Лабораторна робота 2
Дослідження процесу розробки програмного забезпечення. Специфікування вимог

Мета – отримати навички специфікування вимог до програмного забезпечення.

Теоретичні відомості

Кожна програмна система протягом свого існування проходить з певною послідовністю фази або стадії від задуму до його втілення в програми, експлуатацію та вилучення. Така послідовність фаз називається *життєвим циклом (Software life cycle). К*ожна фаза містить три складових:

- процеси;
- продукти;
- ресурси.

На кожній фазі відбувається певна сукупність процесів, кожен з яких породжує певний продукт, використовуючи певні ресурси.

Усі продукти (рис.1) всіх процесів являють собою певні описи — тексти вимог до розробки, погодження домовленостей, документацію, тексти програм, інструкції з експлуатації тощо.

До процесу розроблення входять такі дії:

- специфікування вимог;
- проектування;
- кодування й тестування.

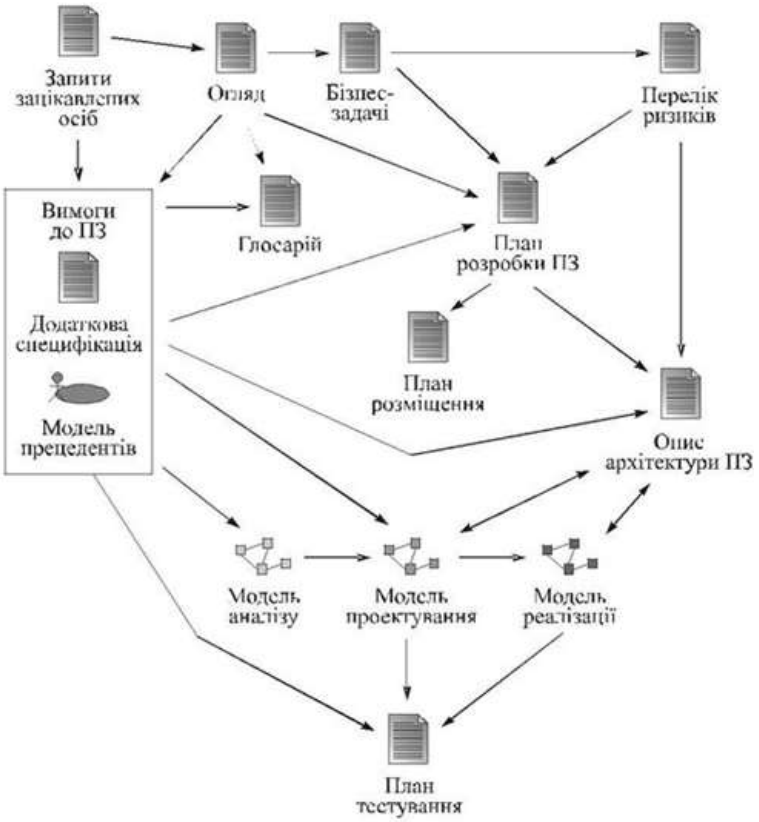

Рис.1 Робочі продукти процесу розробки ПЗ та взаємозвязок між ними

Методичні рекомендації

Варіант використання (use case) - зовнішня специфікація послідовності дій, які система або інша сутність можуть виконувати в процесі взаємодії з акторами, без розгляду внутрішньої структури цієї системи.

Діаграми варіантів використання показують взаємодію між варіантами використання та діючими особами, відображаючи функціональні вимоги до системи з погляду користувача.

Опис предметної області.

Банкомат – це автомат для видачі готівки за кредитними пластиковими картками. До його складу входять такі пристрої: дисплей, панель управління з кнопками, приймач кредитних карток, сховище готівки та лоток для видачі, принтер для друку довідок.

Банкомат зв'язаний з комп'ютером банку, де зберігаються відомості про рахунки клієнтів. Обслуговування клієнта розпочинається з моменту розміщення пластикової картки в банкомат. Після розпізнання типу пластикової картки банкомат видає на дисплей запрошення ввести персональний код – чотиризначне число (ПІН код). Банкомат перевіряє правильність введеного коду і дозволяє клієнту вибрати операцію або зняття готівки, або перевірки залишку рахунка.

Якщо клієнт обирає зняття готівки, він повинен вказати суму (10, 50, 100, 200, 500, 1000 грн.). На дисплеї формується повідомлення про друк довідки, а банкомат посилає запит головному комп'ютерові банку. Якщо дозвіл на операцію отримано, банкомат перевіряє наявність коштів у своєму сховищі. На дисплей виводиться повідомлення «Вийміть картку». Після виймання картки банкомат видає вказану суму готівки до лотка. Банкомат друкує довідку про операцію, якщо вона була затребувана клієнтом. Коли ж клієнт бажає отримати інформацію про залишок на рахунку, він обирає відповідний пункт меню, банкомат посилає запит до головного комп'ютера банку і виводить суму на екран, або друкує довідку за бажанням клієнта.

Глосарій.

