учбового проекту Use Cases з коротким описом всіх його елементів.

4. Висновки.

Контрольні питання

1. Яким чином можна запустити RequisitePro?
2. Для чого призначено вікно Let's Go RequisitePro? Яка його структура?
3. Опишіть елементи призначеного для користувача інтерфейсу вікна RequisitePro.
4. Які розділи входять до меню Help? Для чого вони призначені.
5. Охарактеризуйте вікно списку проектів. Які дії над проектами можна виконати в цьому вікні?
6. Яким чином можна відкрити проект?
7. Як проглянути властивості відкритого проекту?
8. Охарактеризуйте вкладки вікна Properties.
9. Що відбувається при активізації документа Word з вікна відкритого проекту?
10. Які існують можливості по роботі з вбудованою системою допомоги?

Лабораторна робота №2 Створення нового проекту

Цілі лабораторної роботи:

1. Створення нового проекту.
2. Вивчення параметрів конфігурації RequisitePro.

Методичні вказівки до виконання

Завдання 2.1. Створення проекту в RequisitePro

У завданні 2.1 потрібно виконати наступне:

1. Створити новий проект на основі порожнього шаблону Blank.
2. Створити новий проект на основі шаблону Composite Template, ознайомитися з його структурою.
3. Створити новий проект на основі шаблону Traditional Template, ознайомитися з його структурою.
4. Створити новий проект на основі шаблону Use-Case Template, ознайомитися з його структурою.

Порядок виконання

1. Запустіть RequisitePro. На екрані з'явиться вікно Create Project.
2. У поточному вікні виберіть закладку New (рис. 5).



Рис. 5. Вікно Create Project

3. Виберіть шаблон Blank, натисніть кнопку ОК. З'явиться вікно Rational RequisitePro Project Properties для налаштування параметрів проекту (рис.6).

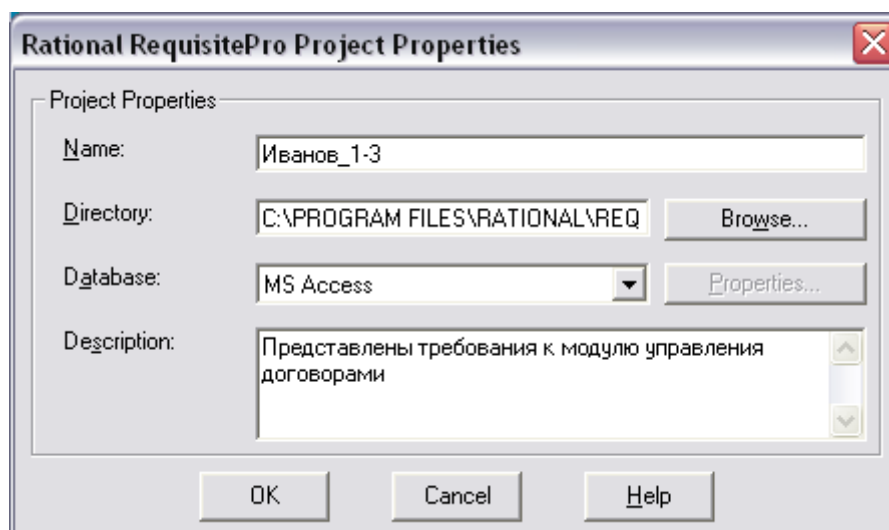


Рис. 6. Вікно властивостей проекту

4. У полі Name введіть назву проекту, наприклад, Іванов_1_3. В якості імені проекту вкажіть прізвище, курс і групу студента.

5. Натисніть кнопку Browse. Задайте каталог проекту. Потім натисніть кнопку OK. Використовуйте шлях – C:\ program files \ Rational \ RequisitePro \ Projects \ Назва_проекту\.

6. У полі Description наводиться будь-яке корисне пояснення змісту проекту, наприклад, "Подані вимоги до модуля управління договорами".

7. Для створення проекту натисніть кнопку **OK**. Проект буде створений і поміщений до списку проектів RequisitePro.

8. При появі повідомлення Project directory does not exist. Do you want to create it? (рис. 7) натисніть на кнопку **Да**. Це повідомлення з'являється, якщо каталог проектів у пункті 5 не був заданий.



Рис. 7. Вікно – запит на створення директорії

9. Після закінчення створення проекту з'являється вікно Create Rational RequisitePro Project (рис. 8) з повідомленням: The RequisitePro Project was successfully created. Натисніть кнопку Close. З'являється вікно проекту (рис. 9).

10. Закрийте проект, виконавши команду File→Close Project.

Увага! Проект, створений на основі порожнього шаблону не видаляти. З ним ви працюватимете в наступних лабораторних роботах.

11. Спробуйте створити проекти за допомогою інших шаблонів: Composite Template, Traditional Template, Use-Case Template. Прочитайте коментарі до кожного з шаблонів проекту. Ознайомтеся із структурою створених проектів.

12. Видаліть проекти створені на основі шаблонів Composite Template, Traditional Template, Use-Case Template.

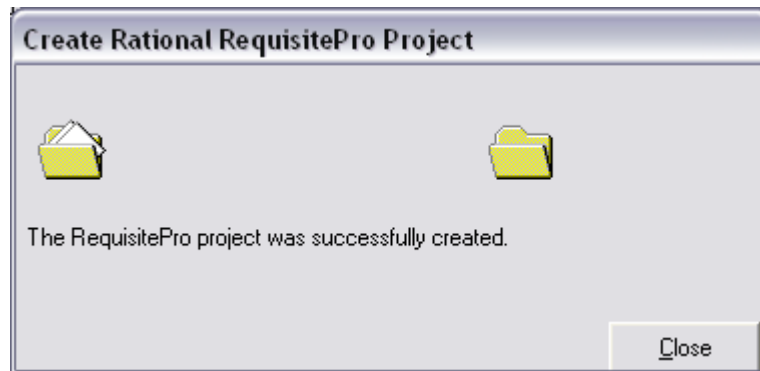


Рис. 8. Підтвердження створення проекту

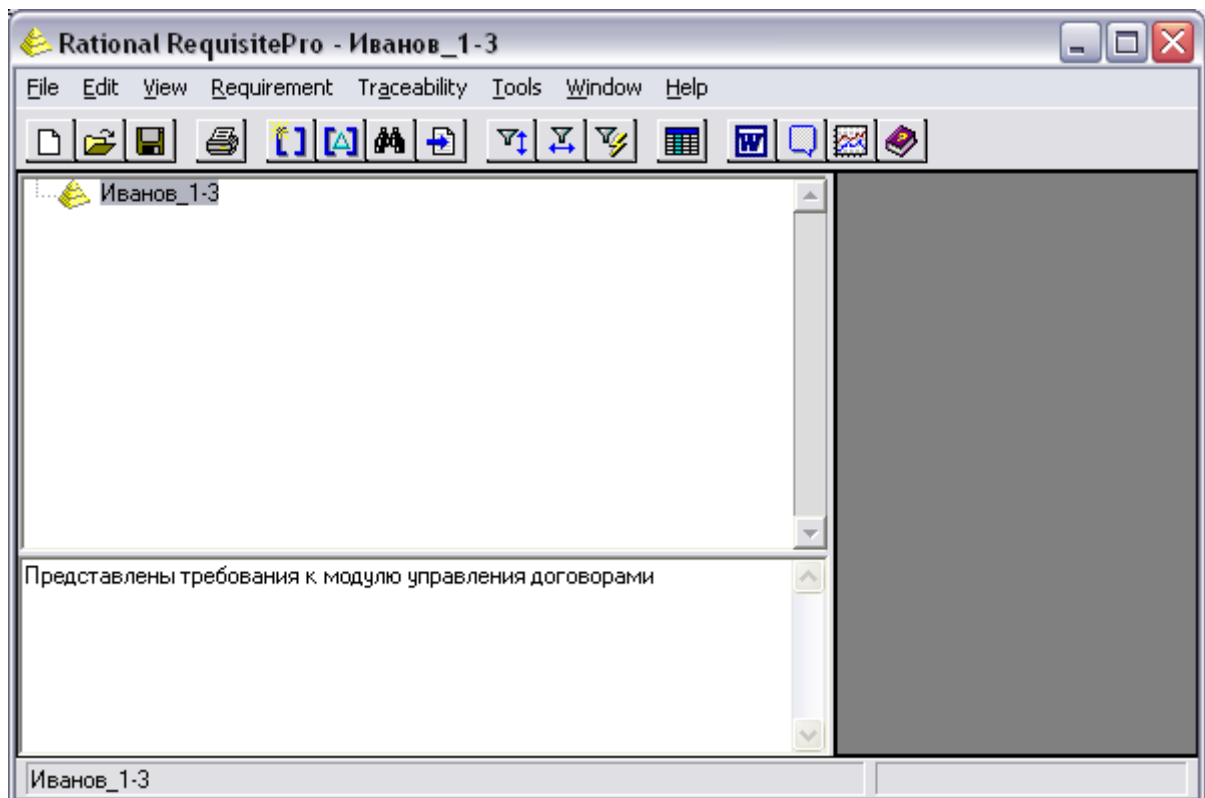


Рис. 9. Вікно RequisitePro з створеним проектом

Завдання 2.2. Установка параметрів конфігурації RequisitePro

У завданні 2.2 потрібно виконати наступне:

1. Ознайомитися з параметрами конфігурації RequisitePro.
2. Встановити параметри конфігурації для свого проекту.

У RequisitePro є можливість налаштовувати опції (параметри конфігурації) для роботи з проектами. Наприклад, можна задати параметри призначеного для користувача інтерфейсу, прописати шляхи

доступу до файлів проекту і тому подібне. Далі приведений опис опцій для шляхів доступу до файлів проекту.

Document Outlines (Шаблони документів) – це поле визначає вторинний шлях для доступу до шаблонів документів. Коли RequisitePro встановлюється у визначеному місці на ПК, то за умовчанням шлях доступу до шаблонів документів буде наступним:

<install drive> :\ Program Files \ Rational \ RequisitePro \ outlines.

Ця опція використовується для визначення додаткового розташування шаблонів документів (наприклад, директорія на сервері для спільних шаблонів документів).

Примітка: основний і вторинний шляхи до шаблонів документів зберігаються в машинній конфігурації, а не в конфігурації проекту RequisitePro.

Working Path (робочий шлях) – це поле вказує на поточний каталог проектів RequisitePro. Цей каталог використовується для запису туди нових проектів і тимчасових файлів RequisitePro.

Project Templates (шаблони проектів) – це поле визначає вторинний шлях для зберігання шаблонів проектів. Коли RequisitePro встановлюється у визначеному місці на ПК, то за умовчанням шлях доступу до шаблонів проектів буде наступним:

<install drive>:\ Program Files \Rational \ RequisitePro \templates.

Ця опція використовується для визначення додаткового розташування шаблонів проектів (наприклад, директорія на сервері для спільних шаблонів проектів).

Порядок виконання

1. На панелі інструментів RequisitePro виберіть в меню пункти Tools→Options. На екрані з'явиться вікно Options з параметрами конфігурації RequisitePro (рис. 10).

2. У вікні опцій є декілька груп параметрів конфігурації. Експериментуйте з параметрами конфігурації. Перемикаючи параметри, зверніть увагу на поведінку панелі інструментів. Скиньте всі параметри.

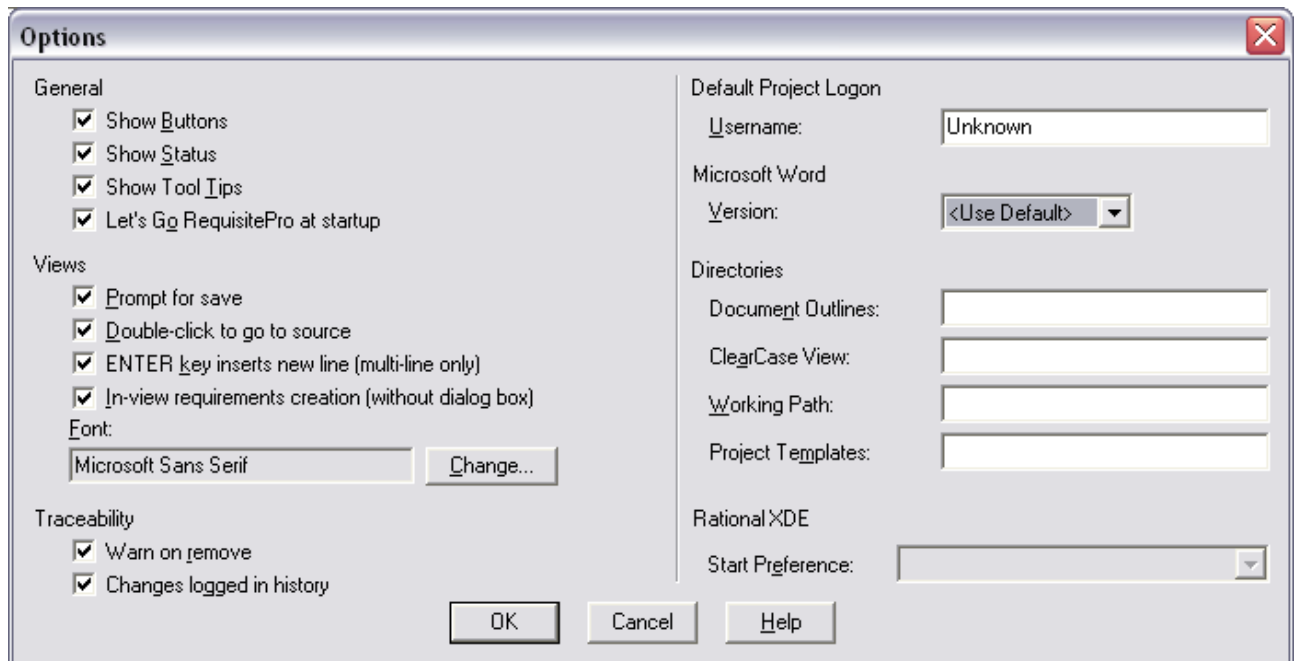


Рис. 10. Вікно опцій

3. Перегляньте інші доступні опції. Для отримання додаткової інформації про параметри конфігурації натисніть кнопку Help або F1. Після вивчення параметрів конфігурації закрийте вікно Options.

Зміст звіту з лабораторної роботи № 2

1. Мета роботи
2. Створення нового проекту.

Коротко описати порядок створення нового проекту. Навести роздруківку вікна з властивостями створеного проекту, основного вікна створеного проекту Exploring RequisitePro.

3. Вікно конфігурації RequisitePro.

Навести роздруківку вікна опцій із установленими параметрами конфігурації. Коротко описати призначення всіх груп опцій і параметрів конфігурації.

- 4 Висновки.

Контрольні питання

1. Яким чином створюють новий проект у RequisitePro?
2. Які типи шаблонів використовуються для створення проекту в RequisitePro? Охарактеризуйте кожен з цих шаблонів проекту?

3. У якій директорії створюються проекти за умовчанням?
4. Які групи опцій представлені у вікні Options у RequisitePro?
5. Охарактеризуйте кожну з опцій (параметри конфігурації) Requisite Pro.
6. Для чого призначені директорії: Document Outlines, ClearCase View, Working Path, Project Templates?

Лабораторна робота № 3

Створення типів вимог і атрибутів типів вимог

Цілі лабораторної роботи:

1. Створення типів вимог.
2. Створення атрибутів вимог і їх значень.

Методичні вказівки до виконання

Завдання 3.1. Створення типів вимог

У завданні 3.1 потрібно створити типи вимог.

Тип вимоги встановлює його опис, а також інформацію, яка використовуватиметься при створенні вимоги. Тип вимоги служить шаблоном для всіх вимог одного типу і використовується для класифікації або угруповання вимог у проекті. Кожний тип вимоги має унікальний набір атрибутів.

У документі План управління вимогами (додаток А, табл. 3.2) подано опис типів вимог, які будуть використані в проекті.

Порядок виконання

1. Використовуючи панель інструментів RequisitePro, відкрийте створений проект. Виберіть проект з поточного списку проектів.
2. Використовуючи контекстне меню проекту виберіть пункт Properties. На екрані з'явиться вікно з вкладками для завдання різних атрибутів проекту
3. Виберіть вкладку Requirement Types. На екрані з'явиться вкладка типів вимог – Requirement Types (рис. 11).

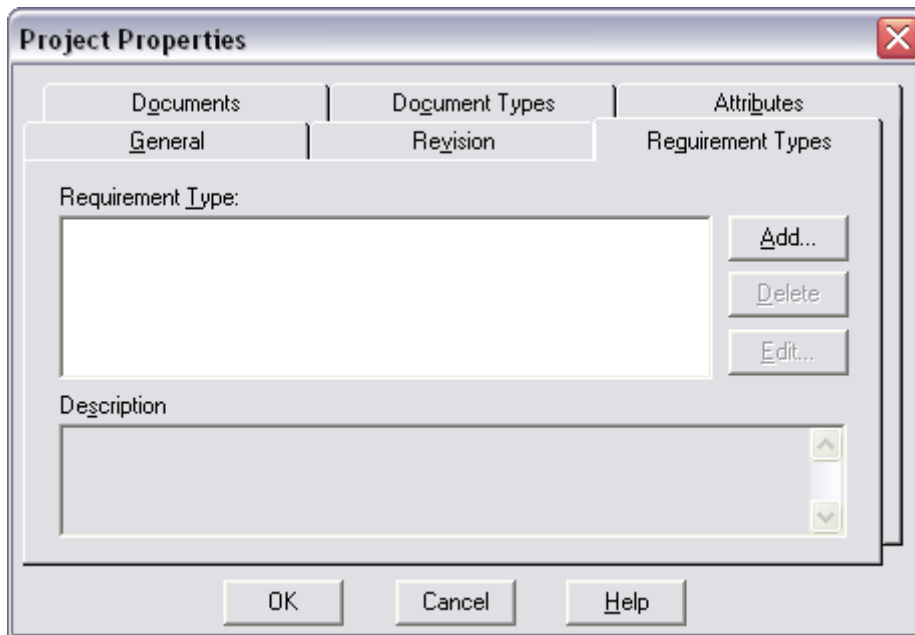


Рис. 11. Вкладка Requirement Types

4. Натисніть кнопку Add. Відкриється вікно Requirement Type для створення нової вимоги (рис. 12).

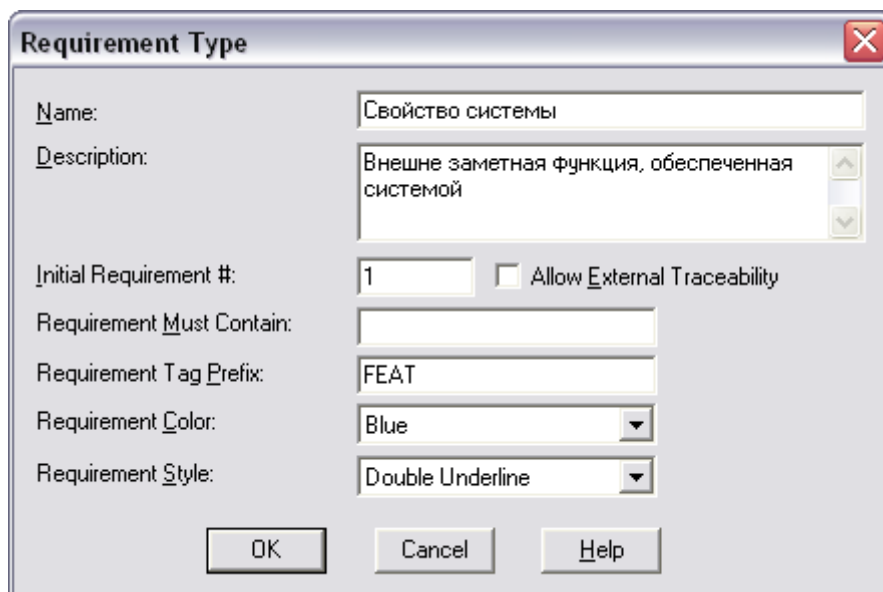


Рис. 12. Вікно Requirement Type

5. У полі Name задається ім'я новому типу вимоги. Введіть ім'я типу вимоги, наприклад, властивість системи.

6. У полі Description задається опис вимоги. Введіть опис нового типу вимоги, наприклад, зовні помітна функція, забезпечена системою.

7. У полі Initial Requirement # задайте номер першої вимоги для даного типу вимоги. Первинний номер вимоги за умовчанням буде # 1.

8. Упевніться, що прапор Allow External Traceability не включений. Це поле відноситься до властивості відстеження зв'язків проектів у RequisitePro.

9. Залиште порожнім поле Requirement Must Contain. У полі вказується слово або фраза, яка має бути присутньою в кожній вимозі цього типу. Залишаючи порожнім це поле, ви не задаєте жодних специфічних обмежень на формулювання вимог цього типу.

10. У полі Requirement Tag Prefix введіть короткий текст, для позначення вимоги з таким типом, наприклад, FEAT для типу вимоги Властивість системи (див. додаток А, табл. 3.2). Значення цього поля уміщено надалі в полі Tag (мітка), яке передує полю з описом вимоги цього типу.

11. Прийміть за умовчанням значення для обох полів Requirement Color, Requirement Style або задайте їх на свій розсуд.

12. Натисніть кнопку ОК. На екрані з'явиться вкладка Requirement Types. Створений тип вимоги відобразиться у вікні Requirement Type.

13. Аналогічним чином додайте інші типи вимог відповідно до додатку А (табл. 3.2). Вводиться відповідна інформація для кожного з наведених типів вимог.

Завдання 3.2. Створення атрибутів вимог і їх значень

У завданні 3.2 потрібно виконати наступне:

1. Створити атрибути і значення атрибутів вимог.
2. Додати атрибути до вибраного типу вимоги.

Перед створенням вимог потрібно визначити їх атрибути і значення атрибутів.

Атрибут вимоги – це поле, що містить описову інформацію, пов'язану з вимогою.

Атрибут вимоги в Rational RequisitePro є або системним (визначається RequisitePro), або призначеним для користувача (визначається проектувальником).

Наприклад, до системних атрибутів відносяться:

номер версії (Revision number);

автор (Author);
дата (Date);
час (Time) і так далі.

Прикладами призначених для користувача атрибутів можуть бути:

пріоритет (Priority);
трудність (Difficulty) і так далі.

Значення атрибуту вимоги представляє інформацію, призначену для атрибуту вимоги. Значення атрибуту можуть бути текстові або числові. Наприклад, атрибут Пріоритет може мати значення:

обов'язкове;
рекомендоване;
опційне.

У документі План управління вимогами (дод., табл. 3.3-3.4) поданий опис атрибутів типів вимог і значень цих атрибутів типів вимог, які будуть використані в проекті. Атрибути типів вимог використовуються в RequisitePro для управління вимогами.

Порядок виконання

1. Використовуючи панель інструментів RequisitePro, відкрийте створений проект, вибравши проект з поточного списку проектів.

2. Використовуючи контекстне меню проекту, виберіть пункт Properties. На екрані з'явиться вікно з вкладками для завдання різних атрибутів проекту.

3. Виберіть вкладку Attributes. На екрані з'явиться вікно для завдання атрибутів вимог.

4. Виберіть перший тип вимоги, з якою працюватимете, із списку типів вимог, наприклад FEAT: Властивість системи. Натисніть стрілку "вниз" у випадному списку вимог (рис. 13).

