

Аналіз вимог до програмного забезпечення

Лабораторні роботи

Лабораторна робота №1

Виявлення та характеристика вимог до ПЗ

Теоретичні положення

Вимоги – це властивості, які повинно мати програмне забезпечення для адекватного визначення функцій, умов та обмежень виконання ПЗ, а також об'ємів даних, технічного забезпечення і середовища функціонування.

Методи виявлення вимог:

Інтерв'ю, опитування, анкетування;

Мозковий штурм, семінар;

Спостереження за виробничою діяльністю, «фотографування» робочого дня;

Аналіз нормативної документації;

Аналіз моделей діяльності;

Аналіз конкурентних продуктів;

Аналіз статистики попередніх версій системи.

Основна ціль робочого процесу визначення вимог базується на тому, щоб направити процес розробки на отримання вірної системи.

Можна визначити наступні кроки робочого процесу визначення вимог:

Перелік можливих вимог;

Усвідомлення контексту системи;

Визначення функціональних вимог;

Визначення нефункціональних вимог.

Перелік можливих вимог

Починаємо збирати всі можливі вимоги та ідеї на рахунок майбутньої системи. Кожна пропозиція в переліку повинна мати коротку назву та короткий опис, який може складатися з стану пропозиції (наприклад, запропоновано, затверджено, відібрано та ін.); вартість реалізації; пріоритет

(наприклад, критичний, важливий або допоміжний); рівень ризику, пов'язаний з реалізацією пропозиції (наприклад, критичний, значимий або звичайний). Цей перелік може змінюватися.

Усвідомлення контексту системи

Для того щоб вірно визначити вимоги розробники системи повинні розуміти контекст (частину предметної області), в якій працює система. Існує два підходи до опису контексту системи:

Моделювання предметної області;

Бізнес-моделювання.

Модель предметної області описує важливі поняття області і їх зв'язки між собою. Не можна плутати модель предметної області з логічною або фізичною моделями системи. Модель предметної області описує тільки об'єкти предметної області, але не показує як програма буде з ними працювати. Дана модель також дозволяє скласти глосарій системи для кращого її розуміння користувачами та розробниками.

Бізнес-модель описує процеси (існуючі чи майбутні), які повинна підтримувати система. Бізнес-модель можна представити як підмоделі предметної області. Крім визначення бізнес - об'єктів, які задіяні в проекті, ця модель визначає працівників, їх обов'язки, і дії, які вони повинні виконувати.

Визначення функціональних вимог

Підхід до виявлення системних вимог базується на використанні варіантів використання системи (Use Cases), які обхвачують як функціональні так і нефункціональні вимоги, які специфічні для конкретного варіанта використання.

Для користувача важливо, щоб система виконувала певні дії, при цьому користувач певним чином взаємодіє з системою, використовуючи її для своїх цілей. Таким чином, якщо визначити всі можливі варіанти використання

системи користувачем або іншими зовнішніми процесами, то ми отримаємо функціональні вимоги до неї.

Визначення нефункціональних вимог

До нефункціональних вимог відносяться такі властивості системи, як обмеження середовища і реалізації, продуктивність, залежність від платформи, розширеність, надійність. Під надійністю розуміються такі характеристики, як природність, точність, середнє напрацювання на відмову, число похибок на тисячу рядків програми, число похибок на клас.

Вимоги по продуктивності – це швидкість, пропускна здатність, час відклику, використана пам'ять. Багато вимог, які пов'язані з продуктивністю повинні бути описані в конкретних варіантах використання, а в розділах, які відносяться до всієї системи.

Також можна відмітити, що часто нефункціональні вимоги не можуть бути прив'язані до конкретного варіанту використання і повинні бути винесені в окремий список додаткових вимог до системи.

Завдання

1. Вибрати свій варіант завдання.
2. Визначити учасників проекту.
3. Провести аналіз методів виявлення вимог.
4. Застосувати один із методів виявлення вимог.
5. Виявити вимоги до запропонованої системи.
6. Описати вимоги використовуючи зручний для себе спосіб.

Варіанти завдань

1. Система розрахунку заробітної плати бухгалтера приватного підприємця.
2. Система нарахування стипендії студентам.
3. Система для ведення реєстру співробітників.
4. Система для автоматизації робочого місця діловода.

5. Система для автоматизації робочого місця працівника відділу кадрів для роботи з студентами.
6. Система пропускнуго контролю на підприємстві.
7. Система пропускнуго контролю в 18 корпус.
8. Система для автоматизації роботи старости групи.
9. Система для формування додатків до диплому студентам.
10. Система нарахування заробітної плати механіка СТО.
11. Система підрахунку витрачених коштів в родині.
12. Система автоматизації роботи касира пункту обміну валют.
13. Система автоматизації робочого місця секретаря.
14. Система нарахування заробітної плати водія тролейбуса.
15. Система підрахунку пропусків студентів за навчальний рік.

Лабораторна робота №2

Виявлення вимог: учасники процесу, взаємозв'язки між ними у вигляді UML діаграм

Теоретичні положення

Під час процесу виявлення вимог визначаються учасники процесу та розподіляються ролі.

Типові приклади ролей:

Користувачі (Users) – це група, яка охоплює тих людей, які будуть безпосередньо використовувати програмне забезпечення.

Користувачі можуть описати задачі, які вони вирішують чи будуть вирішувати за допомогою програмної системи, а також сподівання по відношенню до атрибутів якості, які відображаються в користувацьких вимогах.

Замовники (Customers) – це особа або організація, що відповідають за замовлення програмного продукту, а також мають пряме або другорядне відношення до використання цього продукту.

Аналітики (Market analysts) – особи, що відповідають за маркетинг, збір вимог, написання вимог.

Регулятори (Regulators) – особи, що проводять регулювання областей використання відповідно керуючій документації, а також проводять контроль над етапами та процесами формування вимог та розробкою програмного продукту.

Інженери-програмісти (Software Engineer) – особи, що розробляють програмний продукт, а також вони відповідальні за технічну оцінку шляхів вирішення поставлених задач та реалізацію вимог замовників.

Більшу частину користувацьких вимог виявляють на стадії моделювання. Спочатку рекомендується виявити всіх зацікавлених осіб (акторів або actors), потім - визначити їх дії (варіанти використання або

прецеденти або use cases), а при необхідності – описати сценарії цих дій на відповідних діаграмах. Ці варіанти використання, по суті, і будуть являтися вимогами до системи.

Уніфікована мова моделювання UML представляє собою мову для визначення, представлення, проектування і документування програмних систем, організаційно-економічних систем, технічних систем та інших систем.

Як правило, для моделювання поведінки користувачів системи використовується мова моделювання UML, діаграми варіантів використання, діаграми активності, діаграми послідовності і взаємодії, а також діаграми стану.

Найбільш популярні інструменти для UML моделювання:

Microfocus Together

Microsoft Visio

Enterprise Architect

IBM/Rational

Rose/RSA

Power Designer

Umbrello

Завдання

7. Виберіть свій варіант завдання з лабораторної роботи №1.
8. Проаналізуйте виявлені Вами вимоги в лабораторній роботі №1.
9. Виберіть учасників процесу виявлення та складання вимог.
10. Побудуйте діаграму співвідношення між учасниками процесу (використовуючи тип Use Case Definition).
11. Опишіть їх функціональні можливості (обов'язки) засобами діаграми класів (UML Class Diagram).
12. Оформіть лабораторну роботу відповідно вимог до звіту.