Банкомат – термінал, який дає можливість клієнту здійснити транзакцію, використовуючи для ідентифікації свою пластикову картку. Банкомат взаємодіє з клієнтом для отримання необхідної інформації для транзакції та з головним комп'ютером банку, який одержує інформацію, перевіряє та видає дозвіл або заперечує проведення транзакції.

Картка – електронна пластикова картка клієнта. Кожна картка містить код банку, номер картки, персональний код клієнта.

Клієнт – власник одного або декількох рахунків у банку.

Транзакція – одиничний інтегрований запит на виконання деякої послідовності операцій над рахунками одного клієнта.

Рахунок – одиничний банківський рахунок, над яким виконуються транзакції. Клієнт може мати декілька рахунків.

Предметна область може бути представлена у вигляді описів користувача (user story):

Як [роль користувача] я хочу [діяльність], щоб я міг [вигода]

Як [роль користувача] я можу [діяльність], щоб [вигода]

Роль користувача - хто? (новий користувач, гість, шукач роботи)

Діяльність - що? (функціональність, дія системи)

Вигода - чому? (цінність для кінцевого користувача)

Наприклад, «Як користувач я хочу, щоб веб-сторінки завантажувались протягом 2 або 3 секунд, щоб я міг швидко працювати».

Побудова діаграми варіантів використання.

Окремий варіант використання (прецедент) позначається на діаграмі еліпсом, усередині якого міститься його коротке ім'я у формі дієслова (рис. 2) с пояснювальним текстом. Текст імені варіанта використання повинен починатися із великої літери.

Рис. 2. Графічне позначення варіанта використання

Актором (actor) або діючою особою називається будь-який об'єкт, суб'єкт або система, які взаємодіють з бізнес-системою, що моделюється, ззовні. Це може бути людина,

технічний пристрій, програма або будь-яка інша система, що служить джерелом впливу на систему, що моделюється.

Рис. 3. Графічне позначення актора

Відношення (relationship) - семантичний зв'язок між окремими елементами моделі. У мові UML є кілька стандартних видів відносин між акторами й варіантами використання:

– асоціації (association relationship) (рис. 4), яка специфікує семантичні особливості взаємодії акторів і варіантів використання в графічній моделі системи, позначається суцільною лінією між актором і варіантом використання;

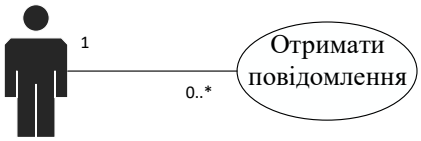

Рис. 4. Графічне позначення відношення «асоціація»

– включення (include relationship) (рис. 5), відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного), позначається пунктирною лінією зі стрілкою, спрямованою до залежного елементу зі стереотипом <<include>>;

Рис. 5. Графічне позначення відношення «включення»

– розширення (extend relationship) (рис.6) позначається як відношення залежності у формі пунктирної лінії зі стрілкою, спрямованої від того варіанта використання, що є розширенням для базового варіанта використання зі стереотипом <<extend>>;

Рис. 6. Графічне позначення відношення «розширення»

– узагальнення (generalization relationship) (рис.7) визначає зв'язок, коли два й більше актори мають загальні властивості позначається суцільною лінією зі стрілкою у формі не зафарбованого трикутника, що вказує на батьківський варіант використання.

Рис. 7. Графічне позначення відношення «узагальнення»

Для побудови діаграми варіантів використання (рис.8) моделі банкомату необхідно виділити акторів і прецеденти.

Актори:

– актор на ім’я Банк;
– актор на ім’я Клієнт банкомату.

Прецеденти:

– отримання довідки про стан рахунку;

- зняття готівки;
- перевірка ПІН коду;
- блокування кредитної картки.

Рис.8. Діаграма варіантів використання банкомату

Кожен прецедент має бути описаний (рис.9) для подальшого використання зі специфікування вимог.

ADM_UC_0.1	Создать учетную запись
Description	Создание учетной записи пользователя Администратором
Actor	Администратор
Pre-Conditions	1. Администратор залогинился в систему 2. У Администратора достаточно прав для создания учетной записи
Main Flow	1. Администратор вызывает функцию 2. Система предлагает ввести информацию про учетную запись 3. Администратор заполняет необходимые данные 4. Система проверяет корректность введенной информации 5. Система создает учетную запись 6. Система уведомляет пользвателя (для кого создавалась учетная запись) про создание учетной записи 7. Система уведомляет Администратора про успешность создания учетной записи
Alternative Flow	1.a Администратор отказывается от выполнения операции 1. Система отображает предупреждение 2. Если дминистратор подтверждает свои действия 2.1 Система заканчивает выпонение основного сценария 3. Если Администратор не подтверждает свои дейсвия 3.1 Система возвращается на Шаг 3 основного сценария 4. а Информация была введена некорректно 4.1 Система уведомляет Администратора про наличие ошибок 4.2 Система возвращается на Шаг 3 основного сценария
Post-Conditions	Учетная запись создана. Пользователь, для которого создавалась запись уведомлен.