5. Натисніть кнопку Add, для додання атрибуту. На екрані з'явиться вікно Add Attribute для додання нового атрибуту до вибраного типу вимог (рис.14).

6. Введіть ім'я нового атрибуту в полі Label, наприклад, Тип. Ім'я атрибуту береться відповідно до плану управління вимогами (див. табл. 3.3 дод.). Імена мають бути унікальні лише усередині даного типу вимоги. Таким чином, можна використовувати одне і те ж ім'я атрибуту в декількох різних типах вимог в одному проекті.

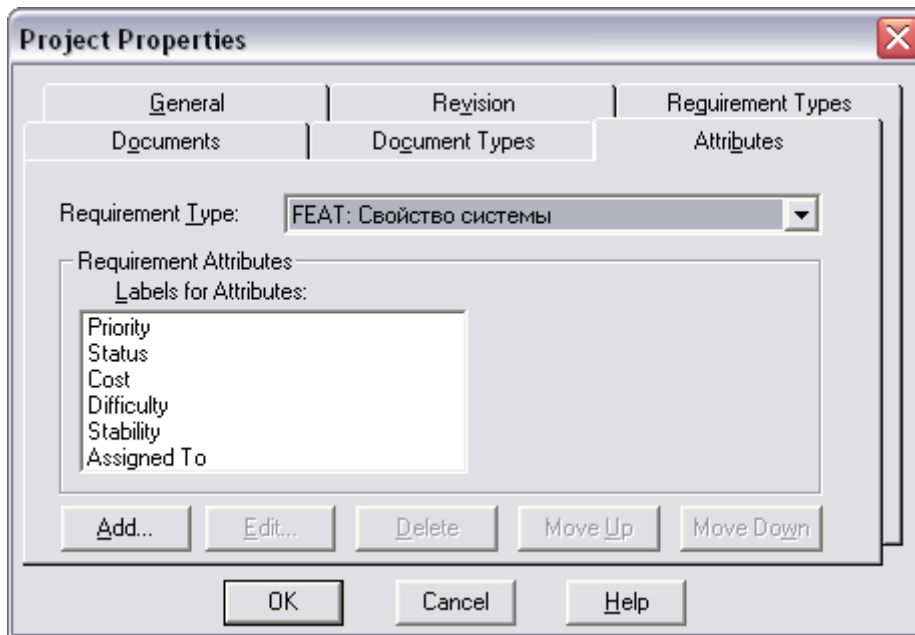


Рис. 13. Вкладка Attributes

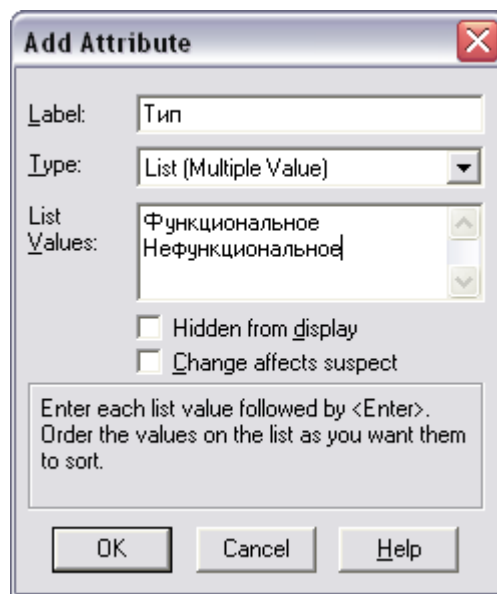


Рис. 14. Вікно Add Attributes

7. Визначте тип представлення значень атрибутів (Type) з випадного списку, вибравши тип List (Multiple Value). Вибраному типу відповідає список різних значень для введення або значення за умовчанням. Введіть у поле List Values значення атрибуту: Функціональне, Нефункціональне (див. рис. 14). Введення кожного значення повинне підтверджуватися натисненням клавіші ENTER.

8. Включіть прапор Change affects suspect. Цей прапор встановлюється для відстеження зв'язків між вимогами.

9. Натисніть кнопку ОК, для додання нового атрибуту до вибраного типу вимоги.

10. Натисніть на створений атрибут, щоб відобразилися його поточні значення. Значення атрибутів відображається у вікні Values per Attribute (рис. 15).

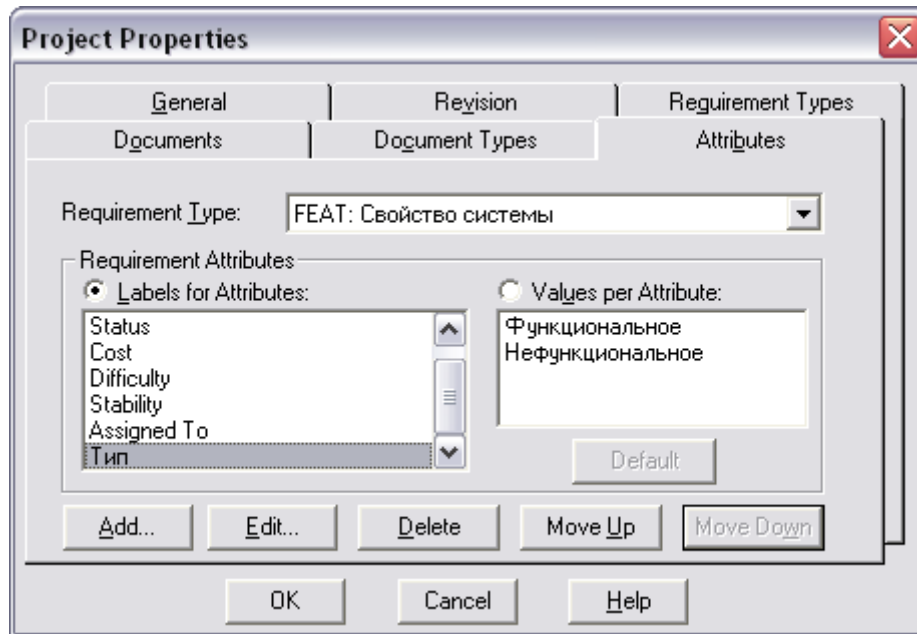


Рис. 15. Значення атрибутів типу вимоги

11. Виберіть значення атрибуту, яке присвоюється йому за умовчанням. Виділіть це значення в списку у вікні Attribute Value і натисніть кнопку Default. Після цього в дужках поряд із значенням атрибуту з'явиться відмітка (Default) (рис. 16). Значення за умовчанням встановлюється автоматично усякий раз, коли буде створюватися нова вимога.

12. Натисніть кнопку ОК. Значення атрибуту буде додано до бази даних проекту. Ті ж самі значення атрибутів можуть використовуватися в інших атрибутах.

Значення атрибуту може бути видалене. Для видалення значень атрибуту виберіть значення атрибуту, натисніть кнопку Delete. Вибране значення атрибуту буде видалено.

13. Аналогічно для кожного типу вимоги додати атрибути і їх значення. Атрибути і їх значення додаються відповідно до плану

управління вимогами (див. дод., табл. 3.3 – 3.4). Атрибути і значення атрибутів також можуть бути видалені.

Увага! Будьте обережні при їх видаленні, оскільки при видаленні можна знищити пов'язану з ними інформацію проекту.

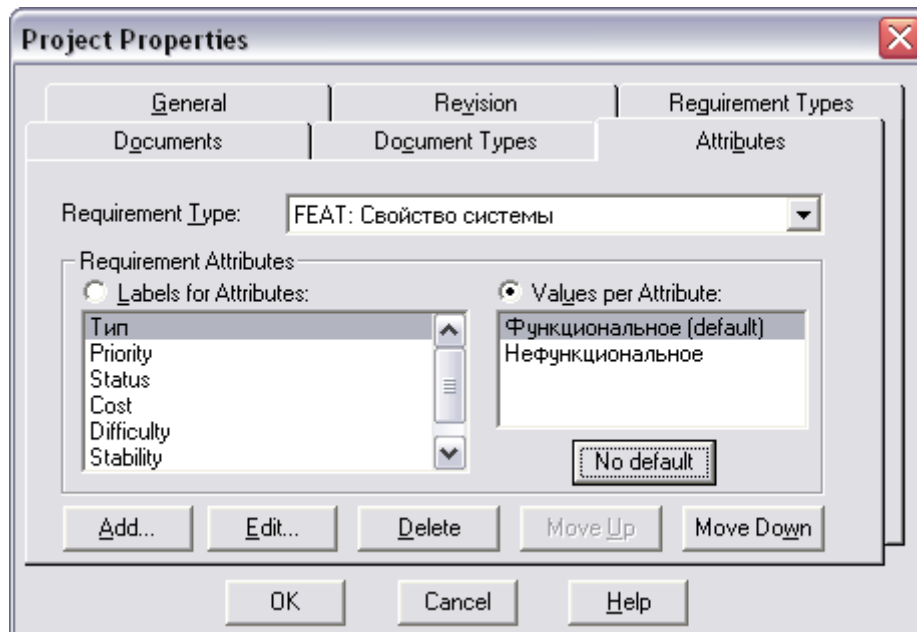


Рис.16. Вибір значення атрибуту за умовчанням

14. Видаліть не потрібні атрибути типу вимоги. Для видалення атрибуту виберіть атрибут типу вимоги, натисніть кнопку Delete. Вибраний атрибут буде видалений.

15. Натисніть кнопку ОК у діалоговому вікні Project Properties для завершення роботи з властивостями проекту.

Зміст звіту з лабораторної роботи № 3

1. Мета роботи.

2. Створення типів вимог.

Навести вікно з випадним списком типів вимог і таблицю з описом типів вимог.

3. Створення атрибутів вимог і їх значень.

Навести вікна із списками атрибутів для кожного типу вимоги і таблиці з описом атрибутів і їх значень.

4. Висновки.

Контрольні питання

1. Що таке тип вимоги?
2. У якому вікні створюються типи вимог? Як його активізувати?
3. Перерахуйте типи вимог.
4. Для кожного типу вимоги наведіть його атрибути.
5. Наведіть приклади значень атрибутів.
6. У якому документі визначаються типи вимог, типи і значення атрибутів типів вимог?
7. Які поля потрібно заповнити при створенні нового типу вимоги? Поясніть їх значення.
8. У якому випадку потрібно активізувати прапор Allow External Traceability?
9. У яких випадках заповнюється поле Requirement Must Contain?
10. У якому вікні створюються атрибути типів вимог? Як його активізувати?
11. Де надалі використовується значення поля Requirement Tag Prefix?
12. Які поля потрібно заповнити при створенні нового атрибуту типу вимоги? Поясніть їх значення.
13. Які типи представлення значень атрибутів можна використовувати для атрибутів?
14. Наведіть приклади атрибутів для конкретного типу вимоги.
15. У яких випадках в атрибутах типів вимог використовується перемикач Default? Наведіть приклад.

Лабораторна робота № 4 Створення шаблонів документів і типів документів в RequisitePro

Цілі лабораторної роботи:

1. Створення шаблонів документів.
2. Визначення типів документів, зв'язування типів документів з шаблонами документів.

Методичні вказівки до виконання

Завдання 4.1. Створення шаблонів документа

У завданні 4.1 потрібно виконати наступне:

1. Створити шаблони документів *План управління вимогами, Бачення, Глосарій, Специфікація варіантів використання і Додаткові специфікації*.
2. Створити файли описів шаблонів документів.
3. Розмістити файли з шаблонами і описами шаблонів у каталозі для їх зберігання.

У RequisitePro вимоги проекту можна як зберігати в БД, так і включати в проектні документи. Відображення вимог у проектних документах зручно для подальшого управління, як вимогами, так і документами в проекті.

Document (документ) – це будь-який документ Microsoft Word. Документ може бути частиною проекту Rational RequisitePro або не входити до проекту, може містити або не містити вимоги.

Requirements document (документ вимог) – це документ, створений у Microsoft Word або Rational RequisitePro, який включає вимоги і використовується для зв'язку з вимогами до продукту, що розробляється. Кожен документ вимог звертається до типів вимог, таких як наприклад властивості системи, варіанти використання і додаткові специфікації.

Документ вимог відрізняється від документа Word тим, що з нього можна звернутися до атрибутів вимоги і іншої інформації безпосередньо.

Outline (Шаблон документа) – це довідковий документ або шаблон документа, який використовується для створення нових документів у Rational RequisitePro. Outline може включати формати, інформацію про параметри сторінки, шрифти, стилі Word. Outline використовується для підтримки складання документів одного і того ж типу.

Інформація про Outline міститься в .def файлі. Вона включає:
the outline's full name (повне ім'я шаблону документа);
a description of the outline (опис шаблону документа);

the associated Word template's DOS filename (.dot) (DOS-сумісне ім'я файлу шаблону Word (.dot)).

Порядок виконання

1. Створіть в Word шаблон *Плану управління вимогами* відповідно до дод. Збережіть його у файлі *План управління вимогами.dot*.

2. Створіть в якому-небудь текстовому редакторі, наприклад в редакторі Блокнот, файл опису шаблону з розширенням .def з таким же ім'ям як ім'я шаблону .dot, наприклад *План управління вимогами.def* і збережіть його в своїй папці.

У редакторі файл опису шаблону повинен включати 3 рядки, відокремлених один від одного символом переведення рядка.

У першому рядку вказується повне ім'я шаблону документа, в другому – опис шаблону, в третьому – DOS-сумісне ім'я файла шаблону Word (рис. 17).

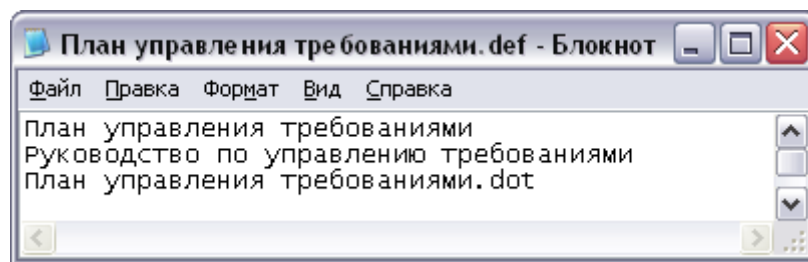


Рис. 17. Опис шаблону документа в редакторі Блокнот

3. Скопіюйте файли з шаблонами документів:

Імя_файла_шаблону.dot

Імя_файла_опису_шаблону.def

у каталог, який ви задали в опціях для зберігання шаблонів документів, наприклад: S:\Иванов_1_3\RequisitePro\OUTLINES.

У цій директорії знаходяться шаблони документів вашого проекту.

4. Аналогічно створіть шаблони документів *Бачення, Глосарій, Специфікація варіантів використання, Специфікація додаткових вимог, використовуючи їх шаблони.*

Завдання 4.2. Створення типів документів в RequisitePro

У завданні 4.2 потрібно виконати наступне:

1. Визначити типи документів, які використовуватимуться як шаблони для створюваних надалі фактичних документів.

2. Зіставити тип документа з шаблоном документа.
3. Приєднати заданий за умовчанням тип вимоги до знов створеного типу документа.
4. Ознайомитися з властивостями контролю і фіксації змін в проекті.

Перед створенням документів у RequisitePro, спочатку слід створити тип документа і зіставити його з існуючим шаблоном документа. Якщо проект створювався на основі існуючого проекту, він вже містить багато різних типів документів.

Тип документа – визначає опис і використання типу документа. Тип документа служить шаблоном для кожного створюваного документа вимог. Він також визначає розширення файла документа, як наприклад .gmr. Всі документи одного і того ж типу мають однакове розширення файла. Кожен тип документа має тип вимоги, яка використовується в документі за умовчанням (default). Кожна нова вимога в документі буде створена на основі типу вимоги за умовчанням, якщо не буде явно вказаний інший тип. Тип документа також ідентифікує шаблон (outline) для документа, який визначає структуру документа, його зміст, управляє форматуванням сторінок, тексту, стилів і так далі.

Типи документів, які використовуватимуться в проекті описані в документі *План управління вимогами* (див. табл. 3.1 в дод.).

Порядок виконання

1. Відкрийте свій проект в RequisitePro.
2. На панелі інструментів виберіть пункти меню File→Project Administration→Properties. Відкриється вікно Project Properties з вкладками для внесення змін до різних частин проекту.
3. Виберіть вкладку Document Types (рис. 18).
4. Натисніть кнопку Add. На екрані з'явиться вікно Document Type (рис.19).
5. У полі Name задається ім'я новому типу документа. Введіть – План управління вимогами.
6. У полі Description введіть опис нового типу документа, наприклад, – "Цей тип документа описує типи вимог, типи документів, атрибути вимог, а також стратегії відстеження зміни вимог проекту".

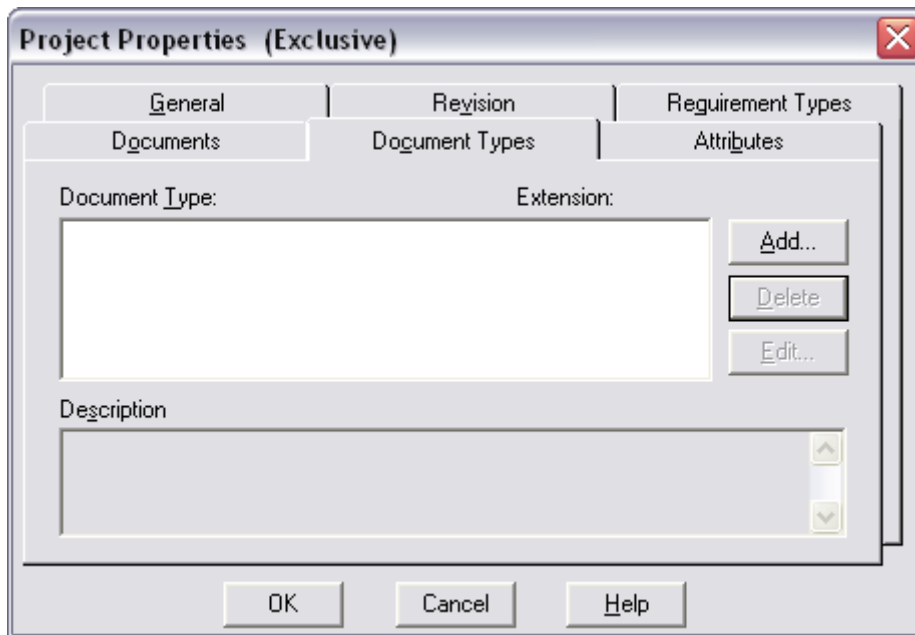


Рис. 18. Вкладка Document Type

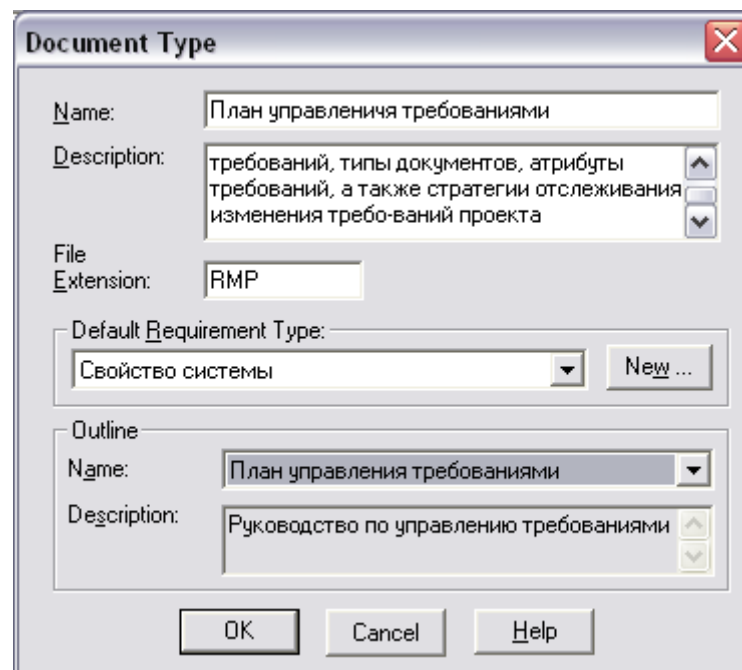


Рис. 19. Вікно Document Type

7. У поле File Extension введіть: DOS – сумісне розширення файлу англійською мовою, наприклад RMP. Не слід вводити розширення, яке вже використовується в проєкті.

8. У полі Default Requirement Type натисніть на "стрілку вниз", щоб відобразити список доступних типів вимог і виберіть тип вимоги:

Властивість системи. Визначається тип вимоги як значення за умовчанням, яке відповідатиме цьому типу документа.

9. У полі Outline Name, натисніть на "стрілку вниз", щоб відобразити список доступних шаблонів документів і виберіть шаблон *План управління вимогою*. Відбувається приєднання стандартного шаблону до нового певного типу документа. Це означає, що всі документи цього типу матимуть структуру і стиль цього шаблону.

10. Натисніть кнопку ОК, щоб закінчити опис нового типу документа. На вкладці Document Type з'являється створений тип документа і відповідне цьому типу розширення (рис. 20).

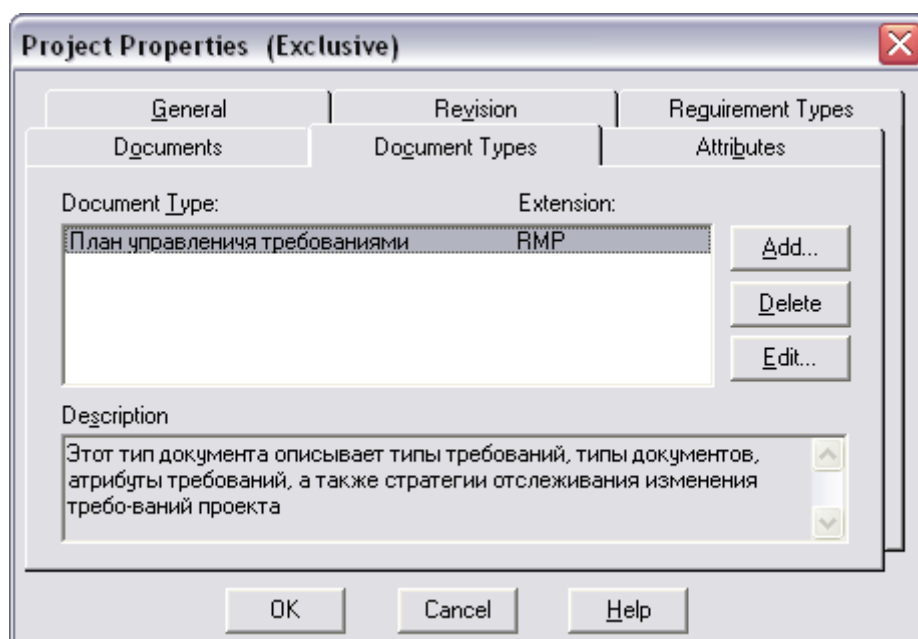


Рис. 20. Вкладка "Document Type" із створеним типом документа

11. Аналогічно додайте інші типи документів, керуючись табл. 3.1 дод.

13. У RequisitePro передбачена можливість контролю фіксації і зміни записів у проекті, як у конкретному документі, так і в кожній окремій вимозі проекту.

Знаходячись в діалоговому вікні Project Properties, виберіть вкладку Revision (рис. 21). Зверніть увагу на поле Change Description: у ньому відбивається історія змін внесених до проекту. Натисніть кнопку History для перегляду попередніх змін у проекті (рис. 22).