Лабораторна робота №3

Збір даних та формулювання вимог

Моделювання процесу виявлення вимог

Теоретичні положення

Інструментальне середовище VPwin та побудова моделі IDEF0

Моделювання процесів, як правило виконується за допомогою case-засобів. До таких засобів відноситься і VPwin.

VPwin підтримує три методології моделювання: функціональне моделювання (IDEF0); опис бізнес-процесів (IDEF3); діаграми потоків даних (DFD). Кожна із методологій вирішує свої специфічні задачі. Також можна побудувати і змішані моделі, тобто модель буде мати діаграми всіх трьох методологій.

Модель в VPwin розглядається як сукупність робіт, кожна з якої працює з деяким набором даних. Робота відображається у вигляді прямокутника, дані – у вигляді стрілок. Якщо клацнути по любому об'єкту моделі лівою кнопкою миші, з'являється контекстне меню, кожний пункт якого відповідає редактору певної властивості об'єкта.

Найбільш зручною мовою моделювання бізнес-процесів являється IDEF0, де система представляється як сукупність взаємодіючих робіт або функцій. Така чисто функціональна орієнтація є чисто принциповою – функції системи аналізуються незалежно від об'єктів, якими вони керують. Це дозволяє провести чітке моделювання логіки і взаємодіючих організацій.

Процес моделювання системи в IDEF0 починається з створення *контекстної діаграми* – діаграми найбільш абстрактного рівня опису системи в цілому, яка має визначення суб'єкта моделювання, цілі і точки зору на модель.

Під суб'єктом розуміється сама система, при цьому необхідно чітко встановити, що входить в систему, а що лежить за її межами, тобто, визначити,

що буде надалі розглядатися як компоненти системи, а що як зовнішні фактори. На визначення суб'єкта системи буде суттєво впливати позиція, з якої розглядається система, і ціль моделювання, тобто питання, на які побудована модель повинна дати відповідь. Отже, спочатку необхідно визначити спосіб моделювання.

Опис області побудови як системи в цілому, так і її компонентів являється основою побудови моделі. Хоча область може корегуватися в процесі моделювання, але загалом її треба сформулювати на початку, оскільки саме область визначає напрямок моделювання. При формулюванні області необхідно враховувати два компонента – широту і глибину. Широта визначає границі моделі – що буде розглядатися в середині системи, а що ззовні. Глибина визначає на якому рівні деталізації модель буде завершеною. При визначенні глибини системи необхідно пам'ятати про обмеження часу – праця ємність побудови моделі росте в геометричній прогресії зі збільшенням глибини декомпозиції. Після визначення границь моделі передбачається, що нові об'єкти не повинні вноситися в модельовану систему.

Ціль моделювання

Ціль моделювання формується з визначення питань: для чого процес повинен бути змодельованим, що повинна показувати модель, що отримає клієнт.

Під *точкою зору* розуміється перспектива, з якою спостерігалась система при побудові моделі.

IDEF0 – модель представляє наявність чітко сформованої цілі, єдиного суб'єкта моделювання і однієї точки зору. Для занесення області, цілі та точки зору в модель IDEF0 в VPwin слід вибрати пункт меню Model/Model Properties викликаючий діалог Model Properties (рис. 1). В закладці Purpose слід внести ціль і точку зору, а в закладку Definition – визначення моделі та опис області.

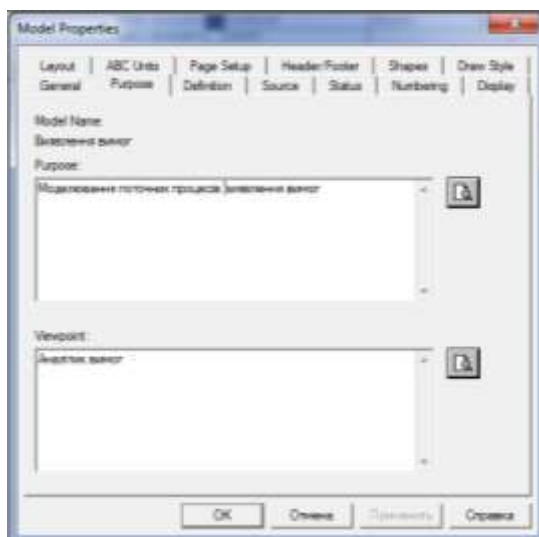


Рис. 1 Діалог внесення властивостей моделі

В закладці Status того ж діалогу можна описати статус моделі (робочий варіант, черновий, завершений та ін.), час створення і останнього редагування (відслідковується на далі автоматично по системній даті). В закладці Source описуються джерела інформації для побудови моделі (наприклад, «Опитування експертів предметної області та аналіз документації»). Закладка General служить для внесення імені проекту і моделі, імені і ініціалів автора та тимчасових рамок моделі – AS-IS і TO-BE (рис. 2).

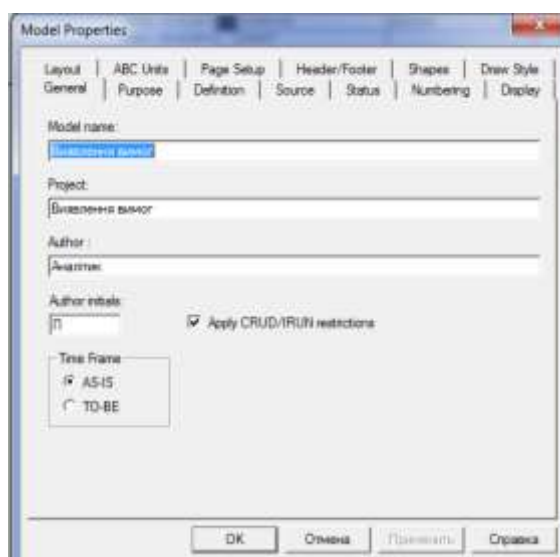


Рис. 2 Внесення імені проекту та моделі, імені та ініціалів автора і тимчасових рамок моделі

Моделі AS-IS і TO-BE

Зазвичай спочатку будується модель існуючої організації роботи - AS-IS (як є). Аналіз функціональної моделі надає змогу зрозуміти, де знаходяться найбільш слабкі міста, в чому будуть переваги нових бізнес-процесів і наскільки глибоким змінам піддається існуюча структура організації бізнесу. Деталізація бізнес процесів дозволяє виявляти недоліки організації навіть там, де функціональність здається наявною. Знайдені в моделі AS-IS недоліки можна справити при створенні моделі TO-BE (як буде) – моделі нової організації бізнес – процесів.

Технологія проектування ІС передбачає створення спочатку моделі AS-IS, її аналіз та покращення бізнес-процесів, тобто створення моделі TO-BE, і тільки на основі моделі TO-BE будується модель даних, прототип, а потім кінцевий варіант ІС.

Результат опису моделі можна отримати у звіті Model Report. Діалог налаштування звітів викликається із пункту меню Tools/Reports/Model Report.

В діалозі налаштування слід вибрати необхідні поля, при цьому автоматично відображається черговість виведення інформації в звіт (рис. 3, рис.4).