Рис.9 Приклад опису прецеденту

Вимога - це докладний опис того, що має бути реалізовано. Опис вимог:
<Id> <система> повинна (shall) <дія>
Функціональні вимоги – яку поведінку повинна підтримувати система;
Нефункціональні вимоги – особлива властивість або обмеження, що накладаються на систему

Вимоги вказують що має бути побудовано, але не говорять як це зробити.

Приклад функціональних вимог:

Ф1. Система АТМ повинна перевіряти дійсність вставленої в банкомат картки.

Ф2. Система АТМ повинна перевіряти достовірність PIN-коду, введеного користувачем.

Ф3. Система АТМ повинна видавати по одній АТМ картці не більше $ 250 на добу.

Приклад нефункціональних вимог:

НФ1. Система АТМ повинна бути написана на С ++.

НФ2. Система АТМ повинна обмінюватися інформацією з банком, використовуючи 256-розрядну кодування.

НФ3. Система АТМ повинна перевіряти дійсність картки АТМ протягом не більше трьох секунд.

НФ4. Система АТМ повинна перевіряти достовірність PIN-коду протягом не більше трьох секунд.

Завдання:

1. Сформувати робочу групу (3-5 чоловік), розподілити ролі.
2. За узгодженням з викладачем обрати варіант завдання (табл..1) для виконання лабораторних робіт.
3. Провести попередній аналіз предметної області, визначити функції ПЗ, що проектується.
4. Побудувати діаграму прецедентів на основі проведеного попереднього аналізу.
5. Специфікувати вимоги.
6. Обрати та обґрунтувати обрану модель життєвого циклу.
7. Підготувати захист у вигляді мітингу із замовником.

Варіанти

Таблиця 1

№	Назва системи	Опис системи
1	Система клімат контролю	дозволяє встановлювати і підтримувати температуру і вологість повітря у кімнаті.
2	Інформаційна система бібліотеки	дозволяє шукати книги у своєму каталозі, враховувати видачу книг на руки й повернення книг.
3	Інформаційна система поліклініки	дозволяє ставити і знімати хворих з обліку, записувати хворих на прийом до лікарів, ураховувати факт прийому.
4	Інформаційна система деканату	дозволяє приймати й відраховувати студентів, вести облік успішності за підсумками сесії, переводити студентів із групи в групу й з курсу на курс
5	Інформаційна система складу	дозволяє враховувати надходження й відхід товарів зі складу, а також визначати місце зберігання товарів на складі.
6	Система обліку робочого часу	дозволяє керівникам видавати завдання й відслідковувати хід їхнього виконання, а виконавцям - вести облік робочого часу, витраченого на виконання кожного завдання.

7	Інформаційна система житлового агентства	дозволяє квартиронаймачам дібрати й зняти житло, а власникам житла - запропонувати й здати житло.
8	Інформаційна система технічної експертизи	дозволяє здобувачам грантів подавати заявки, незалежним експертам оцінювати заявки, а власникам фонду ухвалювати рішення щодо видачі грантів за результатами експертизи заявок.
9	Інформаційна система бронювання авіаквитків	забезпечувати туриста і турагента інформацією про туристичні продукти і послуги, автоматизувати процес бронювання авіаквитків, замовлення додаткових послуг, прямий обмін повідомленнями.
10	Інформаційна система паркування автомобілів	дозволяє визначати наявність вільних місць для паркування за маршрутом і оповістити водія, коли знайдено досить велике місце для паркування
11	Касовий апарат у торговому центрі	має у своєму складі сканер, якій дозволяє відсканувати штрих-код товару, перевірити в базі даних наявність і ціну товару, сформувати чек і залишок товару в базі даних.
12	Інформаційна система ЖЕКу	дозволяє формувати платіжні документи з оплати житлово-комунальних послуг, оплату послуг, контролювати надані послуги.
13	Інформаційна система ДАІ	дозволяє обрати місце навчання, здійснити реєстрацію, оплатити навчання, отримати посвідчення водія на право керування транспортними засобами.

Контрольні питання

1. Покажіть розвиток інженерії і зокрема інженерії програмного забезпечення
2. Наведіть модель Code and fix та вкажіть недоліки
3. Поясніть, в чому системо утворююча роль життєвого циклу програмного забезпечення в інженерії
4. Наведіть Step wise модель, поясніть роль її в інженерії програмного забезпечення та вкажіть недоліки
5. Основні складові фаз життєвого циклу, дайте визначення, наведіть приклади
6. Типи програм за Леманом, загальні визначення
7. Комп’ютерні програми як системи
8. Типи програмних систем
9. Вкажіть користувачів продуктів інженерії програмного забезпечення. Наведіть приклади продуктів інженерії програмного забезпечення-робочих продуктів.
10. Специфікування вимог, поняття, витоки
11. Проектування, поняття, витоки, результат
12. Конструювання, поняття, витоки, результат
13. Супроводження, поняття, витоки, результат
14. Доменний аналіз, поняття
15. Утилізація, поняття, витоки