14. Натисніть кнопку ОК в діалоговому вікні Project Properties. Діалогове вікно Project Properties буде закрито.

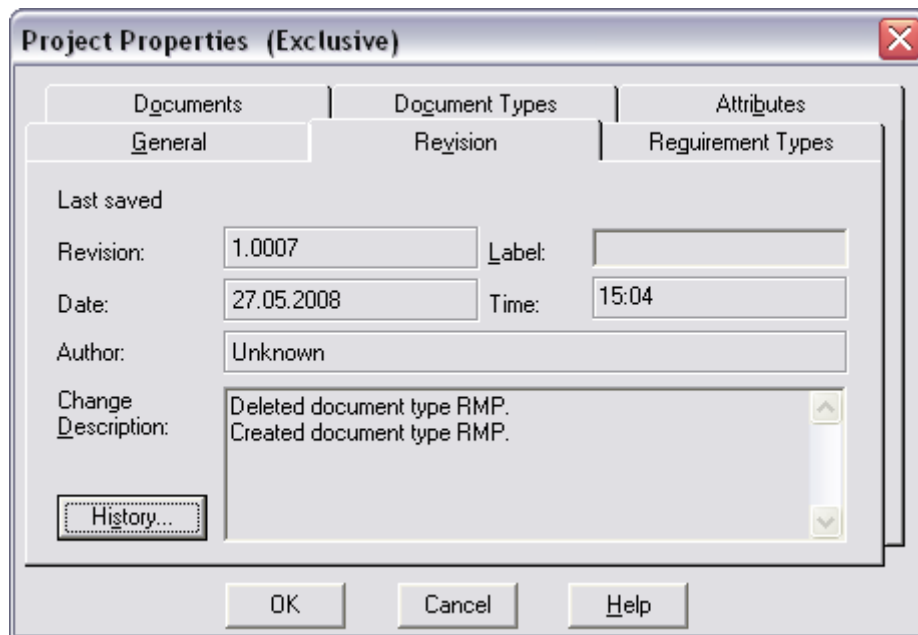


Рис. 21. Вкладка Revision

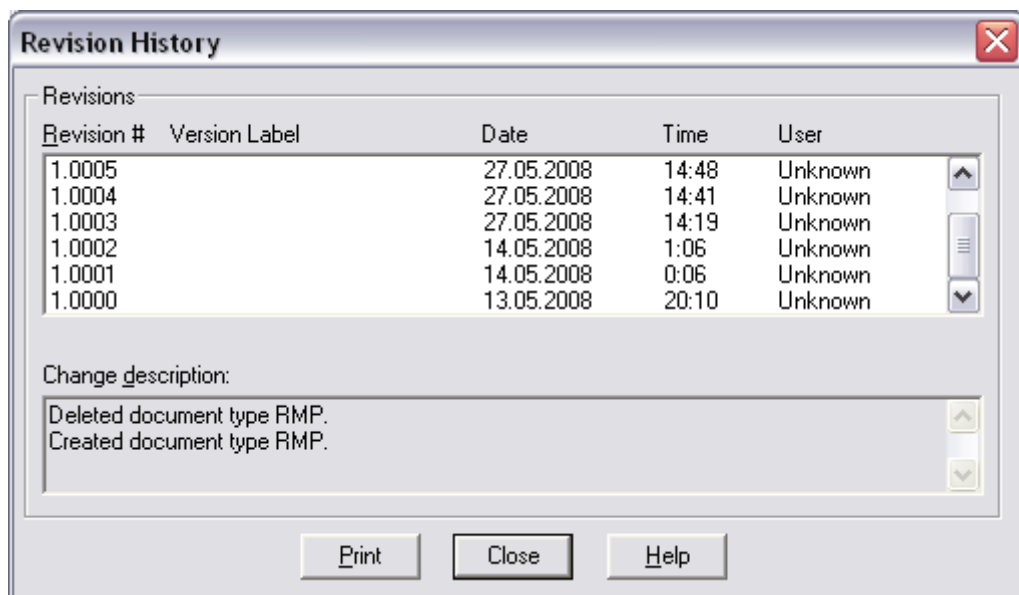


Рис. 22. Вікно Revision History

Зміст звіту з лабораторної роботи № 4

1. Мета роботи.
2. Створення шаблонів документів.

Навести вікна, створені в текстовому редакторі, з описом шаблонів документів: *План управління вимогами, Бачення, Глосарій, Специфікація варіантів використання і Специфікація додаткових вимог.*

Привести роздруківку вікна папки Outlines із створеними шаблонами документів.

3. Створення типів документів.

Привести роздруківки вікон для кожного створеного типу документа.

4. Версії та історія змін.

Привести роздруківки вікна Revision і Revision History.

5. Висновки.

Контрольні питання

1. Що в Rational RequisitePro розуміється під документом, документів вимог?

2. Чим відрізняється документ вимог від звичайного документа?

3. Що таке Outline в Rational RequisitePro?

4. Що входить до шаблону документа? Для чого він використовується?

5. Де розташована інформація про шаблон документа і що вона включає?

6. Що таке тип документа?

7. Для чого призначений тип документа?

8. Назвіть і охарактеризуйте типи документів, які створювалися в роботі.

9. Яким чином створюється тип документа?

10. Які поля заповнюються при створенні типу документа? Охарактеризуйте ці поля.

7. Яким чином можна переглянути версії проекту і історію їх змін?

Лабораторна робота № 5

Створення документів: *План управління вимогами, Глосарій у RequisitePro*

Цілі лабораторної роботи:

1. Створення документа План управління вимогами імпортуванням інформації із зовнішнього джерела в документ RequisitePro.
2. Створення документа *Глосарій* на основі шаблону документа.
3. Створення вимог типу термін глосарію в документі *Глосарій*.
4. Створення представлення Матриця вимог з атрибутами для термінів глосарію.
5. Редагування вимог у документі і БД.

Методичні вказівки до виконання

Завдання 5.1. Створення документа *План управління вимогами в RequisitePro*

У завданні 5.1 потрібно виконати наступне:

1. Імпортувати документ *План управління вимогами із зовнішнього джерела*.
2. Заповнити документ *План управління вимогами* інформацією про проект.

План управління вимогами описує документацію вимог, типів вимог і відповідні їм атрибути вимог, визначає інформацію і механізми управління, які використовуватимуть для вимірювання, складання звітів і управління зміною вимог до системи. З цього документа власне починається робота з управління вимогами. Цей документ розробляється на стадії *Початок* і модифікується при досягненні кожної головної віхи. Структура документа *План управління вимогами* подано в дод. А.

Порядок виконання

1. Відкрийте свій проект у RequisitePro.

2. На панелі інструментів виберіть пункти меню File→Import. Відкривається вікно Import Wizard, у якому потрібно вказати джерело імпорту (рис. 22).

3. Виберіть як джерело – Microsoft Document Word і вкажіть в полі Name of document import шлях до документа *План управління вимогами*, який буде імпортований з формату WORD у RequisitePro.

4. Натисніть кнопку Next. З'явиться вікно *Майстра імпорту*, в якому виберіть перемикач Document only (рис. 23).

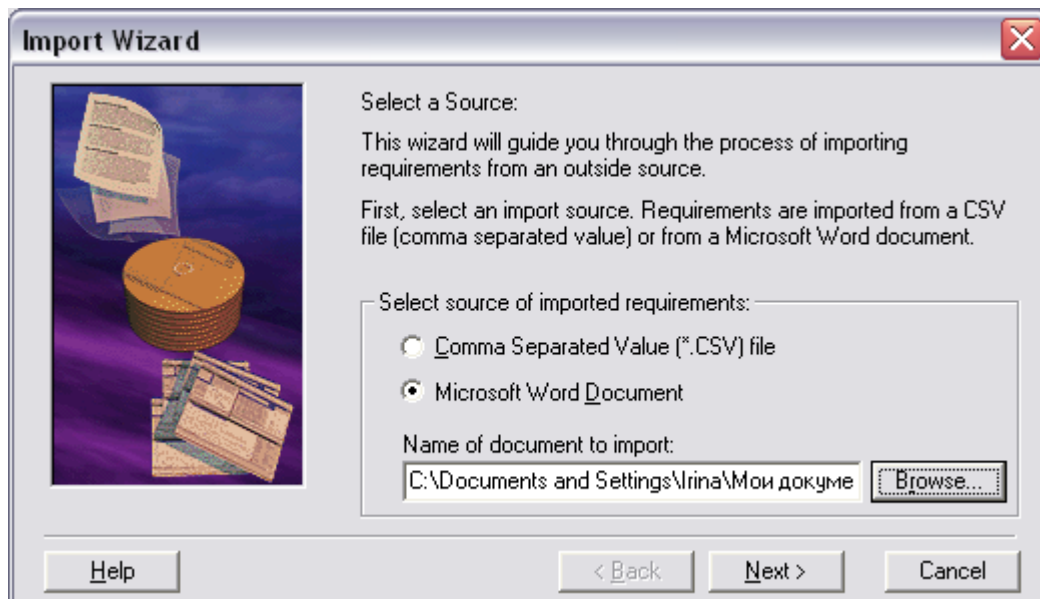


Рис. 22. Вікно Майстра імпорту Select a Source

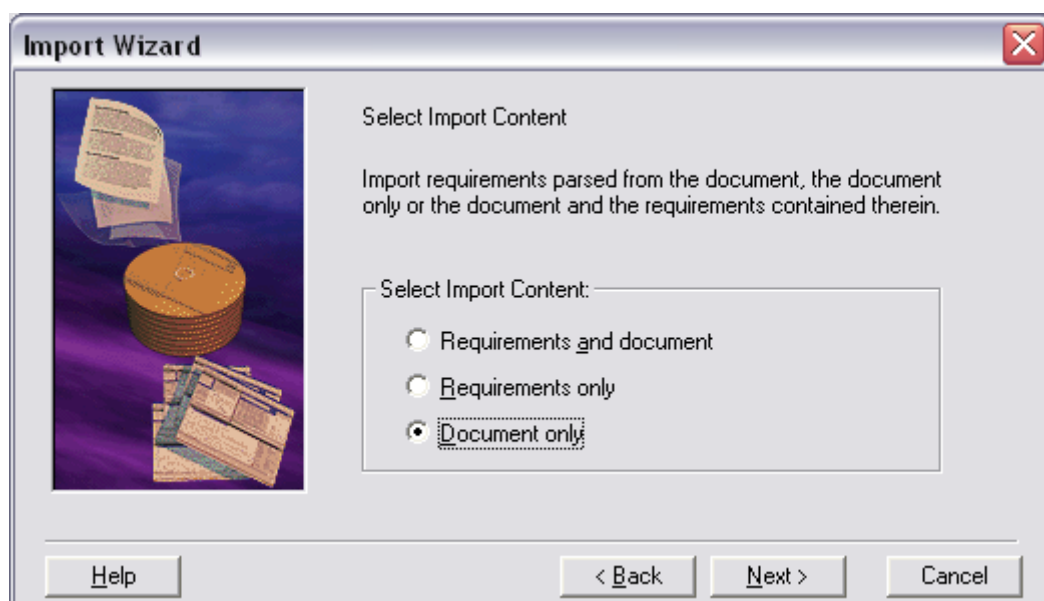


Рис. 23. Вікно Майстра імпорту Select Import Content

5. Натисніть кнопку Next. З'явиться вікно "Document Properties", у якому потрібно заповнити наступні поля для опису властивостей документа (рис. 24):

у полі Name введіть ім'я документа – *План управління вимогами*;

у полі Description введіть опис призначення документа;

у полі File Name введіть ім'я файлу, в якому зберігатиметься документ у RequisitePro. Ім'я файлу має бути унікальне. Якщо введене ім'я файлу вже існує, з'явиться діалогове вікно, що повідомить Вам про це. RequisitePro автоматично створює ім'я документа, додаючи розширення, визначене для цього типу документа;

включіть прапорець Show Tags, що забезпечить надалі відображення мітки вимоги в документі;

задайте директорію для зберігання документа в полі Directory. Натисніть кнопку Browse для завдання директорії;

всі документи, створені в проекті повинні мати певний тип документа. У полі Document Type натисніть "Стрілку вниз", і виберіть тип документа *План управління вимогами*;

натисніть кнопку OK.

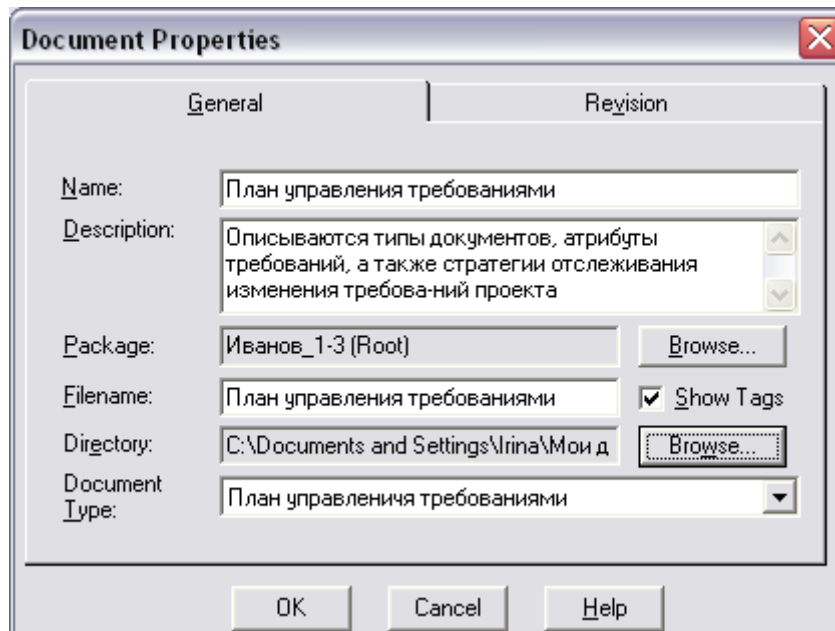


Рис. 24. Вікно Document Properties

6. З'явиться вікно, в якому потрібно вказати, який тип форматування буде використаний (рис. 25). Натисніть кнопку Так для

імпорту як текс-ту документа, так і його формату. Натисніть кнопку Ні, якщо потрібно використовувати форматування, яке використовується в шаблоні даного типу документа.

7. З'явиться вікно *Майстра імпорту Status*, у якому потрібно натиснути кнопку Commit для виконання імпорту документа (рис. 26).

8. Після завершення імпорту документ відкриється в MS Word.

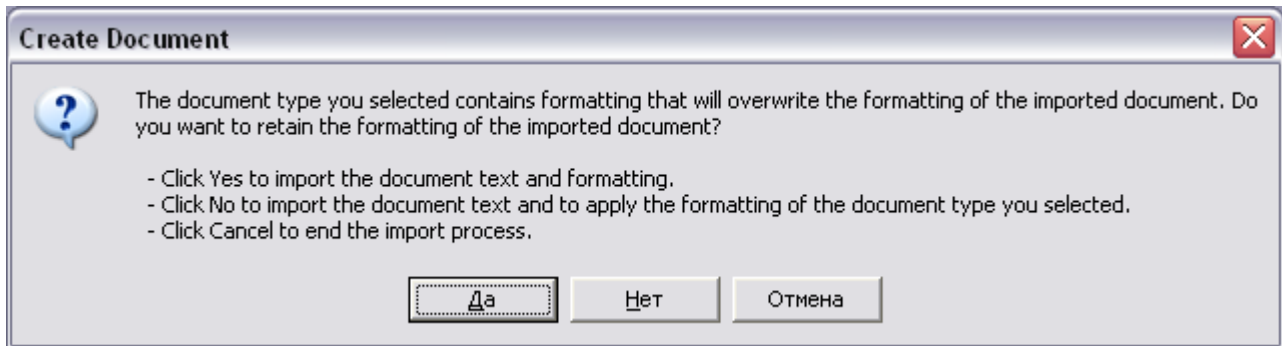


Рис. 25. Вікно Create Document

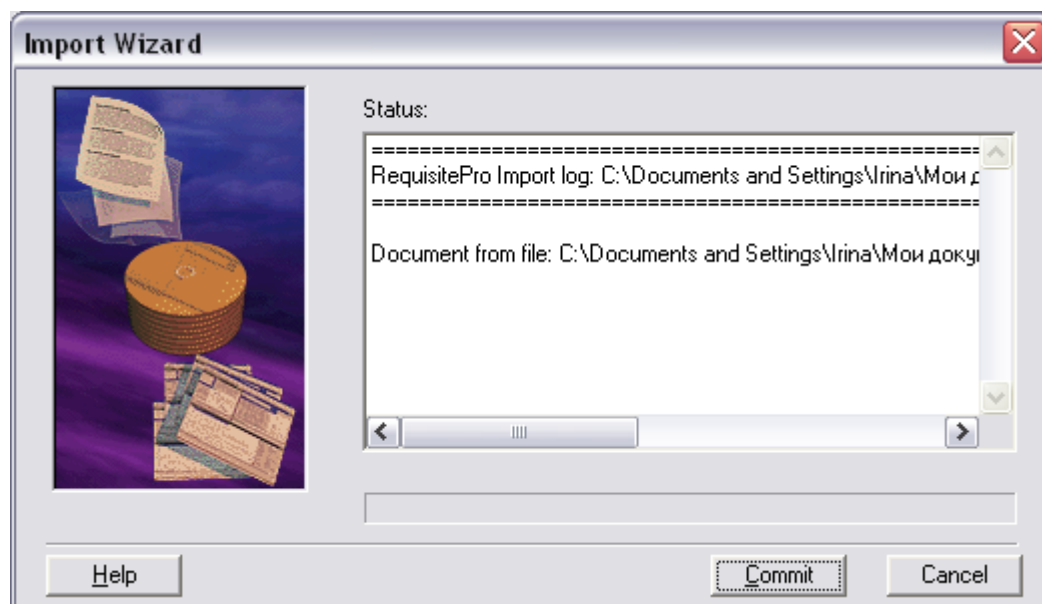


Рис. 26. Вікно Майстра імпорту Status

9. Перейдіть у RequisitePro. У вікні проекту буде відображений створений документ *План управління вимогами* (рис. 27).

10. При імпорті документа в робочій папці проекту створюється файл *План управління вимогами.RMP*, у якому зберігається сам документ, а також файл *import.log*, у якому міститься інформація про час

виконання, джерело і результат імпорту (рис. 28). Зайдіть в робочу папку проекту і перегляньте ці файли.

11. Поверніться в RequisitePro. Двічі клацніть на іконці документа План управління вимогами. Заповніть документ інформацією, якої бракує в проекті.

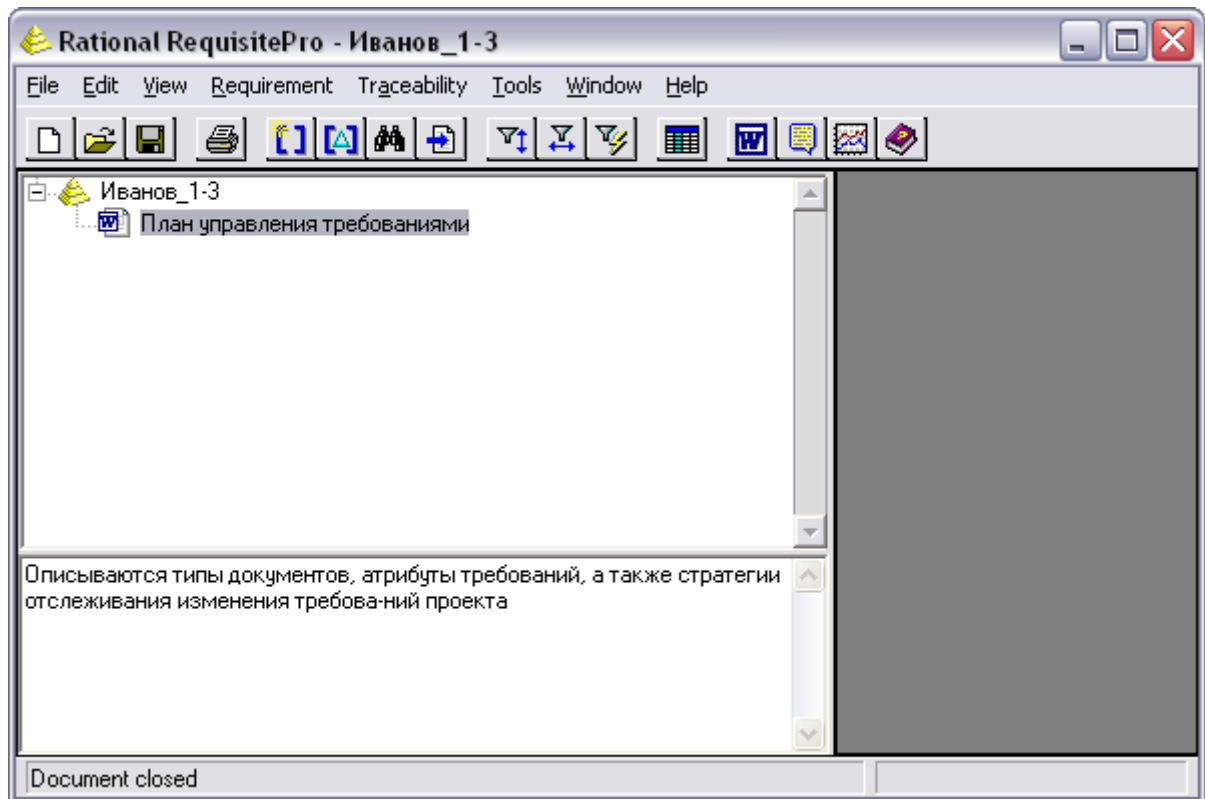


Рис. 27. Вікно проекту з документом *План управління вимогами*

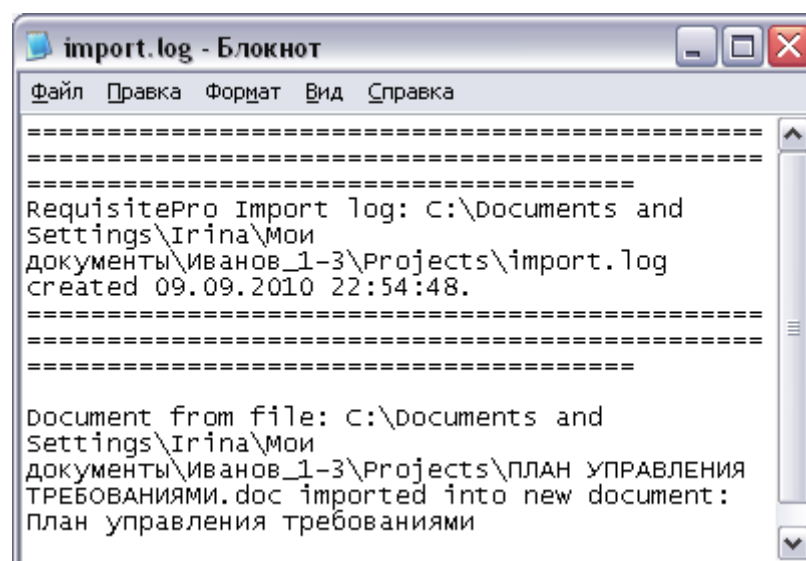


Рис. 28. Файл *import.log*

12. Збережіть зміни в документі, виконавши команду Файл(Зберегти. При цьому зміни зберезуться в документі *План управління вимогами.RMP*.