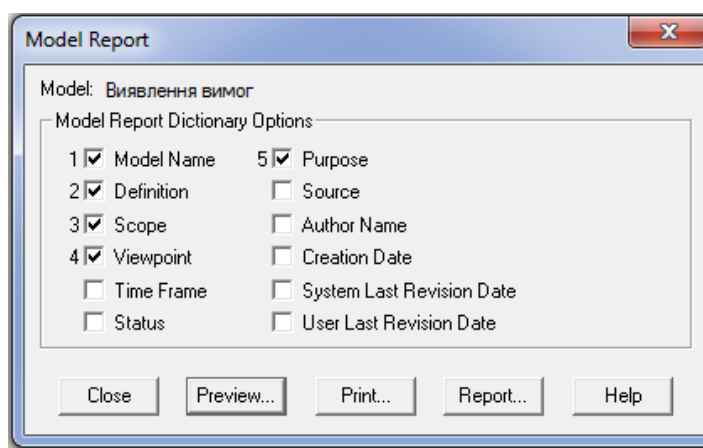


Рис. 3 Діалогове вікно для формування звіту по моделі

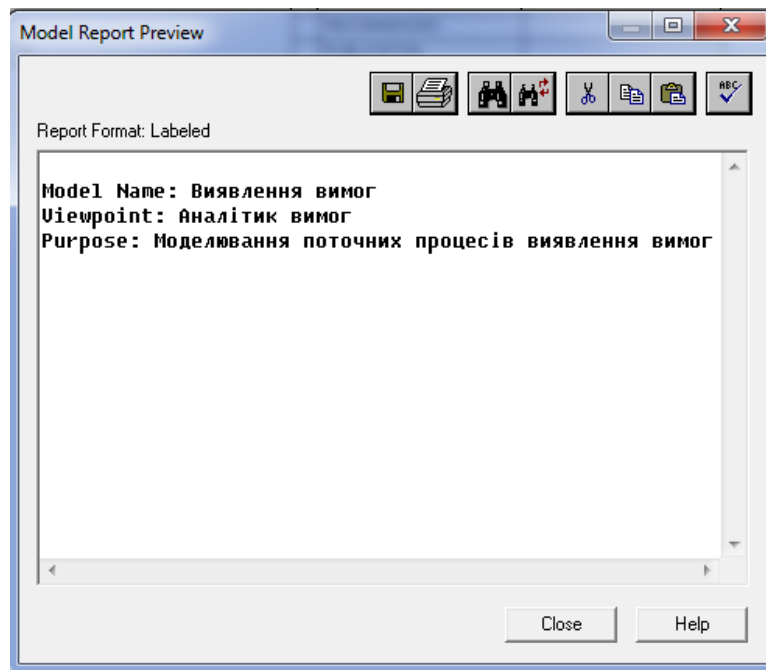


Рис. 4 Попередній перегляд звіту

Основа методології IDEF0

Основу методології IDEF0 складає графічна мова опису бізнес-процесів. Модель в нотації IDEF0 представляє собою сукупність ієрархічно упорядкованих і взаємопов'язаних діаграм. Кожна діаграма являється одиницею опису системи та розташовується на окремому аркуші.

Модель може складати чотири типи діаграми:

1. Контекстна діаграма (в кожній моделі може бути тільки одна контекстна діаграма).
2. Діаграма декомпозиції.
3. Діаграма дерева вузлів.
4. Діаграми тільки для експозиції (FEO).

Контекстна діаграма являється вершиною деревовидної структури діаграм і представляє собою загальний опис системи і її взаємодію з зовнішнім середовищем.

Після опису системи в цілому проводиться розбиття її на крупні фрагменти. Цей процес називається функціональною декомпозицією, а діаграми, які описують кожний фрагмент і взаємодію фрагментів, називаються *діаграмами декомпозиції*. Після декомпозиції контекстної

діаграми проводиться декомпозиція кожного великого фрагмента системи на більш дрібні і так далі, до досягнення потрібного рівня деталей опису. Після кожного сеансу декомпозиції проводяться сеанси експертизи – експерти предметної області вказують на відповідність реальних бізнес-процесів створеним діаграмам. Знайдені невідповідності виправляються і тільки після проходження експертизи без зауважень можна приступати до наступного сеансу декомпозиції. Так досягається відповідність моделі реальним бізнес-процесам на будь-якому рівні моделі.

Діаграма дерева вузлів показує ієрархічну залежність робіт, але не взаємозв'язки між ними. Діаграм дерев вузлів може бути в моделі скільки завгодно, оскільки дерево може бути побудоване на вільну глибину і не обов'язково з кореня.

Діаграми для експозиції (FEO) будуються для ілюстрації окремих фрагментів моделі, для ілюстрації альтернативної точки зору або для спеціальних цілей.

Роботи (Activity) визначають процеси, функції або задачі, які виконуються на протязі певного часу та мають розпізнавальні результати. *Роботи* відображаються у вигляді прямокутників. Всі роботи повинні бути названі і визначені. Ім'я роботи повинно відображати дію («Діяльність компанії», «Прийом замовлення», «Створення інформаційної системи» та ін.).

При створенні нової моделі (меню File/New) автоматично створюється контекстна діаграма з єдиною роботою, яка відображає систему в цілому (рис.5).

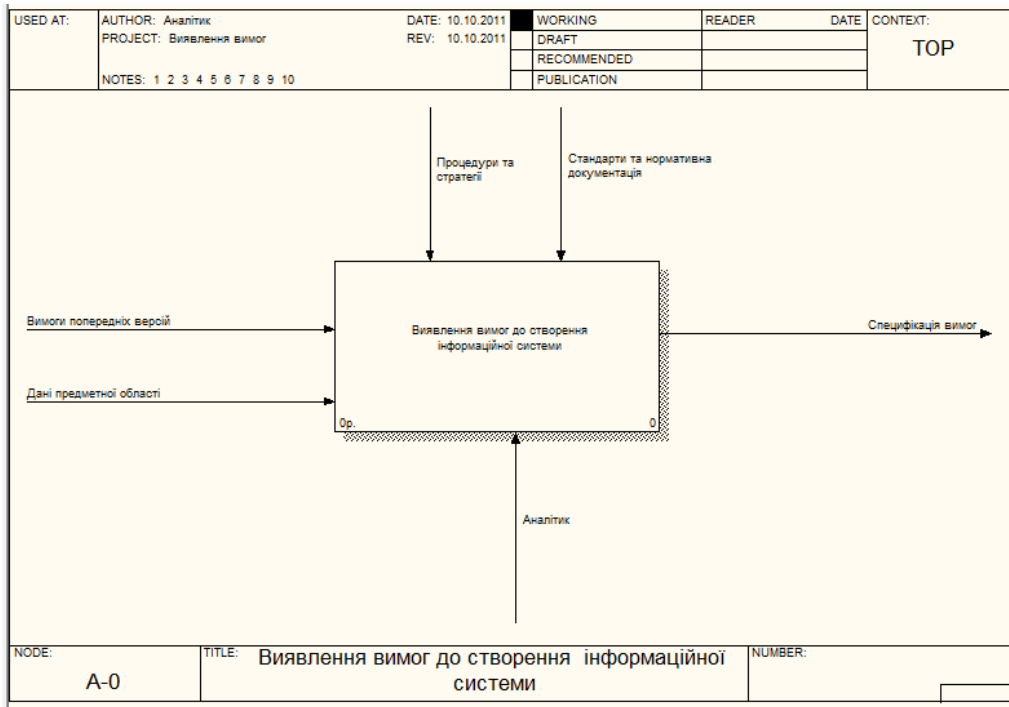


Рис. 5 Приклад контекстної діаграми

Діаграми декомпозиції мають «родственные работы», тобто дочірні роботи, які мають загальну «родственную работу». Далі створюємо діаграму декомпозиції (рис. 6, рис. 6а).