13. Збережіть документ у форматі Word. Для цього на панелі інструментів в Word виберіть пункти RequisitePro→Document→Save As і введіть ім'я файлу.

Завдання 5.2. Створення документа Глосарій у RequisitePro.

У завданні 5.2 потрібно виконати наступне:

1. Створити документ *Глосарій* на основі шаблону даного документа.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Глосарій – це словник основних використовуваних термінів. Цей документ є найпершим результатом концептуального аналізу предметної області. *Глосарій* можна розглядати, як документ, що засвідчує спільне розуміння основної термінології *Замовником і Розробником*.

Крім того, *Глосарій* є відправною точкою для побудови більш розгорнутих моделей предметної області, які, на стадії реалізації інформаційної системи, лягають в основу об'єктної моделі (для об'єктно-орієнтованих застосувань) і моделі даних (для генерації схеми бази даних).

Глосарій оформляється, як текст, що складається з абзаців, кожен з яких визначає значення одного з термінів предметної області. Термін зазвичай виділяють напівжирним кеглем. Іноді предметну область доцільно сегментувати на ряд "під областей" (subject areas). Тоді кожній з них у глосарії виділяється окремий параграф.

Порядок виконання

1. Створіть папку для *Глосарію*. Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки". У меню, що з'явилося, виберіть пункт New→Package. На екрані з'явиться вікно Package Properties (Властивості папки), в якому потрібно заповнити наступні поля (рис. 29):

у полі Name введіть ім'я папки – *Глосарій*;

у полі Description введіть опис призначення папки;

натисніть кнопку ОК.

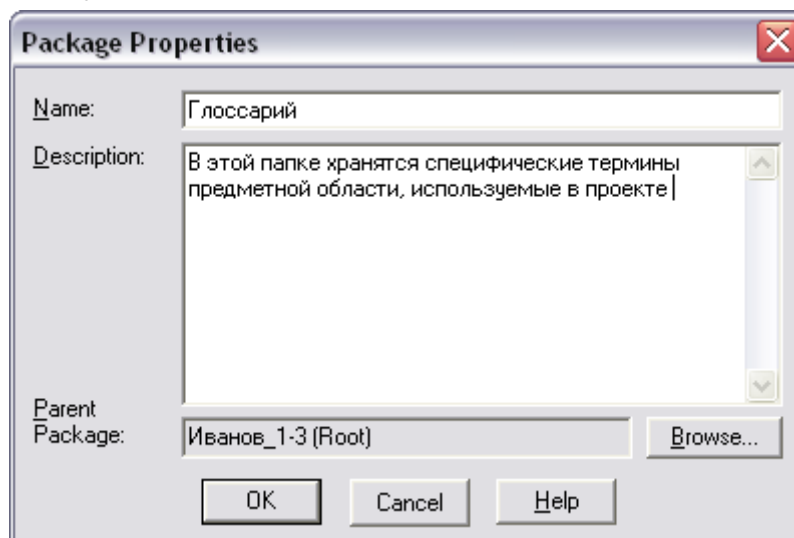


Рис. 29. Вікно Package Properties для Глосарію

2. Клацніть правою кнопкою миші по папці *Глосарій*. У меню, що з'явиться, виберіть пункт меню New→Document. Відкриється вікно Document Properties, у якому потрібно заповнити наступні поля (рис. 30):

у полі Name введіть ім'я документа – *Глосарій*;

у полі Description введіть опис призначення документа;

у полі File Name введіть ім'я файлу, в якому зберігатиметься документ у RequisitePro;

включите прапорець Show Tags;

задайте директорію для зберігання документа в полі Directory.

Натисніть кнопку Browse для завдання директорії;

у полі Document Type натисніть "Стрілку вниз", і виберіть тип документа – *Глосарій*;

натисніть кнопку ОК.

3. У Word відкриється створений новий документ Глосарій, який містить шаблон цього типу документа. Перегляньте документ в Word.

4. Заповніть документ *Глосарій* необхідною інформацією відповідно до шаблону. Введіть визначення основних термінів, що стосуються предметної області проекту.

5. Збережіть документ у форматі Word. Для цього на панелі інструментів у Word виберіть пункти RequisitePro→Document→Save As і введіть ім'я файлу.

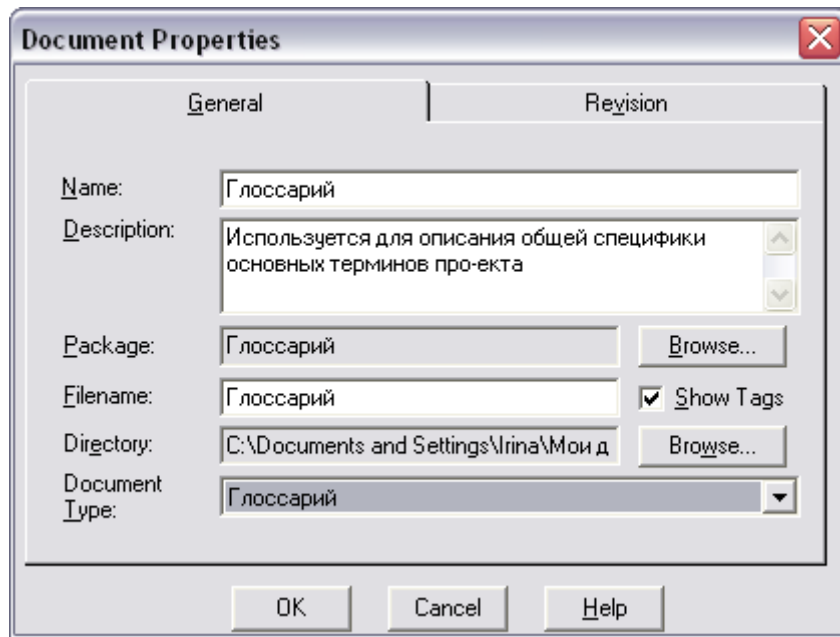


Рис. 30. Вікно **Document Properties** для *Глосарію*

Завдання 5.3. Створення вимог у документі Глосарій

У завданні 5.3 потрібно виконати наступне:

1. Створити вимоги типу *Термін глосарію* (TERM) в документі *Глосарій*.
2. Зберегти вимоги в документі і БД.

Вимоги в RequisitePro можуть створюватися безпосередньо в документі, базі даних або імпортуватися із зовнішнього джерела.

Порядок виконання

1. Перейдіть з RequisitePro в Word, клацнувши на його іконці в панелі інструментів.
2. Виберіть в Word пункти меню RequisitePro→Document→Open. У вікні Open Document виберіть документ *Глосарій* і натисніть кнопку ОК (рис. 31). Вибраний документ відкриється у вікні Word.
3. Перейдіть до розділу документа, де приведені визначення термінів проекту.
4. Виділіть текст, який містить визначення терміну. Далі в меню виберіть пункти RequisitePro→Requirement→New. З'явиться вікно для створення вимоги (рис. 32), в якій потрібно в полі Name ввести назву терміну, а потім натиснути кнопку ОК.

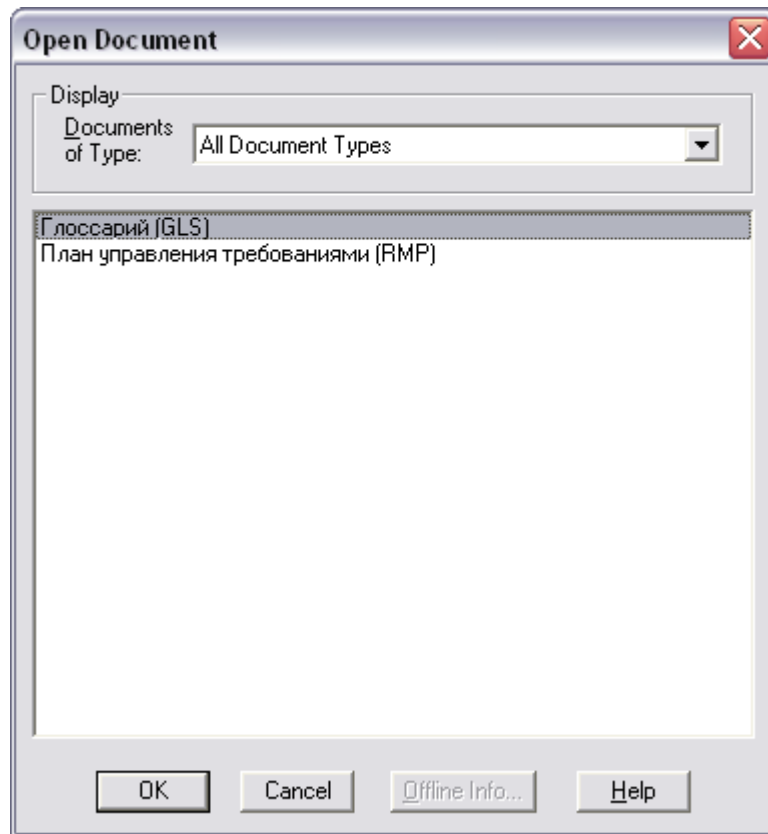


Рис. 31. Вікно Open Document

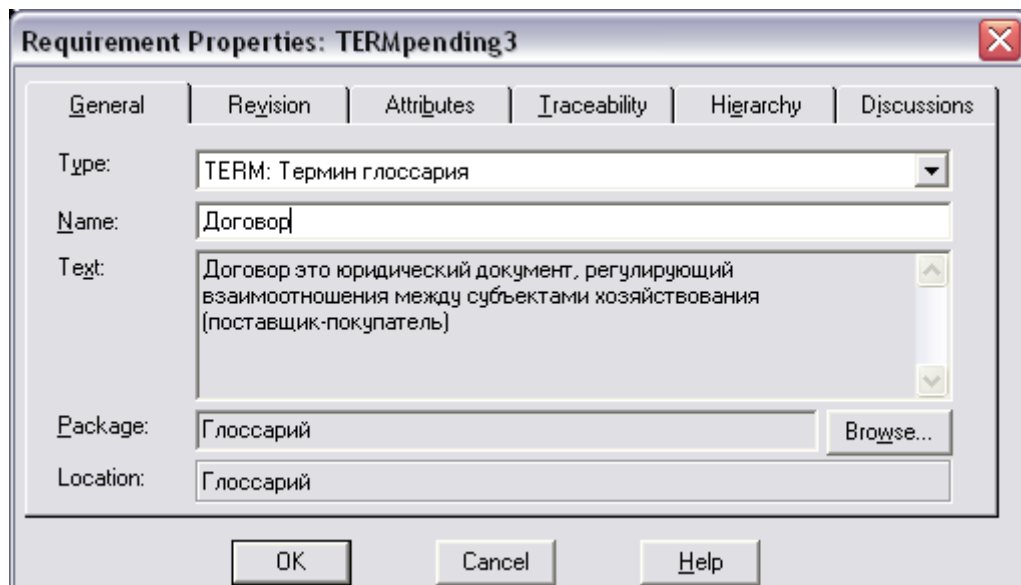


Рис. 32. Вікно Requirement Properties для терміну Глосарію

Після оброблення цієї вимоги термін виглядатиме в документі таким чином:

TERMpending1 Договір – це юридичний документ, який регулює взаємовідносини між суб'єктами господарювання (постачальник – покупець)

Важно! Не можна створювати вимоги всередині поля Word, оскільки RequisitePro не зможе звернутися до них правильно при подальшому використанні вимоги.

5. Аналогічно створіть всі інші вимоги термінів *Глосарію*.

6. Виберіть в меню пункти RequisitePro→Document→Save. Всі створені вимоги фіксуються в базі даних. Зверніть увагу, що всі вимоги, що чекали оброблення, тепер зафіксовані як офіційні вимоги і мають такий вигляд в документі:

TERM4 Договір – це юридичний документ, який регулює взаємовідносини між суб'єктами господарювання (постачальник – покупець)

7. Закрийте документ *Глосарій*, вибравши пункти меню RequisitePro→Document→Close.

8. Перейдіть у вікно RequisitePro. Створені вимоги відіб'ються у вікні проекту в папці *Глосарій* (рис. 34)

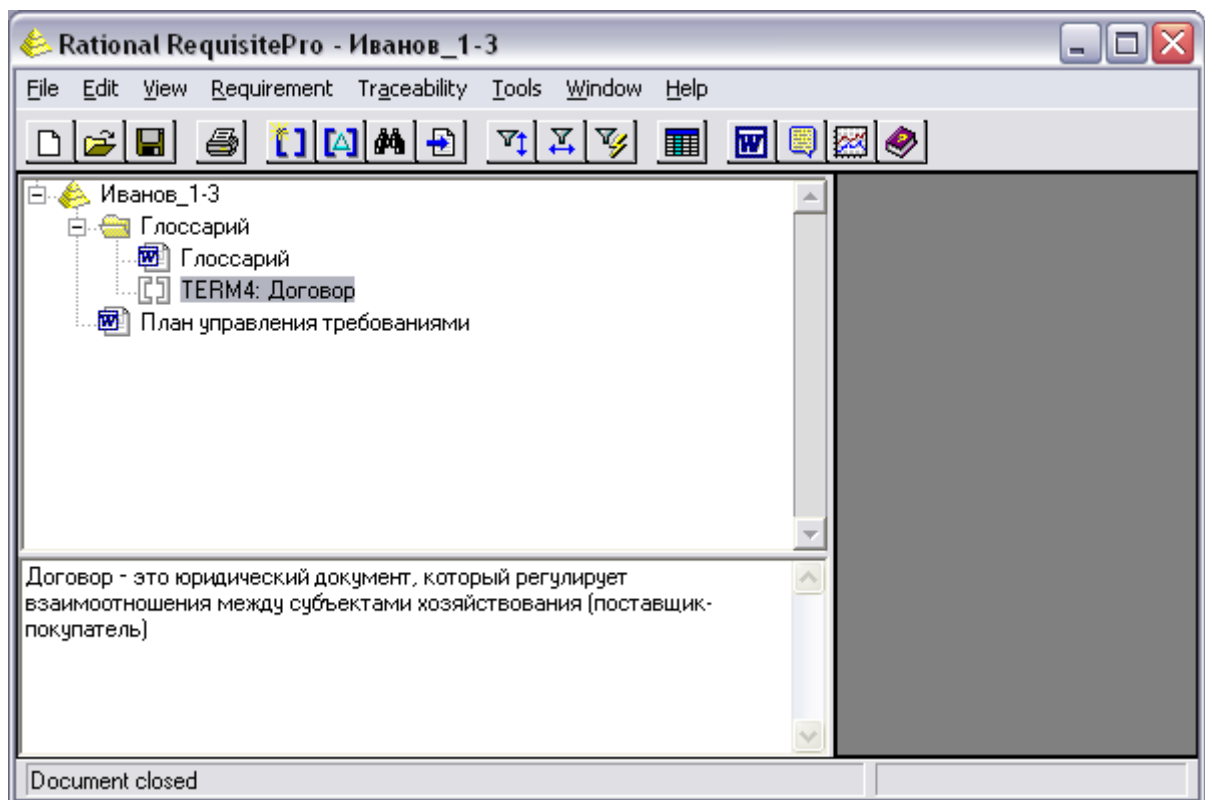


Рис. 34. Вимога – термін

Завдання 5.4. Створення представлення *Матриця вимог з атрибутами* для термінів глосарію

У завданні 5.4 потрібно виконати наступне:

1. Створити представлення типу *Матриця вимог з атрибутами* для типу вимоги *Термін глосарію* (TERM).
2. Ввести значення атрибутів вимог у матриці атрибутів.

Тип представлення *Матриця вимог з атрибутами* є відображенням на екрані вимог певного типу і їх атрибутів. Вимоги відбиваються в рядках. Опис вимоги включає мітку типу вимоги, за якою слідує номер вимоги і власне опис вимоги.

Значення атрибутів вимог відбиваються в стовпцях під заголовком з відповідною назвою атрибуту вимоги. *Матриця вимог з атрибутами* вимог відображає всі вимоги. У цій матриці можуть створюватися нові вимоги, які зберігатимуться в БД.

Порядок виконання

1. Клацніть правою клавішею миші по папці *Глосарій* і в меню, що з'явиться, виберіть пункт New→View. Відкривається вікно View Properties (рис. 35). У цьому вікні потрібно заповнити наступні поля:

у полі Name введіть ім'я представлення – *Всі терміни глосарію*;

у полі Description введіть опис призначення представлення;

у полі View Type (тип області представлення) натисніть "Стрілку вниз" і виберіть тип області представлення Attribute Matrix;

у полі Row Requirement Type натисніть "Стрілку вниз" і виберіть тип вимоги TERM: термін глосарію;

натисніть кнопку ОК.

2. На екрані відобразиться матриця представлення всіх вимог *Глосарію* (рис. 36). Перегляньте значення атрибутів вимог. Заповніть значення атрибуту неоднозначність для кожної вимоги.

4. Роздрукуйте представлення *Всі терміни глосарію*, виконавши команду File→Print.

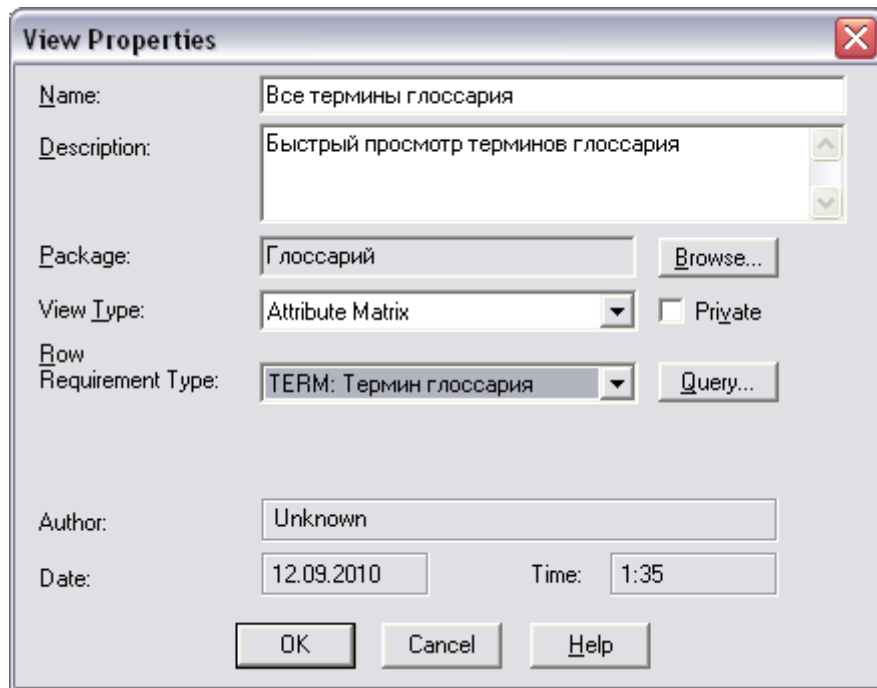


Рис. 35. Вікно View Properties для всіх термінів глосарію

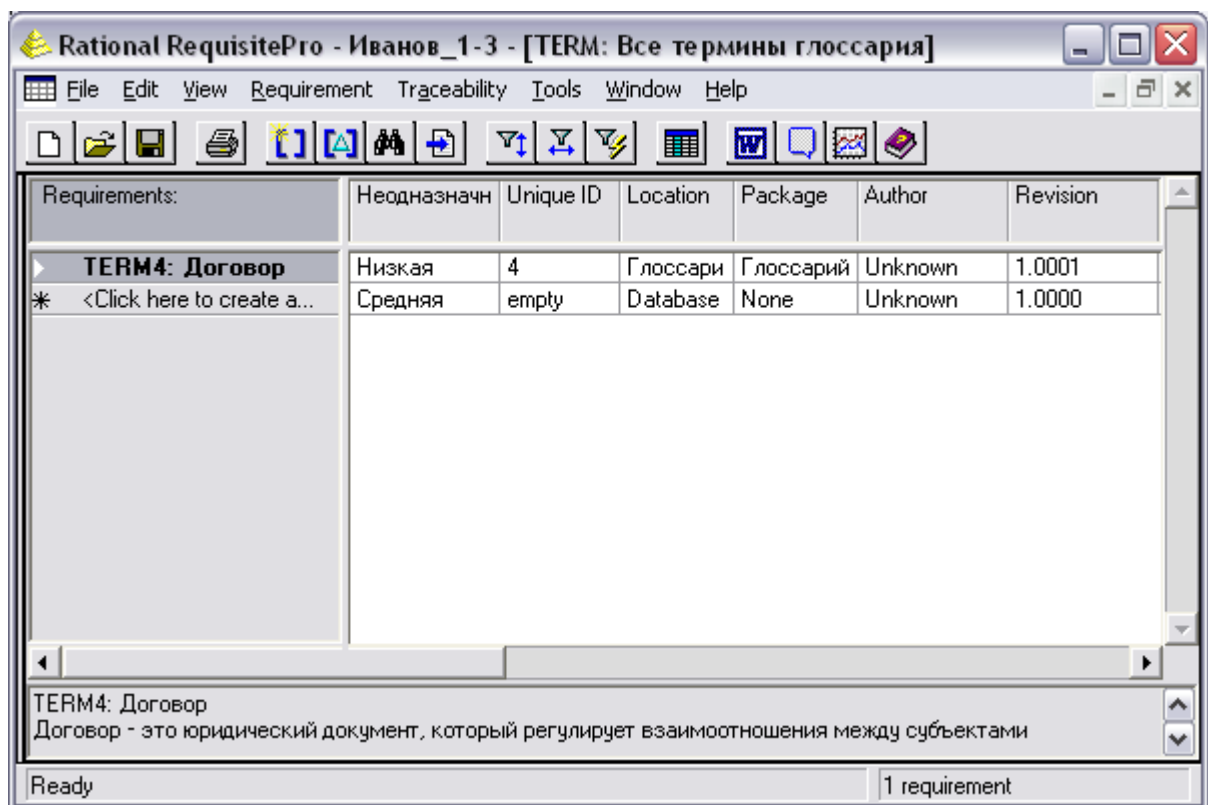


Рис. 36. Представлення Всі терміни глосарію

Завдання 5.5. Редагування вимог у документі і в БД

1. Вилучити вимогу з документа і помістити його до БД.
2. Вилучити вимогу з БД і помістити його в документ.
3. Зняти виділення з вимоги в документі, вилучити вимогу з документа і БД.

При роботі з Word існують різні варіанти доступу до функцій Cut/Copy/Paste в Word:

традиційні, за допомогою пункту *Правка* → *Вирізати* (*Копіювати*, *Вставити*);

через панель інструментів RequisitePro → Requirement → Cut (Copy, Paste).

Панель інструментів RequisitePro в Word призначена для обробки вимог всередині Word. Тому в Word, використовуйте панель інструментів RequisitePro для операцій Cut/Copy/Paste для роботи з вимогами, а панель інструментів Word для обробки тексту.