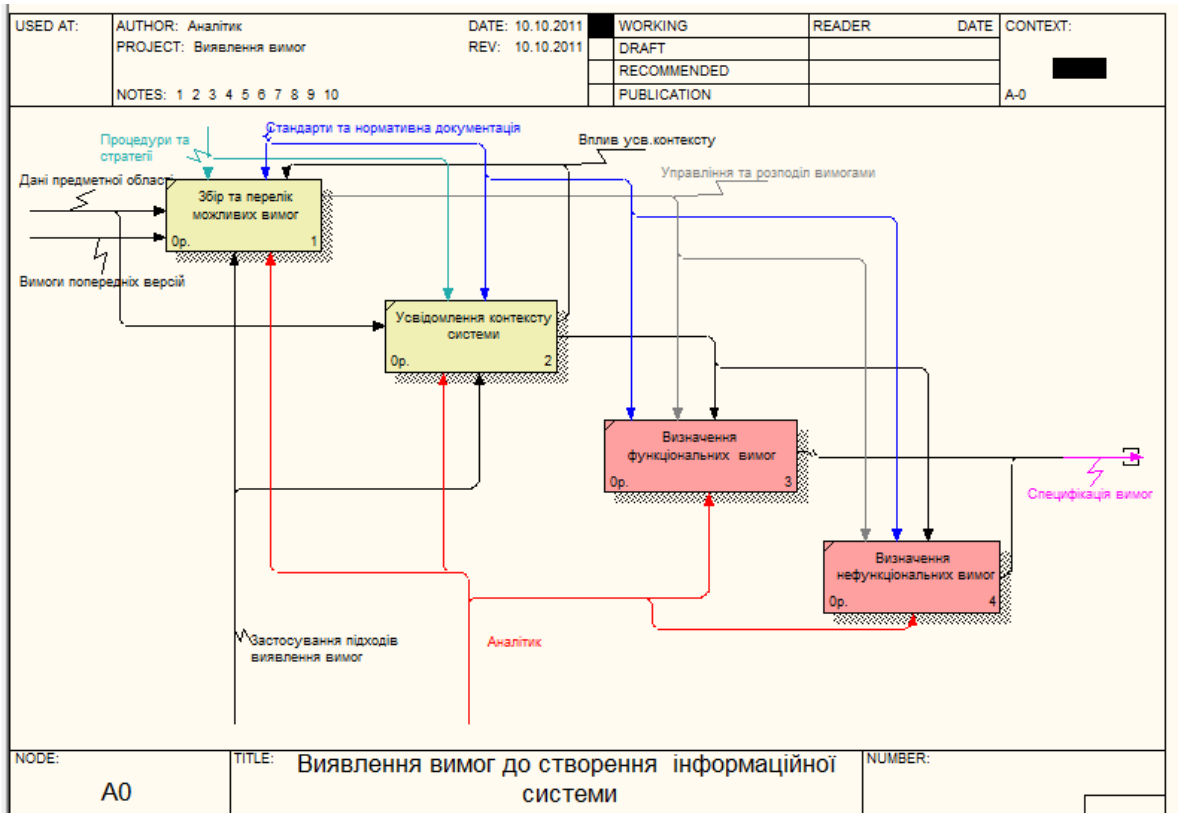


Рис. 6 Приклад діаграми декомпозиції

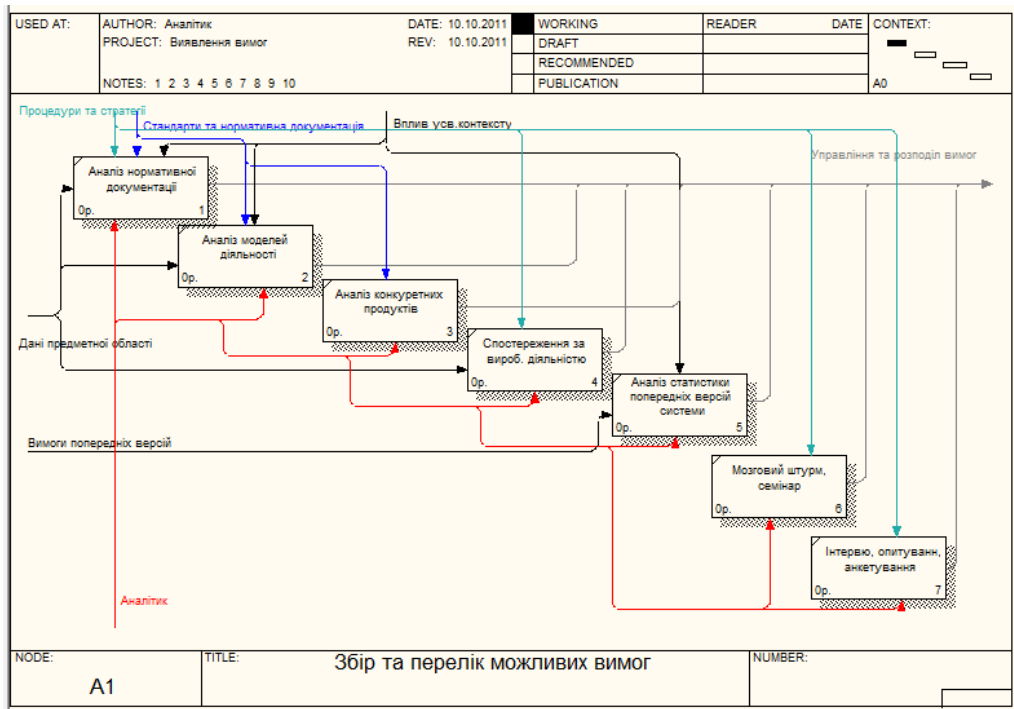


Рис. 6а Приклад діаграми декомпозиції роботи «Збір та перелік можливих вимог»

Кожна із робіт на діаграмі декомпозиції може бути в свою чергу декомповована (рис. 6а). На діаграмі декомпозиції роботи нумеруються автоматично зліва направо. Номер роботи розташований у правому нижньому кутку. У лівому верхньому кутку зображується невелика діагональна риска, яка показує, що дана робота не була декомповована.

Стрілки (Arrow) описують взаємодії робіт та представляють собою деяку інформацію (див. рис. 6).

В IDEF0 розрізняють п'ять типів стрілок:

Вхід (Input) – матеріал для інформації, який використовується або перетворюється роботою для отримання результату (виходу). Допускається, що робота може не мати ні однієї стрілки входу.

Управління (Control) – правила, стратегії, процедури, стандарти, якими керується робота. Кожна робота повинна мати хоча б одну стрілку управління.

Вихід (Output) – матеріали або інформація, які утворюються внаслідок виконання роботи. Кожна робота повинна мати хоча б одну стрілку виходу.

Механізм (Mechanism) – ресурси, які виконують роботу, наприклад персонал підприємства, пристрої та ін.. Стрілки механізму можуть не відображатися в моделі.

Виклик (Call) – спеціальна стрілка, яка вказує на іншу модель роботи. Стрілка виклику малюється як вихідна із нижньої частини роботи (рис.7). Стрілка виклику використовується для вказування того, що деяка робота виконується за межами модельованої системи. В VPwin стрілки виклику використовуються в механізмі злиття та розподілу моделей.

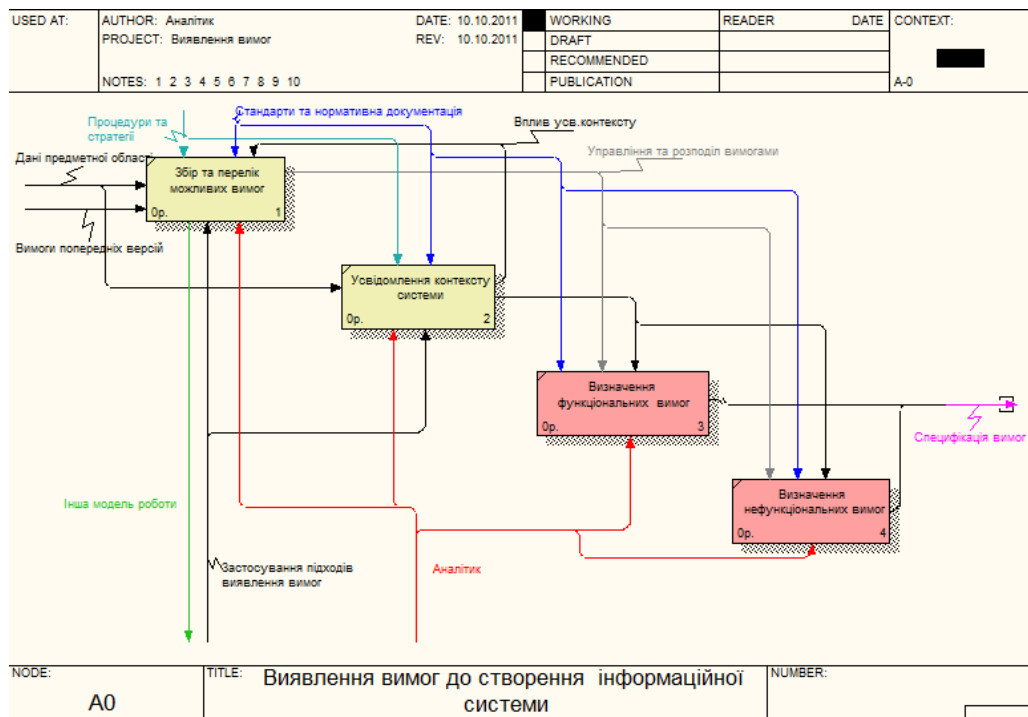


Рис.7 Стрілка виклику, яка з'являється при розщепленні моделі.

Граничні стрілки. Стрілки на контекстній діаграмі служать для опису взаємодії системи з навколишнім світом. Вони можуть починатися на границях діаграми і закінчуватися у робіт або навпаки.

Діаграми дерева вузлів та FEO

Діаграми дерева вузлів показують ієрархію робіт в моделі і дозволяють розглядати всю модель цілком, але не показує взаємозв'язки між роботами. Процес створення моделі є ітераційним, відповідно, роботи можуть змінювати своє розташування в дереві вузлів багатократно.

Для створення діаграми дерева вузлів слід вибрати в меню пункт Diagram/Add Node Tree (рис. 8-11).

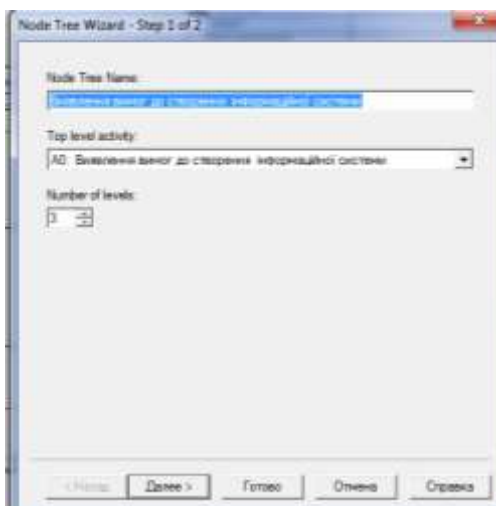


Рис. 8 Діалог налаштування діаграми дерева вузлів (крок 1)



Рис. 9 Діалог налаштування діаграми дерева вузлів (крок 2)

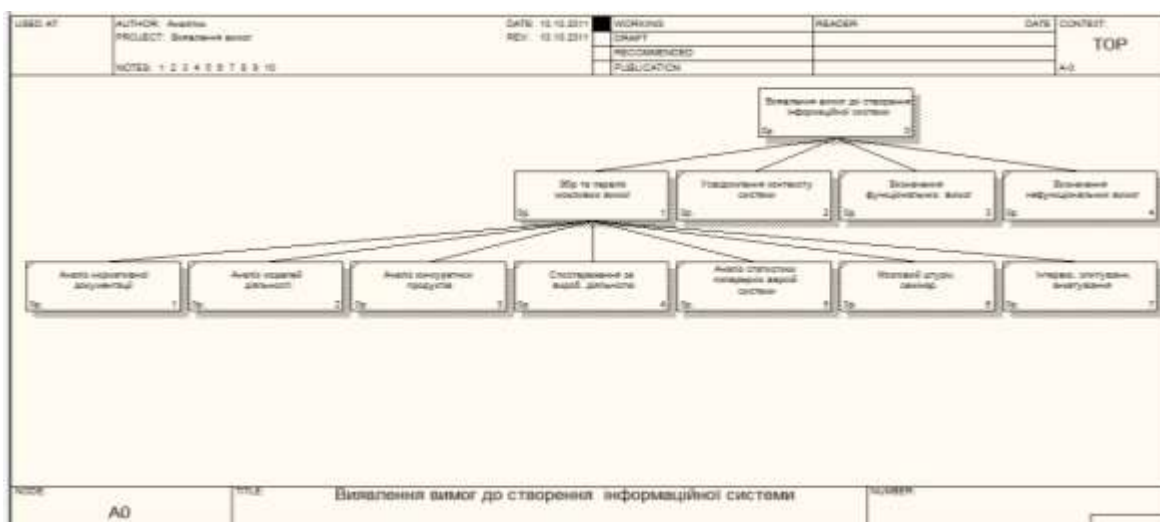


Рис. 10 Діаграма дерева вузлів.

Діаграми «тільки для експозиції» (FEO) часто використовується в моделі для ілюстрації інших точок зору, для відображення окремих деталей,

які не підтримуються явно синтаксисом IDEF0. Діаграми FEO дозволяють порушити любе синтаксичне правило, оскільки являється по суті просто картинками – копіями стандартних діаграм і не включаються в аналіз синтаксису. Для створення діаграми FEO слід вибрати пункт меню Diagram/Add FEO Diagram. У виникаючому діалозі слід вказати ім'я діаграми FEO.

Створення звітів в VPwin

VPwin має потужний інструмент генерації звітів. Звіти по моделі викладаються із пункту меню Report. Всього є сім типів звітів:

1. Model Report. Включає інформацію про контекст системи – ім'я моделі, точку зору, область, ціль, ім'я автора, дату створення та ін.
2. Diagram Report. Звіт по конкретній діаграмі. Включає список об'єктів (робіт, стрілок, схованок даних, зовнішні посилання та ін.).
3. Diagram Object Report. Найбільш повний звіт по моделі. Може включати повний список об'єктів моделі (робіт, стрілок з вказуванням їх типів та ін.) та властивості визначені користувачем.
4. Activity Cost Report. Звіт про результати вартісного аналізу.
5. Arrow Report. Звіт по стрілкам. Може мати інформацію із словника стрілок, інформація про роботу – джерело, роботі- призначенні стрілки і інформацію про розгалуження та злиття стрілок.
6. Data Usage Report. Звіт про результати зв'язку моделі процесів і моделі даних.
7. Model Consistency Report. Звіт, який має список синтаксичних похибок моделі.