Порядок виконання

1. Переміщення вимоги з документа в БД.
 - 1.1. Двічі клацніть на вимогу у вікні представлення. При цьому відбувається перехід до документа *Глосарій*, у якому міститься вимога. Вимога в документі позначається кольором.
 - 1.2. Помістіть курсор на вимогу, що знаходиться в документі Word. Виберіть на панелі інструментів пункт RequisitePro → Requirement → Cut. Вимога буде повністю вилучена з документа і з БД.
 - 1.3. Перейдіть у вікно представлення матриці атрибутів вимог. Виберіть пункт меню Edit, а потім Past. Вимога буде вставлена до БД.
2. Переміщення вимоги з бази даних у документ.
 - 2.1. Створіть в матриці з атрибутами вимог нову вимогу, наприклад, термін – постачальник.
 - 2.2. З матриці атрибутів виріжте цю вимогу. Вирізана вимога в матриці атрибутів відображається з позначкою (cat).
 - 2.3. Перейдіть у вікно Word. Помістіть курсор у документі там, де буде вставлено вимогу. Виберіть пункт RequisitePro → Requirement → Paste. Після виконання цієї команди вимога буде знаходитися і в документі, і в базі даних.
3. Зняття виділення з вимоги і вилучення вимоги.

3.1. На панелі інструментів виберіть пункти меню RequisitePro→Requirement→Delete→Unmark. При появі запиту про підтвердження дії, натисніть Yes. Ця дію знімає виділення вимоги. Вимога зберігається в документі Word як текст.

3.2. Помістіть курсор у тексті будь-якої вимоги. На панелі інструментів виберіть пункти RequisitePro→Requirement→Delete→Remove. При появі запиту про підтвердження дії, натисніть Yes. На відміну від функції Delete→Unmark, функція Delete→Remove видаляє і вимогу, і текст.

3.3. У вікні представлення матриці атрибутів вимог виділіть вимогу, створену в базі даних. Виділити вимогу можна клацанням лівої кнопки миші на вимозі. В області представлення виберіть пункти меню Edit→Delete. При появі запиту про підтвердження дії, натисніть Yes. Виконані дії в області представлення не вимагають збереження документа для підтвердження змін.

Увага! При вилученні вимог через Delete→Unmark, Delete→Remove або Requirement Delete втрачається вся хронологія змін і параметри налаштування атрибутів вимог. Замість того, щоб вилучати непотрібні вимоги, розгляньте можливість збереження їх в базі даних з, наприклад, атрибутом "вилучений".

Зміст звіту з лабораторної роботи № 5

1. Мета роботи.

2. Імпорт документа *План управління вимогами*.

Привести вікно вмісту файла import.log.

3. Створення документа *Глосарій*.

Навести вікно Document Properties для глосарію, роздруківку документа *Глосарій* з виділеними вимогами з текстового редактора Word.

4. Створення представлення *Матриця вимог з атрибутами* для термінів глосарію

Навести роздруківку вікна "View Properties" для представлення *Всі терміни глосарію*, вікно проекту RequisitePro з матрицею вимог для всіх вимог глосарію.

5. Висновки.

Контрольні питання

1. Для чого призначений документ *План управління вимогами*?
2. Яка інформація міститься в документі *План управління вимогами*?
3. Яким чином здійснюється імпорт документів у RequisitePro?
4. Які поля потрібно заповнити у вікні Document Properties для документа *План управління вимогами*?
5. У якому файлі зберігається документ в RequisitePro?
6. Який вміст файлу *import.log*?
7. Для чого призначений документ *Глосарій*?
8. Яким чином можна сегментувати терміни в глосарії?
9. Як створити папку в RequisitePro?
10. Яка інформація зберігається в папці *Глосарій* в RequisitePro?
11. Які поля потрібно заповнити у вікні Document Properties для документа *Глосарій*?
12. Як зберегти документ RequisitePro у форматі Word?
13. Які існують способи створення вимог в RequisitePro?
14. Яким чином можна створити вимогу в документі?
15. Яким чином можна створити вимогу в базі даних?
16. Що представляє собою матриця вимог з атрибутами?
17. Як створюється матриця вимог з атрибутами?
18. Яким чином вводяться значення атрибутів для вимог?
19. Які існують способи редагування вимог у базі даних і в документі? Які для цього використовуються команди в RequisitePro?
20. Які наслідки виконання команд Delete→Unmark, Delete→Remove або Requirement→Delete?

Лабораторна робота № 6

Виявлення високорівневих вимог, створення документа *Бачення* в RequisitePro

Цілі лабораторної роботи:

1. Виявлення і опис високорівневих вимог до системи
2. Створення документа *Бачення* на основі шаблону документа.

Методичні вказівки до виконання

Завдання 6.1. Створення документа *Бачення* в RequisitePro

У завданні 6.1 потрібно виконати наступне:

1. Створити документ *Бачення* на основі шаблону даного документа.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Бачення проекту (Vision) – це короткий опис суті майбутнього продукту. У цьому документі коротко описується, що є продуктом, які цілі і завдання його створення, хто його користувачі і які основні можливості майбутньої системи.

Призначення *Бачення*:

дати загальне уявлення про продукт. *Бачення* дозволяє за допомогою декількох абзаців ознайомити з суттю проекту будь-яку зацікавлену особу;

зібрати високорівневі вимоги. Документ дає загальне уявлення про можливості, які будуть реалізовані продуктом.

Для формування документа *Бачення* необхідно виконати наступні кроки:

- 1) формулювання проблем.

Для опису проблем у RUP пропонується шаблон, представлений у таблиці 1.

Таблиця 1.

Проблема	(опис проблеми)
Стосується	(співвласники, яких стосується проблема)
Її наслідком є	(який вплив проблеми)
Успішне рішення	(список деяких ключових переваг від успішного рішення)

- 2) ідентифікація співвласників і користувачів.

Ідентифікація співвласників і користувачів передбачає пошук і фіксацію зацікавлених осіб проекту – стейкхолдерів (від англ. stakeholder) – представників організації Замовника і організації Розробника, інвесторів, зовнішніх експертів і ін.

У PMI PMBOK® стейкхолдери проекту визначаються як індивідууми або організації, чиї інтереси можуть бути зачеплені внаслідок виконання або завершення проекту. Типами стейкхолдерів можуть бути користувачі і співвласники: департаменти управління, технічні розробники і т. п.

До основних стейкхолдерів проекту відносяться:

керівник проекту (Project Manager) – особа, що відповідає за управління проектом перед керівним комітетом проекту;

продавець (Sales Person) – особа, що відповідає за підписання контракту;

замовник (Customer) – особа або організація, що використовує результати проекту в своїй діяльності;

організація-виконавець (Performing organization) – організація, що надає ресурси для виконання проекту;

спонсор (Sponsor) – особа, що забезпечує фінансування і що підтримує реалізацію проекту;

експерт за якістю (Quality Assurer) – особа, що відповідає за контроль якості проекту;

системний архітектор (System Architect) – особа, що відповідає за технічне вирішення проекту.

Крім перерахованих вище в конкретному проекті можуть бути і інші стейкхолдери: субпідрядники, постачальники, державні організації, власники підприємств та ін.

3) визначення меж системи.

Визначенням меж системи є творчий нетривіальний процес. Для цього використовують контекстні діаграми. RUP у пошуку меж пропонує відштовхуватися від акторів і варіантів використання.

4) ідентифікація обмежень.

Серед джерел обмежень звично виділяють: політичні, економічні, середовища, технічні, виконання, системні.

5) формулювання постановки завдань.

Визначення складу завдань; цілей, призначення і термінів їх виконання; всіх користувачів і інших зацікавлених осіб.

6) визначення можливостей системи.

Опис можливостей системи є формулюванням високорівневих вимог.

7) оцінка результатів.

Оцінюються передбачувані результати проекту.

Шаблон документа *Бачення* в RUP містить наступні основні розділи:

- вступ;
- позиціонування;
- описи співвласників і користувачів;
- короткий огляд продукту;
- можливості продукту;
- обмеження;
- показники якості;
- старшинство і пріоритети;
- інші вимоги до продукту;
- вимоги до документації;
- додаток.

У *Вступі* описуються мета документа, його контекст (зв'язок і взаємовплив з різними проектами), визначення, акроніми і скорочення, посилання на інші документи, короткий зміст.

У розділі *Позиціонування* поміщається визначення вирішуваної проблеми (проблем), вказується цільовий замовник і досліджуються ділові переваги продукту перед аналогічними на ринку.

У описі співвласників і користувачів, окрім власне опису цих двох груп, досліджується демографія ринку: цільові ринкові сегменти, розмір і темпи зростання ринку, існуючі конкурентні пропозиції на ринку, репутація Розробника на ринку.

Короткий огляд продуктів містить резюме продукту, опис його перспектив і ключових можливостей, припущення і залежності, вказується вартість і її калькуляція, розглядаються питання ліцензування і інсталяції.

У розділі *Можливості продукту* виконується докладний опис кожної можливості – в окремому параграфі. Можливості (Feature) – це високорівневі здібності системи, які необхідні для отримання користувачами очікуваної вигоди. Кожна можливість – бажане зовнішнє обслуговування, яке зазвичай вимагає **введення інформації**, щоб досягти бажаного результату. При описі слід уникати формулювання реалізації можливості. Описувати можливість слід на загальному рівні, доступному для розуміння всіма. Потрібно зосередити увагу на

необхідності можливості, на тому, чому (а не як) вона має бути реалізована.

У розділ *Обмеження* слід виносити існуючі технічні, технологічні та ін. обмеження, які необхідно враховувати на даній стадії, зовнішні зв'язки або інші залежності.

Розділ *Показники якості* містить опис найбільш суттєвих нефункціональних вимог до системи (ефективності, надійності, відмовостійкості, застосовності і інші подібні характеристики, які не зафіксовані в наборі можливостей.).

Розділ *Старшинство і пріоритети* ранжирує сформульовані раніше вимоги і можливості системи в міру важливості, черговості реалізації тощо.

Розділ *Інші вимоги до продукту* описує вживані стандарти, вимог апаратних засобів або платформи, експлуатаційні вимоги, вимоги до довкілля.

У *Вимогах до документації* описується документація, яка має бути розроблена для підтримки успішного розгортання додатку, наводяться ключові характеристики посібника користувача, інтерактивної довідки, керівництва з установки і конфігурації, файлу Read Me.

У *Додаток* виносяться атрибути можливостей. RUP рекомендує наступний набір атрибутів: статус, вигода, обсяг робіт, ризик, стабільність, цільовий випуск, призначення, причина.

Порядок виконання

1. Створіть папку для *Бачення*. Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки", виберіть пункт меню New→Package. На екрані з'явиться вікно Package Properties, у якому потрібно заповнити наступні поля (рис. 37):

- у полі Name введіть ім'я папки – Бачення;
- у полі Description введіть опис призначення папки;
- натисніть кнопку ОК.

2. Клацніть правою кнопкою миші по папці *Бачення*. У меню, що з'явиться, виберіть пункт меню New→Document. Відкриється вікно Document Properties, у якому потрібно заповнити наступні поля (рис. 38):

- у полі Name введіть ім'я документа – Бачення;
- у полі Description введіть опис призначення документа;

у полі File Name введіть ім'я файла, в якому зберігатиметься документ у RequisitePro;
включіть прапор Show Tags;
задайте директорію для зберігання документа в полі Directory.
Натисніть кнопку Browse для завдання директорії;
у полі Document Type натисніть "Стрілку вниз", і виберіть тип документа *Бачення*;
натисніть кнопку ОК.

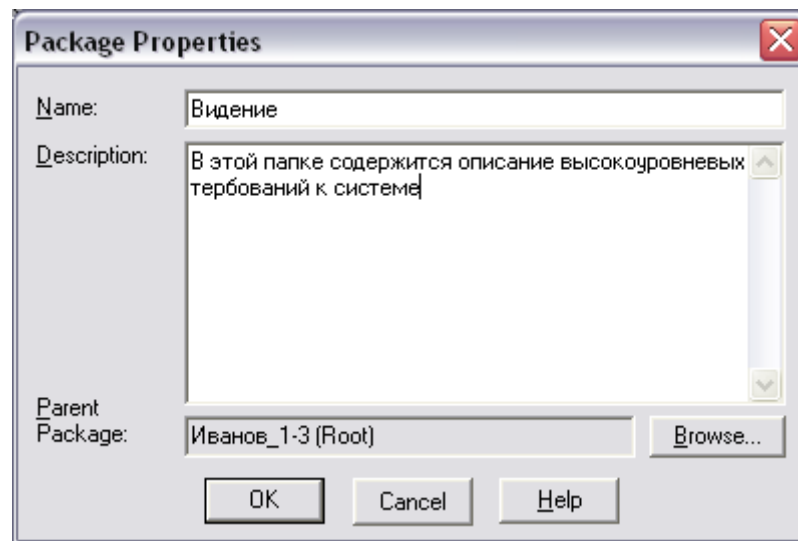


Рис. 37. Вікно Package Properties для *Бачення*

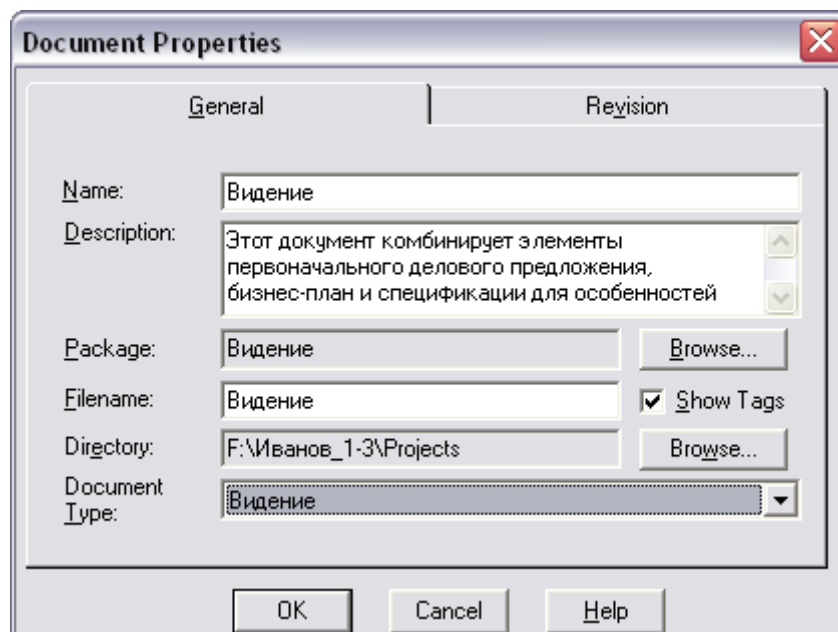


Рис. 38 Вікно Document Properties для *Бачення*

3. У Word відкриється створений новий документ *Бачення*, який містить шаблон цього типу документа. Перегляньте документ у Word.

4. Заповніть розділи документа *Бачення: Вступ, Позичіонування, Описи співвласників і користувачів* – необхідною інформацією відповідно до шаблону.

5. У розділі *Можливості продукту* створіть пункт для кожного завдання проекту. Введіть короткий опис можливостей (Features) для кожного завдання проекту.

6. У останніх розділах опишіть необхідні обмеження і вимоги до продукту.

5. Збережіть документ у форматі Word.

Завдання 6.2. Створення вимог у документі Бачення

У завданні 6.2 потрібно виконати наступне:

1. Створити вимоги типу *Властивість системи* (FEAT) в документі *Бачення*.

2. Створити вимоги типу *Додаткова вимога* (SUPP) в документі *Бачення*.

3. Зберегти вимоги в документі і БД.

Порядок виконання

1. Активізуйте з RequisitePro документ *Бачення* двічі клацнувши по його іконці у вікні провідника проекту. Документ відкриється у вікні Word.

2. Перейдіть до розділу документа, де наведені описи можливостей продукту. Виділіть текст, який містить опис властивості системи. Далі в меню виберіть пункти RequisitePro→Requirement→New. З'явиться вікно для створення вимоги (у полі Text відіб'ється виділений заздалегідь у тексті документа опис властивості системи). У вікні, що з'явилося, потрібно (рис. 39):

у полі Type із списку типів вимог вибрати тип FEAT: *Властивість системи*;

у полі Name ввести назву властивості системи, а потім натиснути кнопку ОК.

4. Аналогічно створіть всі інші вимоги властивостей системи в розділі *Можливості продукту*.

Увага! Якщо в системі передбачено створення інтерактивної допомоги (розділ *Вимоги до документації* документа *Бачення*), то ця вимога позначається також як властивість системи (FEAT).

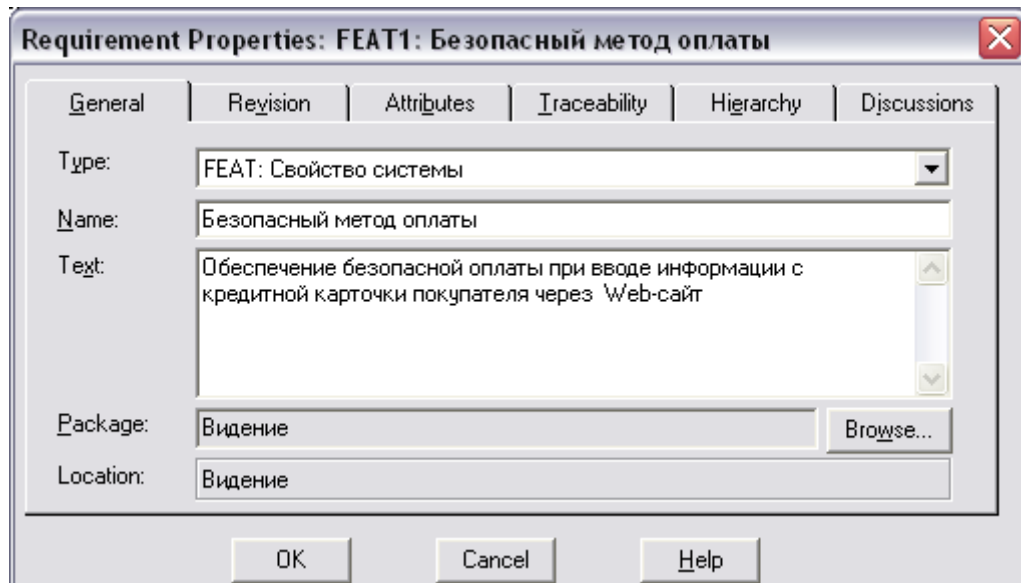


Рис. 39 Вікно Requirement Properties для опису властивості системи

4. Перейдіть до розділів документа, де наведені описи додаткових (нефункціональних) вимог. Виділіть текст, який містить опис додаткову вимогу. Далі в меню виберіть пункти RequisitePro→Requirement→New. З'явиться вікно для створення вимоги (рис. 40), в якій потрібно:

у полі Type із списку типів вимог вибрати тип SUPP: *Додаткова вимога*;

у полі Name ввести назву додаткової вимоги, а потім натиснути кнопку ОК.

6. Аналогічно створіть всі інші додаткові вимоги в документі *Бачення*.

7. Виберіть у меню пункти RequisitePro→Document→Save. Всі створені вимоги фіксуються в базі даних.

8. Закрийте документ *Бачення*, вибравши пункти меню RequisitePro→Document→Close.

8. Перейдіть у вікно RequisitePro. Створені вимоги відіб'ються у вікні проекту в папці *Бачення*.

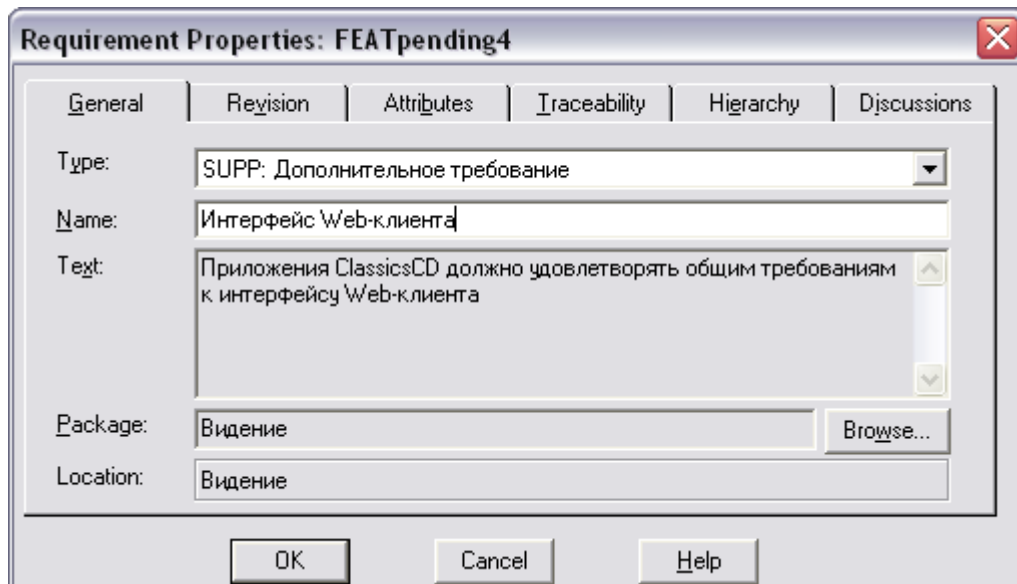


Рис. 40. Вікно Requirement Properties для опису додаткової вимоги

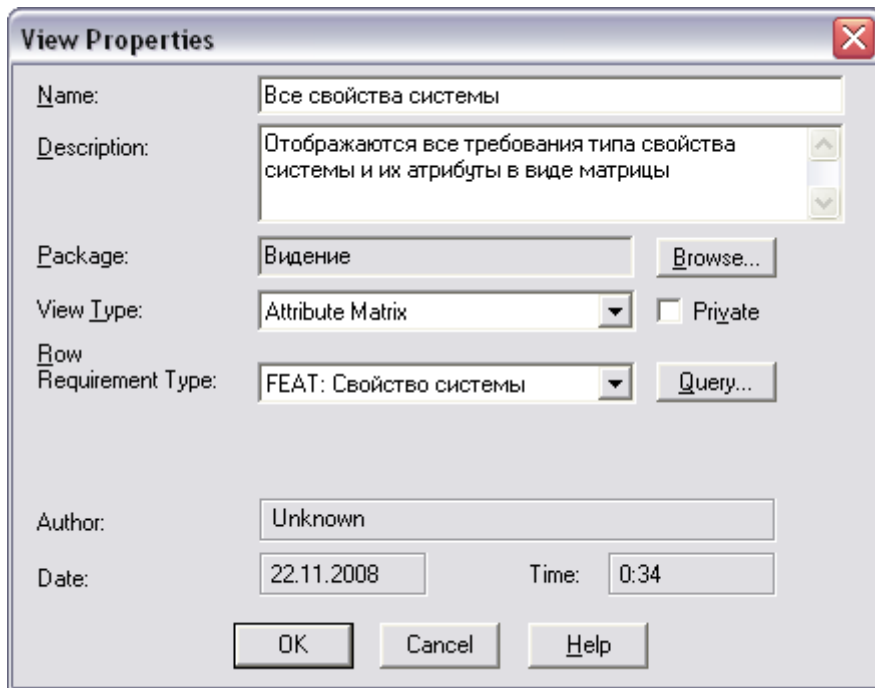
Завдання 6.3. Створення представлення *Матриця вимог з атрибутами* для властивостей системи

У завданні 6.3 потрібно виконати наступне:

1. Створити в папці *Бачення* представлення *Всі властивості системи* для типу вимоги *Властивість системи* (FEAT)
2. Ввести значення атрибутів вимог в матриці атрибутів представлення *Всі властивості системи*.