Завдання

1. Виберіть свій варіант завдання з лабораторної роботи №1.
2. Проаналізуйте виявлені Вами вимоги в лабораторній роботі №1.
3. Створіть модель виявлення вимог до системи відповідно Вашого завдання засобами VPwin.

4. Побудуйте контекстну діаграму системи.
5. Побудуйте діаграму декомпозиції до контекстної діаграми.
6. Побудуйте діаграму декомпозиції до одного із виду робіт.
7. Побудуйте діаграму дерева вузлів.
8. Перегляньте побудову діаграм тільки для експозиції.
9. Коротко опишіть логіку побудови кожної із діаграм.
10. Згенеруйте наступні звіти: Model Report, Diagram Report, Activity Cost Report та представте їх у звіті.
11. Оформіть лабораторну роботу відповідно вимог до звіту.

Лабораторна робота №4

Методи документування вимог.

Розробка технічного завдання засобами «Майстра технічного завдання»

Теоретичні положення

Технічне завдання (ТЗ) — вихідний документ для проектування споруди чи промислового комплексу, конструювання технічного пристрою (приладу, машини, системи керування тощо), розробки автоматизованої системи, створення програмного продукту або проведення науково-дослідних робіт (НДР), відповідно до якого проводиться виготовлення, приймання при введенні в дію та експлуатація відповідного об'єкта.

Згідно з ГОСТ 34.602-89 ТЗ є основним документом, що визначає вимоги і порядок створення (розвитку або модернізації) інформаційної системи, відповідно до якого проводиться її розробка і приймання при введенні в дію. Згідно з діючим стандартами ТЗ повинно включати в себе такі відомості про об'єкт розробки:

1. Найменування об'єкта розробки, та область застосування:

повне найменування об'єкта та його умовне позначення,
шифр теми або шифр (номер) договору,
перелік документів, на підставі яких створюється проект, ким і коли затверджені ці документи,
планові терміни початку та закінчення робіт із створення об'єкта.

2. Підстава для розробки та назва проектної організації:

найменування підприємств розробника і замовника системи та їхні реквізити;
перелік юридичних та фінансових документів, на підставі яких створюється система, ким і коли затверджені ці документи;
відомості про джерела та порядок фінансування робіт.

3. Мета розробки.

4. Джерела розробки.

Тут повинні бути перераховані документи та інформаційні матеріали (техніко-економічне обґрунтування, звіти про закінчені науково-дослідні роботи, інформаційні посилання на вітчизняні і зарубіжні аналоги та ін.), на підставі яких розроблялося ТЗ і які мають бути використані при створенні системи.

5. Технічні вимоги, які включають:

склад об'єкта та вимоги до його конструктивного виконання;
показники призначення та економічного використання сировини, матеріалів, палива і енергії;

вимоги до надійності;

вимоги до технологічності;

вимоги до рівня уніфікації і стандартизації;

вимоги безпеки при роботі обладнання;

естетичні і ергономічні вимоги;

вимоги до складових частин продукції, сировини і експлуатаційних матеріалів;

вимоги патентної чистоти;

вимоги експлуатації, вимоги до технічного обслуговування і ремонту;

вимоги до категорії якості.

6. Економічні показники:

гранична ціна;

економічний ефект;

термін окупності витрат на розробку і освоєння об'єкта;

допустима річна потреба в об'єкті проектування.

7. Порядок контролю і приймання об'єкта:

види, склад, обсяг і методи випробувань системи та її складових частин (види випробувань відповідно до діючих норм, які поширюються на систему, що розробляється);

загальні вимоги до приймання робіт (продукції) за стадіями (перелік учасників, місце і терміни проведення), порядок узгодження і затвердження приймальної документації;

статус приймальної комісії.

Нормативні документи

ГОСТ 2.114-95 Единая система конструкторской документации. Технические условия;

ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению;

ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

При розробці технічного завдання зміст розробляється відповідно своєї розробки.

Майстер технічного завдання

Технічне завдання (ТЗ) можна описувати засобами *MS Word* чи іншого редактора, а також засобами *майстра технічного завдання*.

«*Майстер ТЗ*» допоможе нам скласти швидко та на високому рівні ТЗ.

Після запуску програми вводимо назву проекту та визначаємо до якого виду він відноситься: новий, програма, сайт (Рис. 1).

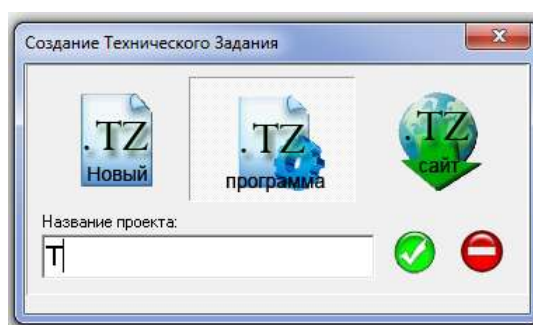


Рис. 1 Діалогове вікно створення проекту

Інтерфейс програми інтуїтивний та простий.

Завдання

1. Виберіть свій варіант завдання за наступною схемою:

№ з/ж	Вид проекту	Постановка завдання
1.	База даних	Магазин по продажі комп'ютерних ігор
2.	База даних	Магазин по продажі взуття
3.	Новий проект	АРМ «Бухгалтера по нарахуванні зарплатні»
4.	База даних	Народних аматорських колективів
5.	Новий проект	АІС «Відділ кадрів»
6.	Розробка сайту	Компанія «Тепла підлога»
7.	Розробка сайту	Інтернет магазину по продажі будівельних матеріалів
8.	База даних	«Книжковий магазин»
9.	База даних	Магазин продажу побутової техніки
10.	Новий проект	АІС обміну електронними документами
11.	Розробка сайту	Інтернет магазин по продажі метало пластикових вікон
12.	Розробка сайту	Група ІІІ-91
13.	База даних	Магазин продажу комп'ютерної техніки
14.	Розробка сайту	Компанія, яка займається організацією мереж
15.	Розробка сайту	Інтернет магазин чоловічого одягу

2. Виявіть вимоги до поставленої перед Вами задачі.
3. Розробіть технічне завдання відповідно Вашого варіанту засобами «Майстра ТЗ».
4. Оформіть лабораторну роботу відповідно вимог до звіту.

Лабораторна робота №5

Створення специфікації вимог до ПЗ

Структуровані мови опису специфікацій

Теоретичні положення

Кожна організація, яка спеціалізується на розробці ПЗ повинна прийняти один або декілька стандартних шаблонів специфікації вимог до ПЗ для використання в проектах. Можна використовувати різні шаблони специфікацій (Davis, 1993; Robertson and Robertson, 1999; Leifingwell and Widrig, 2000). Багато хто використовує шаблони, створені на основі того, що описано в IEEE Standart 830-1998, «IEEE Recommended Practice for Software Requirements Specifications» (IEEE, 1998b). Ці шаблони підходять до різних проектів, але в них зустрічаються обмеження та не зрозумілі місця. Тому, якщо Ви беретесь за проекти різних типів і розмірів, розробіть для проектів великого класу окремі шаблони специфікацій.

Наведемо приклад шаблону специфікації та візьмемо його за основу при написанні своєї власної специфікації.

Шаблон специфікації

1. Введення

- 1.1 Призначення
- 1.2 Узгодження, прийняті у документах
- 1.3 Аудиторія та рекомендації по змісту
- 1.4 Границі проекту
- 1.5 Посилання

2. Загальний опис

- 2.1 Загальний погляд на продукт
- 2.2 Особливості продукту
- 2.3 Класи та характеристики користувачів
- 2.4 Операційне середовище

2.5 Обмеження дизайну та реалізації

2.6 Документація для користувачів

2.7 Передбачення та залежності

3. Функції системи

3.x Функція системи X

3.x.1 Опис та пріоритети

3.x.2 Послідовності «дія-реакція»

3.x.3 Функціональні вимоги

4. Вимоги до зовнішнього інтерфейсу

4.1 Інтерфейси користувача

4.2 Інтерфейси обладнання

4.3 Інтерфейси ПЗ

4.4 Інтерфейси передачі інформації

5. Інші нефункціональні вимоги

5.1 Вимоги до виробничості («производительности»)

5.2 Вимоги до охорони праці

5.3 Вимоги до безпеки

5.4 Атрибути якості

6. Інші вимоги

Додаток А. Словник термінів

Додаток Б. Моделі аналізу

Додаток В. Перелік питань

Наведений шаблон специфікації вимог до ПЗ створений на основі стандарту IEEE 830, в ньому наведено багато прикладів додаткових вимог до продукту, які Ви можете включити в свою специфікацію.

Змініть шаблон відповідно особливостей Вашого проекту. Якщо якийсь розділ шаблону не можна застосувати до Вашого конкретного проекту, не вилучайте його заголовок, а вкажіть, що його не можна використати.

Інколи, при написанні специфікації, фрагмент інформації логічно підходить для декількох розділів шаблону, тоді проаналізуйте інформацію та послідовно розмістіть її у потрібний розділ.

Далі розглянемо, яку інформацію необхідно включити в кожний розділ специфікації вимог до ПЗ:

1. Введення

Введення представляє собою огляд, який дозволяє читачам розібратися в структурі та принципі використання специфікації вимог до ПЗ.

1.1. Призначення

Визначає продукт або додаток, вимоги для якого вказані у цьому документі, а також редакцію та номер випуску. Якщо ця специфікація вимог до ПЗ відноситься тільки до частини системи, ідентифікуйте цю частину або підсистему.

1.2. Узгодження, прийняті у документах

Опишіть всі стандарти або типографічні стандарти, включаючи стилі тексту, особливості виділення або зауваження. Наприклад, вкажіть, чи спадковий пріоритет, вказаний для вимог вищого рівня, всіма їх деталізованими вимогами або кожне положення про функціональні вимоги повинно мати власний пріоритет.

1.3. Аудиторія та рекомендації по змісту

Перерахуйте користувачів, для яких призначена ця специфікація вимог до ПЗ. Опишіть зміст документа та його структуру. Порекомендуйте найбільш зручну послідовність читання документу для кожного з класу користувачів.

1.4. Границі проекту

Коротко опишіть ПЗ та його призначення. Вкажіть, як пов'язаний продукт з користувачами або корпоративними цілями, а також з бізнес-цілями та стратегіями. Якщо є окремий документ про образ та границі проекту, не повторюйте його змісту, а просто зробіть посилання на нього. Якщо специфікацію вимог до ПЗ пропонується розробляти поступово, то

вона повинна мати власне положення про образ та границі продукту в якості підрозділу стратегічного образу з довгим терміном.

1.5. Посилання

Перерахуйте всі документи або інші ресурси, на які Ви посилаєтесь в цій специфікації, а також гіперпосилання на них. Це можуть бути керівництва по стилям користувацького інтерфейсу, контракти, стандарти, специфікації до системних вимог, документи про варіанти використання, специфікації інтерфейсу, концептуальні документи і специфікації вимог до ПЗ для продуктів, на які ви посилаєтесь. Об'єм інформації повинен бути достатнім для того, щоб користувач зумів при необхідності отримати доступ до кожного вказаного матеріалу, а саме: назві, ім'я автора, номер версії, дата та джерело або розташування (наприклад, мережева папка або URL).

2. Загальний опис

В цьому розділі представлений загальний огляд продукту та середовища, в якому він буде застосовуватися, передбачена користувацька аудиторія, а також відомі обмеження, передбачення та залежності.

2.1 Загальний погляд на продукт

Опишіть склад та походження продукту. Поясніть, чи являється він новим членом зростаючого сімейства продуктів, новою версією існуючої системи, заміною існуючого додатку або взагалі новим продуктом. Якщо специфікація вимог визначає компонент більш крупної системи, то вкажіть, яке співвідношення цього ПЗ з усією системою та визначте основні інтерфейси між ними.

2.2 Особливості продукту

Перерахуйте основні особливості продукту або його головні функції. Деталі будуть викладені в розділі 3 специфікації вимог до ПЗ, тут лише потрібно їх вказати. Також тут можна проілюструвати основні групи вимог та їх взаємовідношення, наприклад показати діаграму потоків даних вищого рівня, діаграму варіанту використання або діаграму класів.

2.3 Класи та характеристики користувачів

Визначте різні класи користувачів, які будуть працювати з розроблюваним продуктом та опишіть їх характеристики. Деякі вимоги можуть відноситися тільки до певних класів користувачів. Визначте привілегіовані класи користувачів.

2.4 Операційне середовище

Опишіть робоче середовище ПЗ, включаючи апаратні засоби, операційні системи та їх версії, а також географічне місце розташування користувачів, серверів та баз даних. Перерахуйте всі інші компоненти ПЗ або додатків, з якими система повинна бути сумісна. В документі про образ та границі проекту ця інформація може бути розкрита більш детально.

2.5 Обмеження дизайну та реалізації

Опишіть будь-які фактори, які обмежать можливості, доступні розробникам, а також логічно обґрунтуйте кожне положення. Обмеження можуть бути такого роду:

1. Певні технології, засоби, мови програмування та бази даних, які необхідно використовувати або уникати.
2. Обмеження, які накладаються операційним середовищем продукту, наприклад типи та версії встановлених Web-браузерів.
3. Обов'язкове погодження або стандарти розробки (наприклад, якщо обслуговувати ПЗ будуть клієнти, то вони повинні вказати особливості дизайну і стандарти програмування, які субпідрядник повинен дотриматись).
4. Зворотна сумісність з продуктом, які були випущені раніше.
5. Обмеження, які накладаються бізнес-правилами (вони повинні бути зафіксовані в інших документах).
6. Обмеження, які пов'язані з обладнанням, наприклад вимоги до термінів, обмеження пам'яті або процесора, розмір, вага, матеріали або затрати.

7. Погодження, які пов'язані з користувацьким інтерфейсом існуючого продукту, які необхідно дотримуватися при покращенні існуючого продукту.
8. Стандартний формат обміну даними, наприклад XML.

2.6 Документація для користувачів

Перерахуйте всі компоненти користувацької документації, яка поставляється разом з ПЗ. В них можуть входити керівництва для користувача, он-лайн - довідки та навчальні програми. Визначте всі необхідні формати, стандарти і засоби поставки документації.