Порядок виконання

1. Клацніть правою клавішею миші по папці *Бачення* і в меню, що з'явилося, виберіть пункт *New→View*. Відкриється вікно *View Properties* (рис. 41). У цьому вікні потрібно заповнити наступні поля:
 - у полі *Name* введіть ім'я представлення – *Всі властивості системи*;
 - у полі *Description* введіть опис призначення представлення;
 - у полі *View Type* (тип області представлення) натисніть "Стрілку вниз" і виберіть тип області представлення *Attribute Matrix*;
 - у полі *Row Requirement Type* натисніть "Стрілку вниз" і виберіть тип вимоги FEAT: *Властивість системи*;
 - натисніть кнопку *OK*.



**Рис. 41. Вікно View Properties для представлення
*Всі властивості системи***

2. На екрані відобразиться матриця представлення *Всі властивості системи*. Перегляньте значення атрибутів вимог. Заповніть значення всіх атрибутів для кожної вимоги.

3. Роздрукуйте представлення *Всі властивості системи* виконавши команду File→Print.

Зміст звіту по лабораторній роботі № 6

1. Мета роботи.

2. Створення документа *Бачення*.

Привести вікно "Document Properties" для *Бачення*, роздруківку для документа *Бачення* з виділеними вимогами, виконану з текстового редактора Word.

3 Створення представлення *Матриця вимог з атрибутами* для всіх властивостей системи.

Привести роздруківку вікна View Properties для представлення *Всі властивості системи*, роздруківку представлення *Всі властивості системи*.

4. Висновки.

Контрольні питання

1. Що представляє собою документ *Бачення*?
2. Яке призначення документа *Бачення*?
3. Які кроки потрібно виконати для формування документа *Бачення*?
4. Яка структура документа *Бачення*?
5. Яким чином описується формулювання проблем, для вирішення яких створюється продукт?
6. Хто такі стейкхолдери проекту? Яке визначення стейкхолдерів дається в РМІ РМВОК®? Охарактеризуйте основних стейкхолдерів проекту.
7. Хто такі співвласники проекту? Наведіть приклади співвласників. Хто є співвласниками вашого проекту?
8. Яким чином описуються профілі співвласників проекту?
9. Хто такі користувачі системи? Наведіть приклади користувачів. Хто є користувачами вашої системи?
10. Яким чином описуються профілі користувачів системи?
11. Що таке можливість системи? До якого типу вимоги вона відноситься.
12. Наведіть повні назви завдань, які розробляються у вашому проекті.
13. Перерахуйте властивості системи, що реалізуються в кожному завданні вашої системи.
14. У якому розділі *Бачення* описується вимога створення інтерактивної допомоги і до якого типу вимоги відноситься ?
15. Які нефункціональні вимоги і в яких розділах наводяться в документі *Бачення*?

Лабораторна робота № 7

Специфікація ключових УС, створення документа Специфікація варіанту використання в RequisitePro

Цілі лабораторної роботи:

1. Опис основного потоку подій ключових УС.
2. Опис альтернативного потоку подій ключових УС.

3. Створення документа *Специфікація варіанту використання* на основі шаблону документа.

Методичні вказівки до виконання

Завдання 7.1. Створення документа *Специфікація варіанта використання*

У завданні 7.1 потрібно виконати наступне:

1. Створити документ *Специфікація варіанта використання* на основі шаблону даного типу документа для кожного завдання.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Підвищення інформативності вимог здійснюється на основі оформлення їх у вигляді UC - варіанта використання (use case).

Перш, ніж приступити власне до специфікації вимог у формі варіанта використання, в RUP звичайно складають реєстр (список) акторів (actors) і варіантів використання.

Актор – це хтось або щось, що володіє активністю відносно до програмної системи. Актором звичайно буде користувач системи. Окрім користувача як актор може розглядатися інша програмна система, апаратний пристрій, у ряді випадків – активна компонента самої системи. Пошук акторів корпоративної інформаційної системи зазвичай зводиться до аналізу ролей різних користувачів. Вибір акторів залежить від їх функціональних обов'язків, розмежування доступу, способів використання інформаційної системи.

Варіант використання можна розглядати як функцію, що реалізовується системою. Однак всяка функція повинна мати цінність і давати можливість отримати кінцевий результат для кінцевого користувача продукту або послуги. Тому при специфікації варіанта використання виділяють серед всього функціонала системи лише ту функціональність, яка:

- 1) корисна конкретному кінцевому користувачеві;
- 2) дозволяє отримувати користувачеві конкретні закінчені результати.

Шаблон опису варіанта використання в методології RUP містить наступні розділи:

1. Найменування і короткий опис.

У цьому розділі вказується: найменування варіанта використання, актори варіанта використання, короткий (у один абзац) опис варіанта використання.

2. Потік подій.

2.1. Основний потік подій.

Тут перераховуються кроки основного потоку, починаючи від тригера і аж до досягнення гарантії успіху.

Тригер – подія предметної області, що викликає використання прецеденту (варіанта використання).

Гарантія успіху – це те, що отримують актори-учасники в разі успішного досягнення мети.

2.2. Альтернативні потоки подій.

Кожен з альтернативних сценаріїв описується в окремому параграфі, в тому ж стилі, що і основний потік подій. Альтернативні сценарії описують поведінку системи при будь-яких відхиленнях від основного сценарію, а також поведінку у виняткових ситуаціях.

3. Додаткові вимоги.

Тут перераховуються нефункціональні вимоги, що мають безпосереднє відношення саме до цього варіанта використання.

4. Передумови.

Передумова – прецедент, який має бути обов'язково виконаний, щоб можна було виконати даний УС. Передумову описує стан, у якому система повинна перебувати до початку виконання прецеденту.

5. Постумови.

Постумова – прецедент, який обов'язково має бути виконаний після виконання даного УС. Постумова – це стан, у якому система повинна перебувати після закінчення виконання прецеденту. Постумова – це те, що гарантується акторам-учасникам, не залежно від успіху виконання даного УС. Наприклад, – у разі невдалої транзакції всі дані, що були в системі до її початку, зберігаються незмінними.

Події, що описуються передумовами або постумовами, мають бути станами, які користувач може спостерігати.

При створенні вимог типу варіант використання треба дотримуватися наступних умов.

1. Кожний документ *Специфікація варіанту використання* описує вимогу типу варіант використання, яка має структуру "батько-нащадок".

3. Назвою вимоги-батька буде назва варіанта використання.

4. Вимогами-нащадками будуть: короткий опис варіанта використання; основний потік подій; альтернативний потік 1, альтернативний потік 2, ... і т. д.; додаткова умова 1, додаткова умова 2, і т. д.; передумова; постумова.

Порядок виконання.

1. Створіть папку Специфікації варіантів використання. Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки", виберіть пункт меню New→Package. На екрані з'явиться вікно Package Properties, у якому потрібно заповнити наступні поля:

у полі Name введіть ім'я папки – *Специфікації варіантів використання*;

у полі Description введіть опис призначення папки.

2. У папці *Специфікації варіантів використання* створіть папки для кожного варіанта використання. Для цього клацніть правою клавішею по зображенню папки *Специфікації варіантів використання*. У меню, що з'явилося, виберіть пункт меню New→Package. На екрані з'являється вікно Package Properties, у якому потрібно заповнити наступні поля:

у полі Name введіть ім'я папки – Назва варіанта використання;

у полі Description введіть опис призначення папки.

3. Клацніть правою кнопкою миші по папці для завдання. У меню, що з'явилося, виберіть пункт New→Document. Відкриється вікно Document Properties, у якому потрібно заповнити наступні поля:

у полі Name введіть ім'я документа – Назва варіанта використання;

у полі Description введіть опис призначення документа;

у полі File Name введіть ім'я файлу, в якому зберігатиметься документ в RequisitePro;

включіть прапорець Show Tags;

задайте директорію для зберігання документа в полі Directory;

у полі Document Type натисніть "Стрілку вниз", і виберіть тип документа *Специфікація варіанта використання*;

натисніть кнопку ОК.

3. У Word відкриється створений новий документ *Специфікація варіанта використання*, який містить шаблон цього типу документа. Перегляньте документ у Word.

4. Заповніть документ *Специфікація варіанта використання* необхідною інформацією відповідно до шаблону.

5. У розділі *Актори* приведіть основних і додаткових акторів системи.

6. У розділі *Потік подій* опишіть основний і альтернативні потоки подій.

7. У розділі *Додаткові вимоги* опишіть при необхідності додаткові вимоги не охоплені описами варіанта використання.

8. Опишіть передумови і постумови.

9. Збережіть документ у форматі Word.

10. Аналогічно створіть специфікації варіантів використання для інших завдань проекту в відповідних папках з кодами завдань.

Завдання 7.2. Створення вимог у документі *Специфікація варіанта використання*

У завданні 7.2 потрібно виконати наступне:

1. Створити вимоги типу *Варіант використання* (UC) у документі *Специфікація варіанта використання*.

2. Створити ієрархічні вимоги для варіанту використання.

3. Зберегти вимоги в документі і БД.

Порядок виконання

1. Перейдіть з RequisitePro в Word, клацнувши на його іконці на панелі інструментів.

2. Виберіть в Word пункти меню RequisitePro→Document→Open. У вікні Open Document виберіть документ *Специфікація варіанта використання* і натисніть кнопку ОК. Вибраний документ відкриється у вікні Word.

3. Створіть вимогу, яка в подальшому буде вимогою-батьком. Для цього виділіть на сторінці документа що йде після змісту, назву варіанта використання. Далі в меню виберіть пункти RequisitePro→Requirement→New. З'явиться вікно для створення вимоги, в якому потрібно (рис. 42): у полі Type із списку типів вимог вибрати тип UC: *Варіант використання*; у полі Name ввести назву варіанта використання, а потім натиснути кнопку ОК.

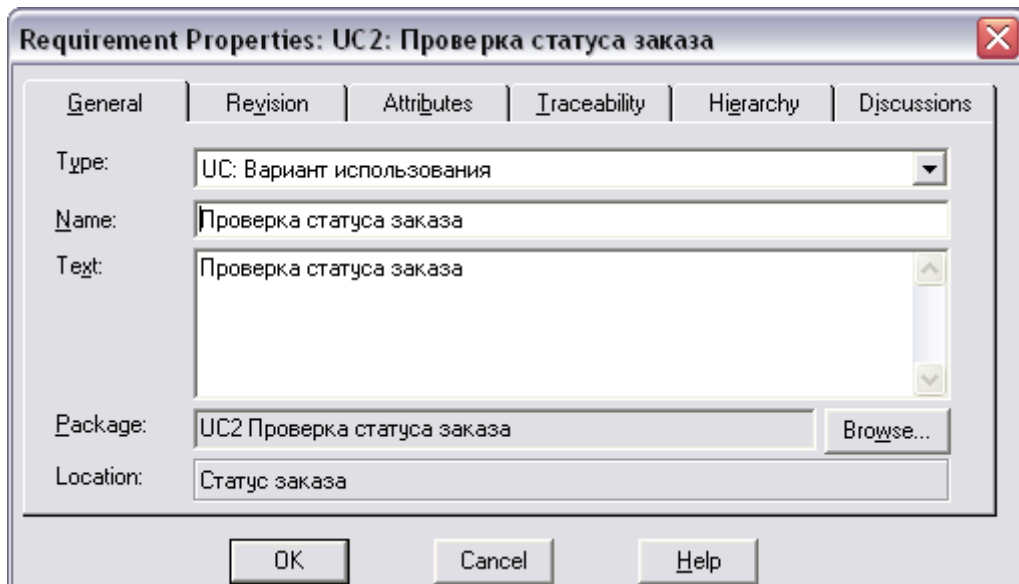


Рис. 42. Вікно для вимоги з назвою варіанта використання

4. Аналогічно створіть усі інші вимоги, які в подальшому будуть вимогами-нащадками. При створенні цих вимог у полі Type із списку типів вимог вибирати тип UC: *Варіант використання*.

4.1. Вимога *Стислий опис*.

У документі виділіть текст з коротким описом варіанта використання. Клацніть на ньому правою кнопкою і виберіть у меню New Requirement. З'явиться вікно для створення вимоги, в якому у полі Name ввести текст *Стислий опис*, а потім натиснути кнопку ОК (рис. 43).

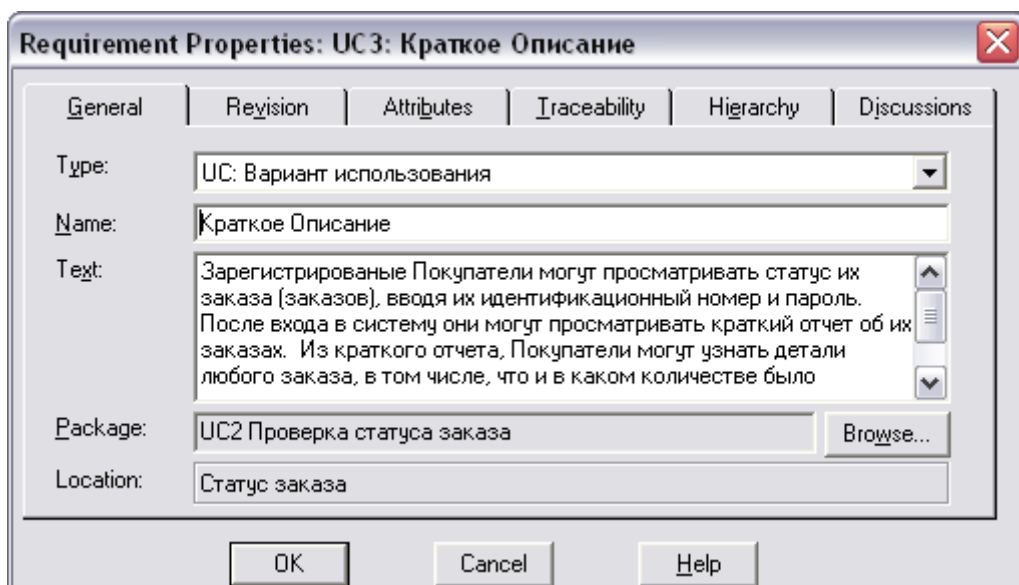


Рис. 43. Вікно для вимоги *Стислий опис*

4.2. Основний потік.

У документі виділити текст, що описує основний потік, починаючи від абзацу *Початок* до останнього абзацу підрозділу *Основний потік*. Клацніть на ньому правою кнопкою і виберіть у меню *New Requirement*. З'явиться вікно для створення вимоги, в якому у полі *Name* ввести текст *Основний потік*, а потім натиснути кнопку *OK* (рис. 44).

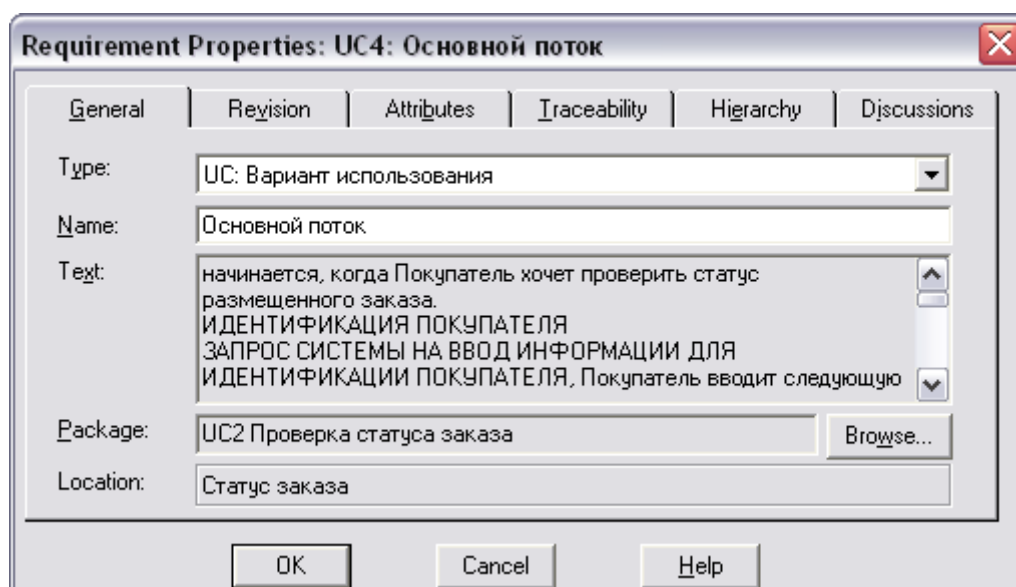


Рис. 44. Вікно для вимоги *Основний потік*

4.3. Альтернативні потоки.

У документі виділіть текст з назвою і описом першого альтернативного потоку. Клацніть на ньому правою кнопкою і виберіть в меню *New Requirement*. З'явиться вікно для створення вимоги, в якому у полі *Name* введіть – <назва альтернативного потоку>, а потім натисніть кнопку *OK* (рис. 45).

Якщо є інші альтернативні потоки, аналогічно створіть для них вимоги.

4.4. Додаткові умови.

У документі виділіть текст з назвою і описом додаткової умову. Клацніть на ньому правою кнопкою і виберіть в меню *New Requirement*. З'явиться вікно для створення вимоги, в якому у полі *Name* введіть – <назва додаткової умови>, а потім натисніть кнопку *OK* (рис. 46).

Якщо є інші додаткові умови, аналогічно створіть для них вимоги.

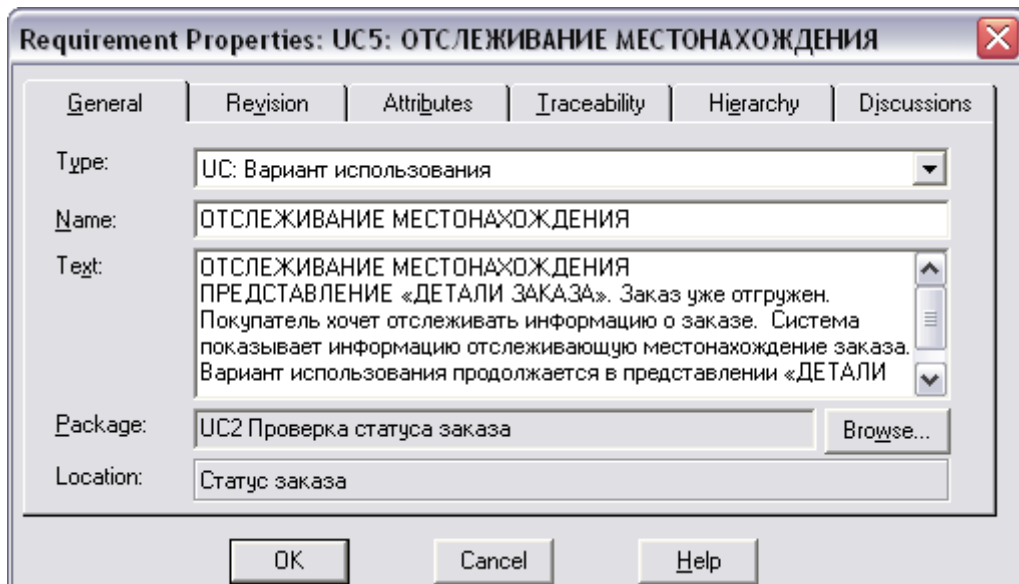


Рис. 45. Вікно для вимоги з назвою альтернативного потоку

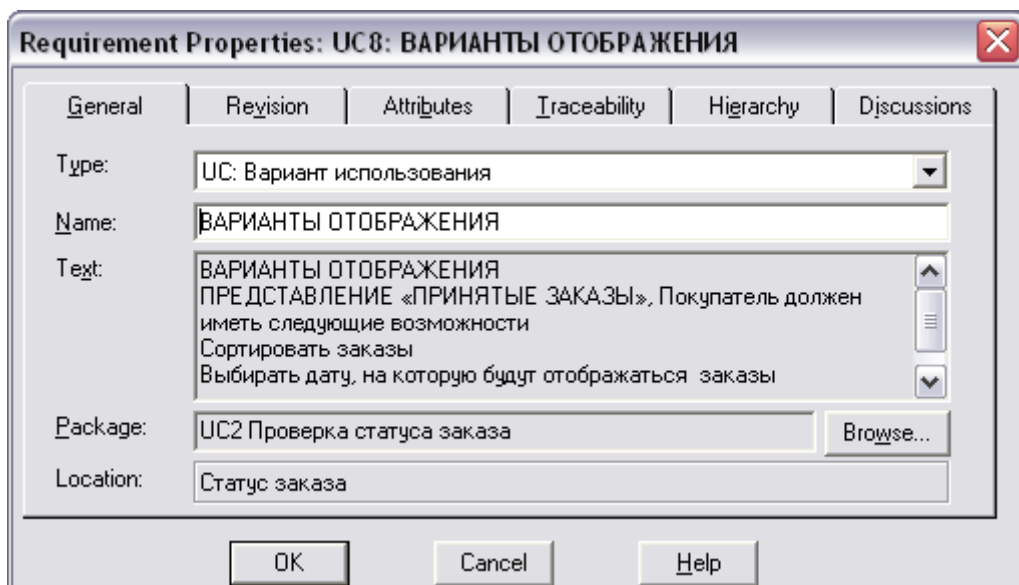


Рис. 46. Вікно для вимоги з назвою додаткової умови

4.5. Передумови, постумови.

У документі виділіть текст з описом передумов. Клацніть на ньому правою кнопкою і виберіть в меню New Requirement. З'явиться вікно для створення вимоги, в якому потрібно (рис. 47):

у полі Type із списку типів вимог вибрати тип UC: *Варіант використання*;

у полі Name ввести – *Передумови*, а потім натиснути кнопку ОК.

Аналогічно створіть вимогу для постумови, якщо така є в документі.

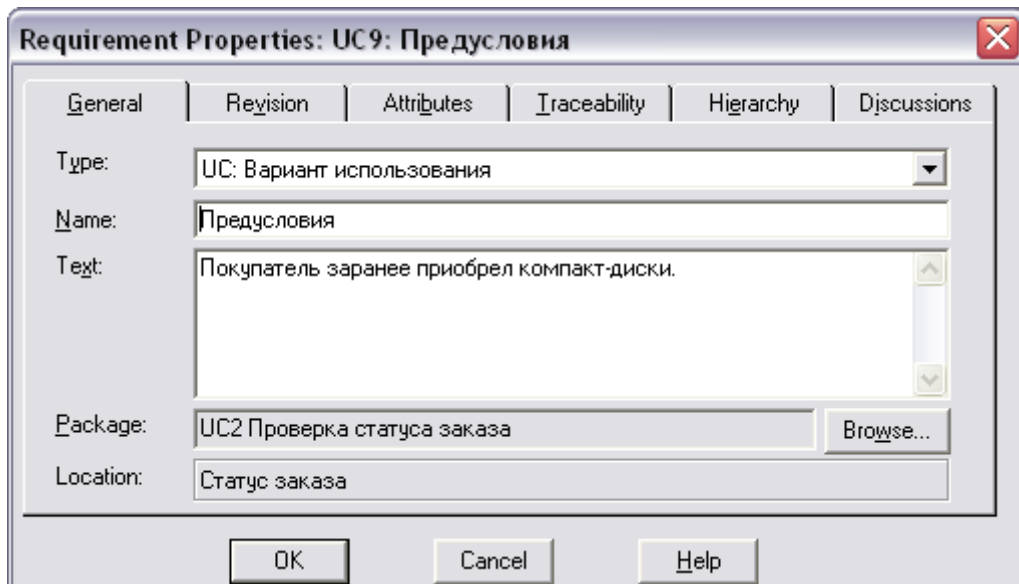


Рис. 47. Вікно для вимоги Передумова

5. Після створення всіх вимог виберіть в меню пункти RequisitePro→Document→Save. Всі створені вимоги фіксуються в базі даних.

6. Закрийте документ *Спецификация вариантов использования*, вибравши пункти меню RequisitePro→Document→Close.

8. Перейдіть у вікно RequisitePro. Створені вимоги відіб'ються у вікні проекту в підпапці для даного варіанта використання, що входить в папку *Спецификация вариантов использования* (рис. 48).

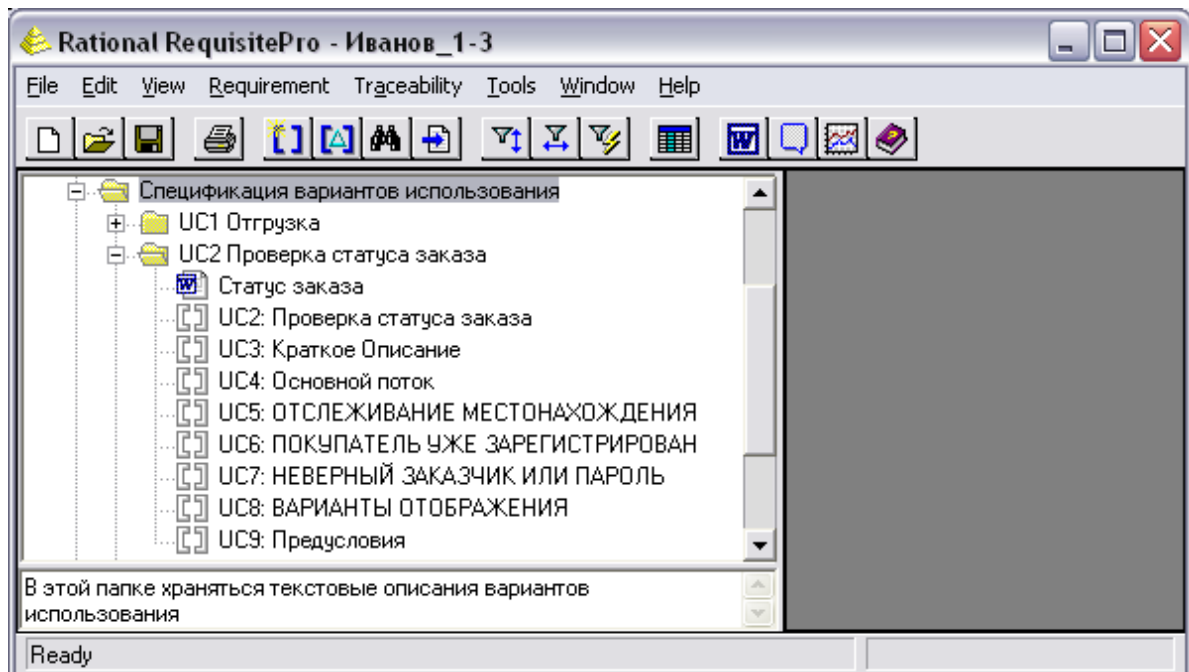


Рис. 48. Вікно проекту зі створеними вимогами типу варіант використання

5. Створіть залежність "батько-нащадок" між вимогами: вимогою-батьком буде вимога з назвою варіанта використання, а вимогами-нащадками усі інші вимоги. Для цього в вікні проекту необхідно виділити першу вимогу-нащадок, у контекстному меню вибрати Change Parent. У вікні Select New Parent з випадного списку вибрати <choose parent> (рис. 49).

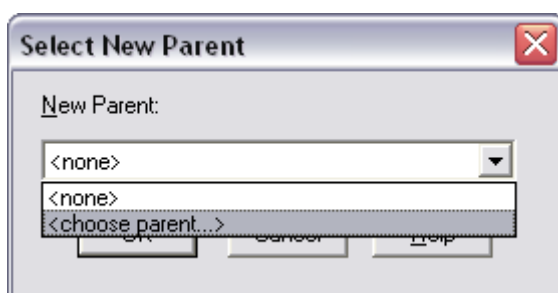


Рис. 49. Вікно Select New Parent

6. З'явиться вікно, в якому буде запропонований список всіх інших вимог у документі. У списку вибираємо вимогу з назвою варіанта використання та натискаємо кнопку ОК (рис. 50). Після цього вікно Select New Parent в якому буде вибрана вимога з назвою варіанта використання (рис. 51). Знову натисніть кнопку ОК. Відкриється вікно Word з документом *Специфікація варіанта використання*, в якому текст модифікованої вимоги буде виділений, а номер вимоги буде змінений.

8. Аналогічно створіть залежність "батько-нащадок" для всіх інших вимог варіанта використання, скориставшись вікном Select New Parent. У випадному списку вже буде запропонована вимога-батько, яку ви обрали на попередньому кроці (рис. 52). Виберіть її, та натисніть кнопку ОК.

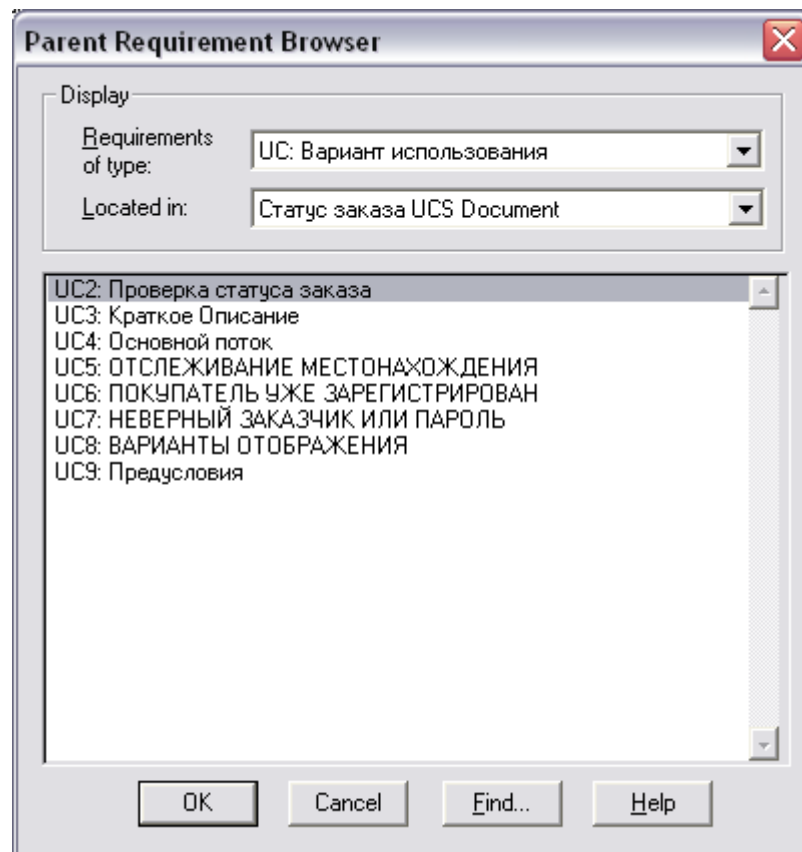


Рис. 50. Вікно Parent Requirement Browser

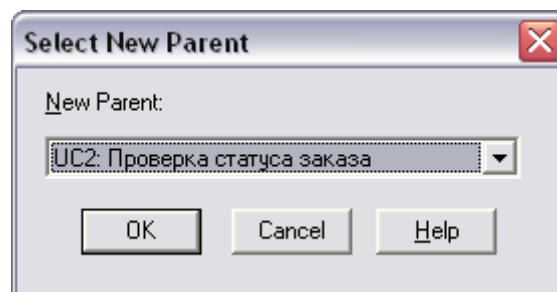


Рис. 51. Вікно Select New Parent із здійсненим вибором



Рис. 52. Вікно Select New Parent

9. Виберіть в меню Word пункти RequisitePro→Document→Save. З'явиться вікно Change description в якому, якщо буде потрібно вводять опис виконаних змін з вимогами. За необхідності цей опис можна переглянути в вікні властивостей для вимоги Properties на вкладці Revision. Включіть прапорець Apply to all modified requirements in document для того, щоб відразу модифікувати всі вимоги в документі (рис. 53). Всі створені зв'язки між вимогами фіксуються в базі даних (рис. 54).

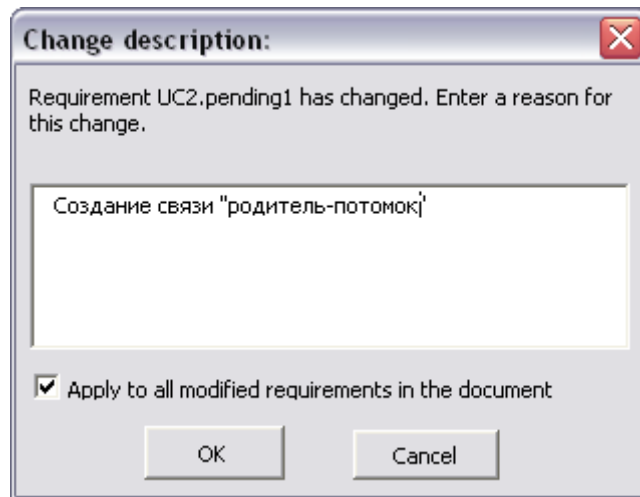


Рис. 53. Вікно Change description

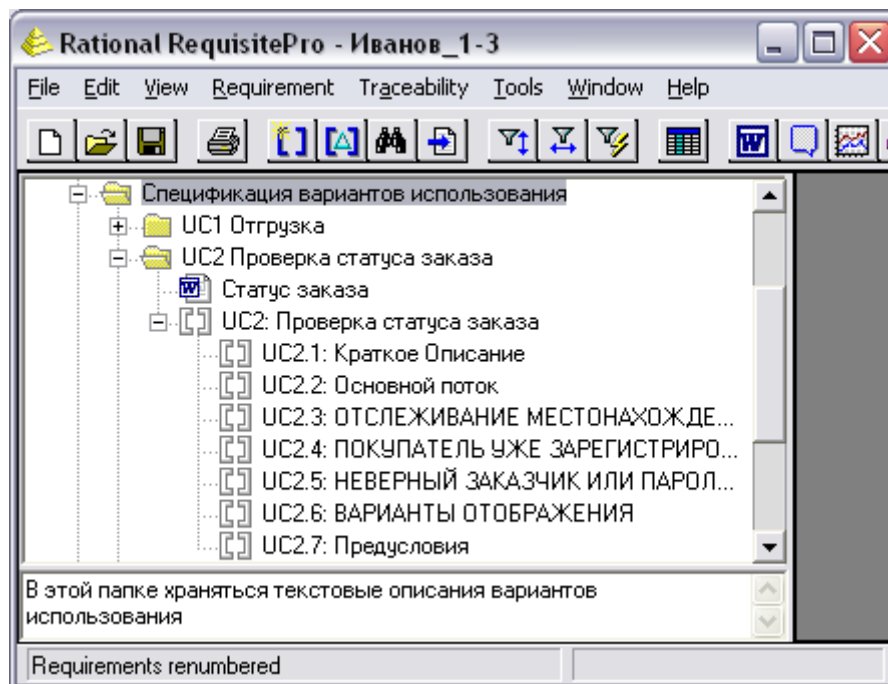


Рис. 54. Вікно проекту зі створеними зв'язками між вимогами типу варіант використання

10. Закрийте документ *Специфікація варіанта використання*, вибравши пункти меню RequisitePro→Document→Close.

11. Аналогічно створіть вимоги в інших специфікаціях варіантів використання.

Завдання 7.3. Створення представлення Матриця вимог з атрибутами для варіанту використання

У завданні 7.3 потрібно виконати наступне:

1. Створити в папці <Назва варіанта використання> представлення для типу вимоги *Варіант використання* (UC).

2. Ввести значення атрибутів вимог у матриці атрибутів.

Порядок виконання

1. Клацніть правою клавішею миші по папці < Назва варіанта використання> і в меню, що з'явилося, виберіть пункт New→View. Відкриється вікно View Properties. У цьому вікні потрібно заповнити наступні поля:

у полі Name введіть ім'я представлення – <Назва варіанта використання> ;

у полі Description введіть опис призначення представлення;

у полі View Type натисніть "Стрілку вниз" і виберіть тип області представлення Attribute Matrix.

у полі Row Requirement Type натисніть "Стрілку вниз" і виберіть тип вимоги UC: *Варіант використання* .

натисніть кнопку ОК.

2. На екрані відобразиться матриця представлення <Назва варіанта використання>. Перегляньте значення атрибутів вимог. Заповніть значення всіх атрибутів для кожної вимоги.

3. Аналогічно створіть представлення для кожного варіанта використання.

Зміст звіту з лабораторної роботи № 7

1. Мета роботи.

2. Створення документа *Специфікація варіанта використання*.

Привести для кожного документа *Специфікація варіанта використання* вікно Document Properties, вікно проекту, в якому показані

всі вимоги типу варіант використання, роздруківку самих документів з виділеними вимогами з текстового редактора Word.

3 Створення представлень *Матриця вимог з атрибутами* для варіантів використання.

Привести для кожного з варіантів використання роздруківку вікна "View Properties" для представлення та роздруківку самих представлень.

4. Висновки.

Контрольні питання

1. Яке призначення документа *Специфікація варіантів використання*?

2. Дайте визначення поняттю актор, хто може бути актором?

3. Що представляє собою варіант використання?

4. Яким чином виділяють з функціонала системи варіанти використання?

5. Які розділи входять в документ *Специфікація варіантів використання*?

6. Який зміст розділів документа *Специфікація варіантів використання*?

7. Що таке тригер, гарантія успіху?

8. Охарактеризуйте передумови і постумови.

Лабораторна робота № 8

Специфікація додаткових вимог, створення документа

Додаткова специфікація вимог у RequisitePro

Цілі лабораторної роботи:

1. Виявлення і опис додаткових вимог до системи.

2. Створення документа *Додаткова специфікація* вимог на основі шаблону документа.

Методичні вказівки до виконання

Завдання 8.1. Створення документа *Специфікація варіанта використання*

У завданні 8.1 потрібно виконати наступне:

1. Створити документ *Додаткова специфікація вимог* на основі шаблону даного документа.
2. Заповнити документ інформацією відповідно до шаблону даного типу документа.

Додаткові специфікації – важливе доповнення до моделі варіантів використання, оскільки разом вони фіксують всі необхідні вимоги до програмного забезпечення (функціональні і нефункціональні), які складають цілковиті технічні вимоги до системи. Закінчені визначення *Специфікації варіантів використання і Додаткові специфікації для конкретної можливості* (FEAT) або деякої частини системи можуть об'єднуватися в документі *Специфікації вимог до програмного забезпечення* (Software Requirements Specification – SRS).

Функціональні вимоги визначають дії, які має бути здатна виконати система, без розгляду фізичних зв'язків і визначають зовнішню поведінку системи типу "введення – виведення". Краще за все вони описуються варіантами використання.

Багато вимог не є функціональними. Нефункціональні вимоги описують тільки атрибути системи або атрибути середовища.

RUP рекомендує концентрувати нефункціональні вимоги в документі, що описує варіант використання у всіх випадках, коли це можливо. У випадку якщо нефункціональні вимоги мають спільний характер і не можуть бути прив'язані до конкретного варіанта використання – вони виносяться в документ *Додаткова специфікація вимог*. Додаткові специфікації фіксують системні вимоги і включають:

- юридичні і нормативні вимоги і вживані стандарти;
- атрибути якості розроблюваної системи, включаючи вимоги застосовності, надійності, ефективності і придатності до експлуатації;
- інші вимоги, типу операційної системи і середовища, сумісності і проектних обмежень.

Документ *Додаткова специфікація вимог* включає наступні розділи:

вступ;

функціональні вимоги;

застосовність;

надійність;
робочі характеристики;
експлуатаційна придатність;
проектні обмеження;
вимоги до призначеної для користувача документації і до системи допомоги;
покупні компоненти;
інтерфейси;
вимоги до ліцензування;
зауваження щодо питань, пов'язаних з авторськими правами;
вживані стандарти.

У *Вступі* міститься короткий огляд всього документа. Він повинний включати мету, контекст, визначення, акроніми, скорочення, посилання і короткий огляд самого документа Додаткова специфікація.

У розділі *Функціональні вимоги* описують функціональні вимоги до системи для тих вимог, які виражені на природній мові (звичайний текст). Коли для фіксації функціональних можливостей використовуються інструментальні засоби розробки (інструментальні засоби вимог, інструментальні засоби моделювання і так далі), цьому розділу документа мають бути доступні зв'язані дані. Відповідно, має бути вказане розташування і назва інструменту, який використовується для фіксації даних.

У цьому розділі може бути вказане посилання на документ, в якому описані функціональні вимоги, наприклад на *Бачення*.

У розділі *Застосовність* включають всі вимоги, які зачіпають застосовність, наприклад:

час, необхідний для навчання звичайних і просунутих користувачів;
вимірний час відгуку для типових завдань;
основні вимоги застосовності нової системи відносно до інших систем, які знають користувачі;

вимоги щодо відповідності загальним стандартам застосовності, наприклад, стандартам інтерфейсу користувача IBM або стандартам графічного інтерфейсу користувача Microsoft для Windows і так далі

У розділі *Надійність* визначаються вимоги до надійності системи, такі як:

доступність – визначає % доступного часу (xx.xx %), час використання, час, що витрачається на обслуговування, порушення режиму роботи і т. д.;

середній час безвідмовної роботи – зазвичай визначається в годинах, але може також визначатися в термінах днів, місяців або років;

середнє напрацювання до ремонту – як довго системі дозволяють працювати до того, коли має бути проведене її обслуговування;

точність – визначає розрядність (роздільну здатність) і точність (за деяким відомим стандартом), які потрібні у вихідних даних системи;

максимальна норма помилок або дефектів – зазвичай виражається в термінах кількості помилок на тисячу рядків коду або кількості помилок у функціональній одиниці.

У розділі *Робочі характеристики* мають бути виділені характеристики продуктивності системи, а саме конкретні характеристики швидкодії.

швидкодія для транзакції (середнє значення, максимальне);

продуктивність (наприклад, кількість транзакцій за секунду);

місткість (наприклад, кількість замовників або транзакцій, яке може розміщувати система);

режими зниженої продуктивності (що є допустимим режимом роботи, коли система погіршилась деяким чином);

використання ресурсів: пам'яті, дискового простору, комунікацій і так далі

За можливості в цьому розділі роблять посилання на ім'я, пов'язаного з цією характеристикою варіанта використання.

У розділі *Експлуатаційна придатність* вказуються всі вимоги, які розширюють експлуатаційну придатність або надійність формованої системи, включаючи стандарти кодування, угоди про імена, бібліотеки класів і утиліти підтримки.

Розділ *Проектні обмеження* повинен містити всі проектні обмеження до формованої системи. Проектні обмеження представляють рішення, які були сформульовані як обов'язкові і повинні твердо витримуватися. Прикладами можуть бути мови програмування, вимоги до технології програмування, обов'язкове використання інструментальних засобів розробки, архітектурні і конструктивні обмеження покупних компонентів, бібліотек класів і так далі.

Розділ *Вимоги до призначеної для користувача документації і до системи допомоги* описує вимоги до інтерактивної документації користувача, до системи довідки, до попереджувальних повідомлень і так далі.

У розділі *Покупні компоненти* описуються всі покупні компоненти, які потрібно використовувати з системою, всі вживані ліцензії або обмеження з використання і всі відомості про сумісність і здібність до взаємодії або про стандарти інтерфейсу.

Розділ *Інтерфейси* визначає інтерфейси, які мають бути підтримані застосуванням. Він повинен містити адекватну специфіку, протоколи, порти і логічні адреси і т. д. так, щоб програмне забезпечення могло бути розроблене і перевірене на відповідність вимогам інтерфейсів. У розділі описуються наступні види інтерфейсів:

призначені для користувача інтерфейси, які мають бути реалізовані програмним забезпеченням;

апаратні інтерфейси, які мають бути підтримані програмним забезпеченням, включаючи логічну структуру, фізичні адреси, очікувану поведінку і т. д.;

програмні інтерфейси з іншими компонентами програмної системи. Це можуть бути покупні компоненти, багато разів використовувані компоненти з іншої прикладної програми або компоненти, що розробляються для підсистем поза контекстом цих специфікацій вимог до програмного забезпечення, але з яким ця прикладна програма повинна взаємодіяти;

комунікаційні інтерфейси до інших систем або пристроїв типу локальних мереж, віддалених послідовних пристроїв і так далі.

У розділі *Вимоги до ліцензування* визначаються всі вимоги обов'язкового ліцензування або інші вимоги обмеження використання, які повинні виконуватися програмним забезпеченням.

Розділ *Зауваження щодо питань, пов'язаних з авторськими правами* описує всі необхідні юридичні застереження, гарантії, оголошення про авторське право, право спадкоємства, торгові марки або емблеми для програмного забезпечення.

Розділ *Вживані стандарти* містить посилання на всі вживані стандарти і на конкретні розділи таких стандартів, які відносяться до описуваної системи. Наприклад, це можуть бути правові і регулюючі стандарти, стандарти якості, промислові стандарти щодо застосовності, здатності до взаємодії, інтернаціоналізації, відповідності операційній системі і т. д.

Порядок виконання

1. Створіть папку для *Додаткової специфікації вимог*. Для цього клацніть правою клавішею по зображенню проекту у вигляді "пірамідки". У меню, що з'явилося, виберіть пункт меню New→Package. У вікні Package Properties заповніть поля:

у полі Name введіть ім'я папки – *Додаткова специфікація вимог*;
у полі Description введіть опис призначення папки.

2. Клацніть правою кнопкою миші по папці *Додаткова специфікація вимог*. У меню, що з'явилося, виберіть пункт меню New→Document. У вікні Document Properties заповніть поля:

у полі Name введіть ім'я документа – *Додаткова специфікація вимог*;
у полі Description введіть опис призначення документа;

у полі File Name введіть ім'я файлу, в якому зберігатиметься документ у RequisitePro;

включіть прапор Show Tags;

задайте директорію для зберігання документа в полі Directory;

у полі Document Type натисніть "Стрілку вниз", і виберіть тип документа *Додаткова специфікація*;

натисніть кнопку ОК.

3. У Word відкриється створений новий документ *Додаткова специфікація вимог*, який містить шаблон цього типу документа. Перегляньте документ у Word.

4. Заповніть розділ *Вступ* документа *Додаткова специфікація вимог* необхідною інформацією відповідно до шаблону. У розділі *Функціональні вимоги* вкажіть посилання на документ *Бачення*.

5. Виберіть необхідні для вашої системи розділи документа, що відбивають нефункціональні вимоги. Непотрібні розділи видаліть з шаблону документа. Введіть в розділах короткий опис додаткових вимог (SUPP).

5. Збережіть документ у форматі Word. Для цього на панелі інструментів у Word виберіть пункти RequisitePro→Document→Save As і введіть ім'я файлу.

Завдання 8.2. Створення вимог у документі *Додаткова специфікація вимог*

У завданні 8.2 потрібно виконати наступне:

1. Створити вимоги типу *Додаткова вимога* (SUPP) у документі *Додаткова специфікація вимог*.

2. Зберегти вимоги в документі і БД.

Порядок виконання

1. Активізуйте з RequisitePro документ *Додаткова специфікація вимог* двічі клацнувши по його іконці у вікні провідника проекту. Документ відкриється у вікні Word.

2. Перейдіть до розділу документа, де приведені описи додаткових вимог до продукту. Виділіть текст, який містить опис додаткової вимоги. Далі в меню виберіть пункти RequisitePro→Requirement→New. З'явиться вікно для створення вимоги. У вікні, що з'явилося, потрібно:

у полі Type із списку типів вимог вибрати тип SUPP: *Додаткова вимога*;

у полі Name ввести назву додаткової вимоги, а потім натиснути кнопку ОК.

6. Аналогічно створіть всі інші додаткові вимоги в документі.

7. Виберіть в меню пункти RequisitePro→Document→Save. Всі створені вимоги фіксуються в базі даних.

8. Закрийте документ *Додаткова специфікація вимог*, вибравши пункти меню RequisitePro→Document→Close.

8. Перейдіть у вікно RequisitePro. Створені вимоги відіб'ються у вікні проекту в папці *Додаткова специфікація вимог*.

Завдання 8.3. Створення представлення *Матриця вимог з атрибутами для додаткових вимог*

У завданні 8.3 потрібно виконати наступне:

1. Створити в папці *Додаткова специфікація вимог* представлення *Додаткові вимоги* для типу вимоги *Додаткова вимога* (SUPP)

2. Ввести значення атрибутів вимог у матриці атрибутів представлення *Додаткові вимоги*.

Порядок виконання

1. Клацніть правою клавішею миші по папці *Додаткова специфікація вимог* і в меню, що з'явилося, виберіть пункт New→View. Відкриється вікно View Properties, в якому потрібно заповнити наступні поля:

у полі Name введіть ім'я представлення – *Додаткові вимоги*;

у полі Description введіть опис призначення представлення;

у полі View Type натисніть "Стрілку вниз" і виберіть тип області представлення Attribute Matrix.

у полі Row Requirement Type натисніть "Стрілку вниз" і виберіть тип вимоги SUPP: *Додаткова вимога*.

натисніть кнопку ОК.

2. На екрані відобразиться матриця представлення *Додаткові вимоги*. Проглянете значення атрибутів вимог. Заповніть значення всіх атрибутів для кожної вимоги.

3. Роздрукуйте представлення *Додаткові вимоги*, виконавши команду File→Print.

Зміст звіту з лабораторної роботи № 8

1. Мета роботи.

2. Створення документа *Додаткова специфікація вимог*.

Привести вікно Document Properties для документа *Додаткова специфікація вимог*, роздруківку для документа *Додаткова специфікація вимог* з виділеними вимогами, яка була виконана з текстового редактора Word.

3 Створення представлення *Матриця вимог з атрибутами* для додаткових вимог.

Привести роздруківку вікна View Properties для представлення *Додаткові вимоги*, роздруківку представлення *Додаткові вимоги*.

4. Висновки.

Контрольні питання

1. Що таке нефункціональні вимоги до системи?

2. У якому випадку нефункціональні вимоги описують в додатковій специфікації вимог, а коли в специфікації варіантів використання?

3. Які типи нефункціональних вимог описуються в документі *Додаткова специфікація вимог*?

4. Які розділи входять в документ *Додаткова специфікація вимог*?

5. Охарактеризуйте зміст системних вимог, які описані в кожному розділі документа *Додаткова специфікація вимог*?

Рекомендована література

1. Вигерс К. Разработка требований к программному обеспечению / Пер. англ. – М.: Изд.-торговый дом "Русская редакция", 2004. – 576 с.
2. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації: Навч. посібн. Ч. 1. – Харків: ХНЕУ, 2007. – 212 с.
3. Ушакова І. О. Основи системного аналізу об'єктів і процесів комп'ютеризації: Навч. посібн. Ч. 2. – Харків: ХНЕУ, 2008. – 312 с.
4. Новичков А. Н. Роль процесса Управления Требованиями при разработке сложных программных систем. Практика применения методологии IBM RUP и инструмента IBM Rational RequisitePro // <http://cmcons.com/>

ДОДАТОК А

ПЛАН УПРАВЛІННЯ ВИМОГАМИ

Версія 1.1

ЛИСТ ЗМІН

Дата	Версія	Опис	Автор
	1.0	Початкова версія	
	1.1	Відредагована версія	

Зміст

1. Вступ	80
1.1. Мета	80
1.2. Межі	80
1.3. Терміни, скорочення і визначення	80
1.4. Посилання	80
1.5. Короткий огляд	80
2. Управління вимогами	81
2.1. Організація, відповідальність і інтерфейси	81
2.2. Таблиця контактів	82
3. Артефакти вимог	83
3.1. Опис артефакту	83
3.1.1. Типи документа	83
3.1.2. Типи вимоги	83
3.1.3. Атрибути	84
3.1.4. Значення атрибутів	87
3.2. Відстеження	89
3.3. Звіти і представлення	89
4. Управління змінами вимог	91
4.1. Обробка і схвалення запитів на зміну	91

ПЛАН УПРАВЛІННЯ ВИМОГАМИ

1. Вступ

1.1. Мета

Мета цього плану – встановити вимоги для проекту <Назва проекту>, які документуватимуться і відстежуватимуться.

1.2. Межі

План управління вимогами надає керівництво для управління вимогами проекту <Назва проекту>.

1.3. Терміни, скорочення і визначення

Словник термінів проекту знаходиться в документі *Глосарій* проекту.

1.4. Посилання

Приклад плану управління вимогами (RM Plan) в Rational Developer Network // www.rational.net

<Інші посилання на документи і літературні джерела>.

1.5. Короткий огляд

У цьому документі описана специфіка проекту <Назва проекту>, а також стратегія управління його вимогами.

Власне з плану починається робота з управління вимогами. План управління вимогами описує документацію вимог, типи вимог і атрибути вимог, які їй відповідають, визначає інформацію і механізми управління, які будуть використані для вимірювання, складання звітів і управління зміною вимог до продукту.

Документ деталізує, як вимоги організовані і як ними управляють в рамках проекту. Тут також описується, як вимоги відстежуватимуться на основі атрибутів і зв'язків з іншими вимогами. Крім того, в документі описується процес управління змінами, що виконуються в цьому проекті.

У плані визначаються:

документи, які мають бути створені;

стандарти, дотримання яких може гарантувати виконання і можливість оцінки розроблюваних вимог.

2. Управління вимогами

2.1. Організація, відповідальність та інтерфейси

Перелік ролей і відповідальностей представлена в таблиці. 2.1

Таблиця 2.1

Ролі і відповідальності

Роль	Відповідальність
Замовник (Customer)	реально існуючий (у організації, якою підпорядкована команда, або поза нею) ініціатор розробки або хто-небудь інший, уповноважений приймати результати (як поточні, так і остаточні) розробки
Менеджер проекту (Project Manager)	відповідає за розвиток проекту в цілому, гарантує, що розподіл завдань і ресурсів дозволяє виконати проект, що роботи і пред'явлення результатів йдуть за графіком, що результати відповідають вимогам. У рамках цих функцій менеджер проекту взаємодіє із замовником і планувальником ресурсів
Гарантія якості (quality assurance)	є відповідальністю менеджера проекту і гарантує, що проектні рішення виконані правильно, відповідно до вимог, верифіковані і повністю завершені
Розробник (Developer)	реалізує проєктовані компоненти, володіє і створює специфічні класи і методи, здійснює кодування і автономне тестування, будує продукт. Це широке поняття, яке може підрозділятися на спеціальні ролі (наприклад, розробник класів). Залежно від складності проєкту команда може включати різну кількість розробників
Керівник команди (Team Leader)	проводить технічне керівництво командою в процесі виконання проєкту. Для великих проєктів можливе залучення декількох керівників підкоманд, які відповідають за вирішення приватних завдань
Архітектор (Architect)	відповідає за проєктування архітектури системи, погоджує розвиток робіт, пов'язаних з проєктом
Специфікатор вимог (Requirements specifier)	Специфікатор вимог деталізує функціональні специфікації системи, описуючи види вимог або варіанти використання, а також додаткові специфікації, що підтримують програмне забезпечення. Специфікатор вимог також відповідає за пакет варіантів використання і підтримує цілісність цього пакету. Відповідальність бізнес-аналітика.

Продовження додатка А
Закінчення табл. 2.1

1	2
Тестер (Tester)	перевіряє функціональність, якість і ефективність продукту. Будує і виконує тести для кожної фази розвитку проекту
Менеджер контролю змін (Change Control Manager)	спостерігає за процесом контролю зміни. Цю роль зазвичай грають представники від всіх зацікавлених сторін, у тому числі замовник, розробники і користувачі. У невеликому проекті цю роль грає зазвичай один член команди, наприклад проект-ний менеджер або архітектор програмного забезпечення

2.2. Таблиця контактів

Відомості для контактів з ролевими кластерами представлені в табл. 2.2

Таблиця 2.2

Відомості для контактів

Роль	ПІБ	Посада	Організація	Контакт
Замовник (для бета-випробування)		Технічний директор		
Зацікавлена особа (Stakeholder)				
Менеджер проекту (project manager)		Менеджер програмного проекту	[розробник]	
Гарантія якості (quality assurance)		Старший тест-менеджер	[розробник]	
Керівник команди (Team Leader)		Старший розробник	[розробник]	
Специфікатор вимог (Requirements specifier)		Старший менеджер проекту	[розробник]	
Адміністратор (administrator)		ІТ директор	[розробник]	
Архітектор (Architect)		Старший програмний інженер	[розробник]	
Менеджер контролю змін (Change Control Manager)		Старший менеджер проекту	[розробник]	

3. Артефакти вимог

3.1. Опис артефакту

3.1.1. Типи документа

Опис типів документів представлений у табл. 3.1

Таблиця 3.1

Опис типів документів

Тип документа	Опис	Тип вимоги за умовчанням
Бачення (Концепція проекту) (Vision (VIS))	Цей документ комбінує елементи первинної ділової пропозиції, бізнес-план і специфікації для особливостей програмного продукту, які розробляються	Властивість системи (Product Feature (FEAT))
Специфікація варіанту використання (Use-Case Specification (UCS))	Текстові описи варіанта використання, як наприклад потік подій, перед- і постумови, додаткові умови для варіанта використання	Варіант Використання (Use Case (UC))
Глосарій (Glossary (GLS))	Використовується для опису загальної специфіки основних термінів проекту	Термін глосарію (Glossary Term (TERM))
Додаткова специфікація вимог (Supplementary Requirements Specification (SUP))	Цей тип документа описує загальносистемні нефункціональні вимоги	Додаткова вимога (Supplementary Requirement (SUPP))
План управління вимогами (Requirements Management Plan (RMP))	Цей тип документа описує типи вимог, типи документів, атрибути вимог, а також стратегії відстеження вимог проекту	Властивості системи (Product Feature (FEAT)) як тип вимоги-замінника (оскільки жодних вимог не буде внесено до RMP)

3.1.2. Типи вимог

Опис типів вимог представлений у табл. 3.2

Таблиця 3.2

Опис типів вимог

Тип Вимоги	Опис	Атрибути
Властивість системи (FEAT)	Зовні помітна функція, забезпечена системою	Тип, Пріоритет, Статус, Трудність, Стабільність, Контакт, Походження
Варіант використання (UC)	Опис поведінки системи в термінах послідовностей дій. У варіантах використання необхідно приділити значну увагу акторові (дійовій особі).	Властивість, Пріоритет, Архітектура ефектів, Контакт, Статус, Трудність
Термін глосарію (TERM)	Терміни, використовувані в спільному словнику наочної області	Неоднозначність
Додаткова вимога (SUPP)	Опис нефункціональних вимог.	Пріоритет, Статус, Трудність, Контакт, Стабільність

3.1.3. Атрибути

Опис атрибутів типів вимог представлений в табл. 3.3

Таблиця 3.3

Опис атрибутів типів вимог

Атрибут	Опис	Тип	Значення списку	Тип вимоги
Тип (Type)	Заповнюється аналітиком. Показує, чи впливає вимога на функціональність системи. Використовується для того, щоб показати, чи є вимога функціональною або нефункціональною	Список	Функціональне Нефункціональне	FEAT

Продовження додатка А
Продовження таблиці 3.3

1	2	3	4	5
Пріоритет (Priority)	Заповнюється аналітиком. Для вимог вводиться і відстежується показник важливості вимог і його шкала. Вводиться при розробці вимог і підтримується впродовж всього проекту. Показує пріоритет реалізації вимоги для клієнта. Використовується при управлінні проектом і визначає пріоритет розробки	Список	Обов'язкове	FEAT, UC, SUPP
			Рекомендоване	
			Опційне	
Статус (Status)	Заповнюється аналітиком для підтвердження статусу вимоги. Представляє статус вимоги. Статус вимоги вказує поточний стан вимоги. Інформація про статус вимоги важлива членам ролевих груп для ефективної організації робіт. Використовується при управлінні проектом і визначає проектний статус (стан) вимоги	Список	Запит на зміну	FEAT, UC, SUPP
			Схвалено	
			Вбудовано в проект	
Трудність (Difficulty)	Заповнюється менеджером проекту. Показує рівень трудовитрат, пов'язаних з реалізацією вимоги. Використовується при управлінні проектом і визначає пріоритет розробки. Трудність виконання вимоги може виражатися у вигляді трудомісткості і вказувати кількість людино-днів, потрібних для його реалізації	Список	Висока	FEAT, UC, SUPP
			Середня	
			Низька	
Стабільність (Stability)	Заповнюється аналітиком і командою розробки. Стабільність вимог відбиває плановану міру або ймовірність їх незмінності в процесі проекту. Для завдання цього параметра використовується кількість планових корекцій вимоги в процесі проекту. Використовується для того, щоб встановити пріоритети розробки і визначити, чи потрібна додаткова інформація.	Список	Висока	FEAT, SUPP
			Середня	
			Низька	

3.1.4. Значення атрибутів

Опис значень атрибутів типів вимог представлений в табл. 3.4.

Таблиця 3.4

Опис значень атрибутів

Значення	Атрибут	Опис
Обов'язкове	Пріоритет	Вимога, критична для успіху (виживання) бізнесу або є безпосереднім замовленням клієнта. Вона має бути виконана і проходитиме перевірку при здачі системи
Рекомендоване	Пріоритет	Вимога доцільна, узгоджена із замовником. Вона покращує характеристики системи, проте її відсутність або неповне задоволення не є підставою для відмови від приймання системи
Опційне	Пріоритет	Вимога можлива, але не обов'язкова для розробки. Вимога бажана з точки зору розробника, доцільність її з боку замовника у нинішній момент не підтверджена
Запит на зміну	Статус	Підтверджується запитом замовника на зміну вимоги
Схвалено	Статус	Схвалено проектним менеджером і гарантією якості (бізнес-аналітиком)
Вбудовано в проект	Статус	Вбудовано в здійснену програму.
Висока	Трудність	Така трудність – це, вірогідність того, що вимога є дуже дорогою в термінах ресурсів або грошей. Вона має бути виконана спочатку або від неї відмовляються
Середня	Трудність	Вимога, яку важко здійснити, але її реалізація ризикована. Вона має бути реалізована, після того, як вимоги високої трудності були виконані або від них відмовилися
Низька	Трудність	Вимога, яку легко здійснити Її реалізують в останню чергу

Продовження додатка А

Закінчення табл. 3.4

1	2	3
Висока	Стабільність	Зміни майже напевно не буде Вимога, яку потрібно задовольнити в першу чергу
Середня	Стабільність	Вимога, можливо, зміниться. Але вона достатньо стійка, щоб почати роботу з її виконання.
Низька	Стабільність	Зміна вимоги ймовірно буде, або вона така невизначена, що потрібне подальше опрацювання перед тим, як роботу над нею можна буде почати
Довідкова інформація	Походження	Інформація, отримана по лініях нашої комерційної або технічної підтримки
Партнери	Походження	Інформація, отримана, від наших партнерів
Конкуренти	Походження	Інформація, що описує особливості головних конкурентів
Відвідувачі клієнта	Походження	Інформація, отримана від відвідувачів і селекторних нарад клієнта
Стислий опис	Властивість	Стислий опис призначення, забезпеченому варіантом використання
Основний потік	Властивість	Основний сценарій для варіанта використання
Альтернативний потік	Властивість	Альтернативні сценарій для варіанта використання
Додаткова умова	Властивість	Нефункціональна вимога, що стосується лише цього варіанта використання
Передумова	Властивість	Необхідна умова (стан) перед цим варіантом використання, щоб він міг початися
Постумова	Властивість	Умови (стани), які повинні з'явитися, коли закінчиться варіант використання
Висока	Неоднозначність	Термін, вживаний лише в даному проекті
Середня	Неоднозначність	Термін, вживаний у даній наочній області
Низька	Неоднозначність	Термін, вживаний у багатьох проектах

3.2. Відстеження

У проекті буде використана стратегія відстеження, представлена на рис. 3.1. Варіанти використання і додаткові вимоги відстежують відносно до властивості системи, яку вони деталізують.

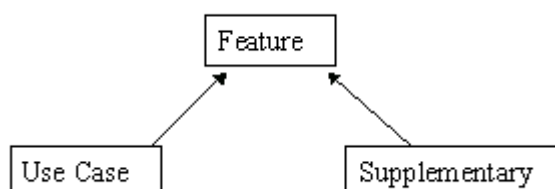


Рис. 3.1. Відстеження вимог

3.3. Звіти і представлення

У середовищі RequisitePro створюються представлення і звіти, характеристика яких наведена в табл. 3.5.

Таблиця 3.5

Опис звітів і представлень

Ім'я представлення	Опис	Ім'я папки	Переваги представлення
Всі властивості системи (All Features)	Відображуються всі вимоги певного типу і їх атрибути у вигляді матриці. Вимоги відбиваються в рядках. Значення атрибутів вимог відбиваються в стовпцях під заголовком з відповідною назвою атрибуту вимоги.	Властивості системи і Бачення (Features and Vision)	Розташування по пріоритетах властивостей системи (Features). У цій матриці можуть створюватися вимоги, які зберігатимуться в БД.
Всі терміни глосарію (All Glossary Terms)	Всі терміни глосарію	Глосарій (Glossary)	Швидкий перегляд термінів глосарію
Всі додаткові специфікації (All Supplementary Requirements)	Відображуються всі додаткові вимоги і їх атрибути у вигляді матриці. Вимоги відбиваються в рядках. Значення атрибутів вимог від	Додаткові специфікації (Supplementary Requirements)	Розташування за пріоритетами нефункціональних вимог У цій матриці мо

Продовження табл. 3.5

1	2	3	4
	биваються в стовпцях під заголовком з відповідною назвою атрибуту вимоги		жуть створюватися вимоги, які зберігатимуться в БД
Всі варіанти використання (All Use Cases)	Відображуються всі варіанти використання і їх атрибути. Вимоги відбиваються в рядках. Значення атрибутів вимог відбиваються в стовпцях під заголовком з відповідною назвою атрибуту вимоги	Варіанти використання (Use Cases)	Розташування по пріоритетах варіантів використання У цій матриці можуть створюватися вимоги, які зберігатимуться в БД
Короткий опис варіантів використання (Use-Cases Brief Description)	Всі варіанти використання і їх короткий опис	Варіанти використання (Use Cases)	Короткий огляд варіантів використання
Властивості системи не пов'язані з додатковими специфікаціями (Features Not Linked to Supplementary Specs)	Список властивостей системи, що не пов'язуються з додатковими специфікаціями	Аналіз покриття (Coverage Analysis)	Відображує всі властивості системи, які не були пов'язані з нефункціональними вимогами
Властивості системи не пов'язані з варіантами використання (Features Not Linked in Use Cases)	Список властивостей системи, що не пов'язуються з варіантами використання	Аналіз покриття (Coverage Analysis)	Відображує всі властивості системи, які не були деталізовані в наборі варіантів використання
Звіт повного покриття (Full Coverage Report)	Дерево з відстеженням всіх зв'язків	Аналіз покриття (Coverage Analysis)	Показує всі зв'язки, які зараз встановлені в проєкті

Закінчення табл. 3.5

1	2	3	4
Покриття функціональних вимог (Functional Requirements Coverage)	Список властивостей системи, пов'язаних з варіантами використання	Аналіз покриття (Coverage Analysis)	Показує зв'язки, які зараз встановлені між функціональними вимогами
Додаткові специфікації, на які впливають зміни властивостей системи (Supplementary Requirements Affected by Feature Changes)	Список додаткових вимог, на які потенційно впливають зміни у властивостях системи	Аналіз впливу (Impact Analysis)	Показує нефункціональні вимоги, на які впливають зміни властивостей системи
Варіанти використання, закріплені за змінами властивостей системи (Use Cases Impacted by Feature Changes)	Список вимог варіантів використання, на які потенційно впливають зміни у властивостях системи	Аналіз впливу (Impact Analysis)	Показує функціональні вимоги, на які впливають зміни властивостей системи

4. Управління змінами вимог

4.1. Обробка і схвалення запитів на зміну

Для відстеження дефектів і поліпшення вимог, використовується продукт Rational ClearQuest, який інтегрується з RequisitePro.

Зміст

Вступ	3
Лабораторна робота № 1. Знайомство з RequisitePro	4
Лабораторна робота № 2. Створення нового проекту	11
Лабораторна робота № 3. Створення типів вимог і атрибутів типів вимог	17
Лабораторна робота № 4. Створення шаблонів документів і типів документів в RequisitePro	24
Лабораторна робота № 5. Створення документів: <i>План управління вимогами, Глосарій</i> у RequisitePro	31
Лабораторна робота № 6. Виявлення високорівневих вимог, створення документа <i>Бачення</i> в RequisitePro	46
Лабораторна робота № 7. Специфікація ключових УС, створення документа <i>Специфікація варіанта використання</i> в RequisitePro	56
Лабораторна робота № 8. Специфікація додаткових вимог, створення документа <i>Додаткова специфікація вимог</i> у RequisitePro	69
Рекомендована література	76
Додаток	77

НАВЧАЛЬНЕ ВИДАННЯ

**Завдання до лабораторних робіт та методичні рекомендації
до їх виконання з навчальної дисципліни
"ОСНОВИ СИСТЕМНОГО АНАЛІЗУ ОБ'ЄКТІВ
ТА ПРОЦЕСІВ КОМП'ЮТЕРИЗАЦІЇ"
(У СЕРЕДОВИЩІ RATIONAL REQUISITPRO)
для студентів напряму підготовки "Комп'ютерні науки"
усіх форм навчання**

**Укладачі: Ушакова Ірина Олексіївна
Плеханова Галина Олегівна**

Відповідальний за випуск Пономаренко В. С.

Редактор Коротчаєва І. О.

Коректор

План 2009 р. Поз. №220.

Підп. до друку Формат 60 × 90 1/16. Папір MultiCopy. Друк Riso.
Ум.-друк. арк. 5,75. Обл.-вид. арк. 7,2. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ, 61001, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк №481 від 13.06.2001 р.*

**Завдання до лабораторних робіт
та методичні рекомендації до їх виконання
з навчальної дисципліни
"ОСНОВИ СИСТЕМНОГО АНАЛІЗУ ОБ'ЄКТІВ
ТА ПРОЦЕСІВ КОМП'ЮТЕРИЗАЦІЇ"
(У СЕРЕДОВИЩІ RATIONAL REQUISITEPRO)
для студентів напряму підготовки "Комп'ютерні науки"
усіх форм навчання**