2.7 Передбачення та залежності

Передбачення (assumption) називається положення, яке вважається істинним при відсутності доказу або визначаючої інформації. Проблеми можливі в тому разі, якщо передбачення не вірні, не знаходяться в загальному використанні або вони змінюються, через це певні передбачення можна віднести до групи ризиків проекту. Один користувач специфікації може вважати, що продукт буде відповідати особливому стандарту користувацького інтерфейсу, тоді як інший користувач передбачає дещо зовсім інше.

Розробник може думати, що певний набір функцій написаний спеціально для цього додатку, аналітик – що він буде взятий з попереднього проекту, а менеджер проекту – що передбачається придбати комерційну бібліотеку функцій.

Визначте всі *залежності* (dependencies) проекту від зовнішніх факторів, таких як, дата випуску наступної версії операційної системи або випуск промислового стандарту. Якщо Ви плануєте встановити в систему компоненти, які розробляються в іншому проекті, то Ви залежите від своєчасної їх поставки. Якщо ці залежності вже десь задокументовані, наприклад, в плані проекту, зробіть посилання на них.

3. Функції системи

Шаблон структурований за допомогою особливостей системи – це ще один спосіб систематизації функціональних вимог. Інші методи класифікації: по варіантам використання, режиму роботи, класам користувачів, стимулам, реакціям, класам об'єктів або функціональної ієрархії. Можливі також комбінації цих елементів, наприклад варіанти використання в середині класів користувачів. Не існує єдиного правильного методу організації; виберіть те, при якому користувачам буде легше розуміти передбачувані можливості продукту. Наведемо приклад опису схеми особливостей.

3.x Функція системи X

Вкажіть назву особливості декількома словами, наприклад «3.1 Перевірка правопису». Також назвіть підрозділи з 3.x.1 по 3.x.3 для кожної особливості системи.

3.x.1 Опис та пріоритети

Коротко опишіть особливості системи та вкажіть, чи має вона високий, середній або низький пріоритет. Пріоритет являється динамічною характеристикою, вони можуть змінюватися в ході проекту. Якщо Ви використовуєте засіб керування вимогами, визначте атрибут вимог для пріоритету.

3.x.2 Послідовності «дія-реакція»

Перерахуйте послідовності дій, які впливають на систему (дії користувачів, сигнали зовнішніх пристроїв та ін.), а також відгуки системи, які визначають реакцію конкретної функції.

3.x.3 Функціональні вимоги

Перерахуйте по пунктам деталізовані функціональні вимоги, які пов'язані з цією особливістю. Для того щоб користувач міг задіяти цю особливість або реалізувати варіанти використання, обов'язково слід передбачити певні функції ПЗ. Опишіть як продукт повинен реагувати на очікувані помилки, неправильне введення інформації або невірні дії. Надайте кожній функціональній вимозі унікальне ім'я.

4. Вимоги до зовнішнього інтерфейсу

Вимоги до зовнішнього інтерфейсу визначають обладнання, ПЗ або елементи бази даних, з якими система або компонент повинні взаємодіяти. Якщо у різних частин продукту різні зовнішні інтерфейси, вставте подібний розділ в деталізовані вимоги для кожної такої частини.

4.1 Інтерфейси користувача

Опишіть логічні характеристики кожного користувацького інтерфейсу, який необхідний системі. Деякі з них представлені нижче:

1. Посилання на стандарти графічного інтерфейсу користувачів або стильові рекомендації для сімейства продукту, яких необхідно дотримуватись.
2. Стандарти шрифтів, значків, назв кнопок, зображень, кольорових схем, послідовностей полів вкладок, елементів управління, які часто використовуються.
3. Конфігурація екрану або обмеження розрішення.
4. Стандартні кнопки, функції або посилання переміщення, однакові для всіх екранів, наприклад кнопка довідки.
5. Швидкі клавіші.
6. Стандарти відображення повідомлень.
7. Стандарти конфігурації для спрощення локалізації ПЗ.
8. Спеціальні можливості для користувачів з вадами зору.

Детально документуйте деталі користувацького інтерфейсу, такі як конфігурація певних діалогових вікон, в окремій специфікації користувацького інтерфейсу, а не в специфікації вимог до ПЗ (Але ці моделі не являються офіційними планами екрану). Якщо в специфікації вимог до ПЗ говориться про покращення існуючої системи, то варто включити у документ зображення екранів в том вигляді, як вони будуть реалізовані.

4.2 Інтерфейси обладнання

Опишіть характеристики кожного інтерфейсу між компонентами ПЗ та обладнанням системи. В опис можуть входити типи пристроїв, які

підтримуються, взаємодія даних і елементів керування між ПЗ та обладнанням, а також протоколи взаємодії, які будуть використовуватися.

4.3 Інтерфейси ПЗ

Опишіть поєднання продукту і інших компонентів ПЗ, а також баз даних, операційних систем, засоби, бібліотеки та інтегровані комерційні компоненти. Вкажіть назви елементів повідомлень, даних та елементів керування, обмін якими відбувається між компонентами ПЗ. Опишіть служби, які необхідні зовнішнім компонентам ПЗ та взаємодію між компонентами. Визначте дані, до яких будуть мати доступ компоненти ПЗ. Якщо механізм надання загального доступу до даних повинен бути реалізований певним чином, наприклад в якості глобальної області даних, то вкажіть його як обмеження.

4.4 Інтерфейси передачі інформації

Вкажіть вимоги для будь-яких функцій взаємодії, які будуть використовуватися продуктом, включаючи електронну пошту, Web - браузер, протоколи мережевого з'єднання та електронні форми. Визначте відповідні форми повідомлень. Опишіть особливості безпеки взаємодії або шифрування, частоту передачі даних та механізм синхронізації.

5. Інші нефункціональні вимоги

В цьому розділі описують інші нефункціональні вимоги, які не відносяться до вимог інтерфейсу, які представлені в розділі 4 та до обмежень, які описані в розділі 2.5.

5.1 Вимоги до виробничості («производительности»)

Вкажіть спеціальні вимоги до виробничості для різних системних операцій. Обґрунтуйте їх необхідність для того, щоб допомогти розробникам прийняти правильне рішення, стосовно дизайну. Наприклад, із-за жорстких вимог до часу відгуку бази даних розробники можуть зеркалізувати базу даних в декількох географічних місцях розташування або денормалізувати пов'язаних таблиць бази даних для отримання більш швидкої відповіді на

запит вашого проекту. Пропустіть цей розділ, якщо всі необхідні вимоги вже розписані у інших розділах.

5.2 Вимоги до охорони праці

5.3 Вимоги до безпеки

5.4 Атрибути якості

6. Інші вимоги

Додаток А. Словник термінів

Поясніть вимоги, які необхідно знати користувачу для правильного розуміння специфікації вимог до ПЗ, включаючи скорочення. Розшифруйте кожне скорочення та наведіть їх визначення.

Додаток Б. Моделі аналізу

В цьому не обов'язковому розділі описуються та наводяться діаграми потоку даних, діаграми класів, діаграми переходу стану та діаграми «сутність-зв'язок».

Додаток В. Перелік питань

Це динамічний перелік ще не вирішених проблем, пов'язаних з вимогами. Це можуть бути елементи, які помічають так «TBD» (to be determined – необхідно визначити), відкладені рішення, необхідна інформація, невирішені конфлікти та ін. Все це не обов'язково включати в специфікацію вимог до ПЗ.

Приклад та більш детальний опис шаблону специфікації наведений в конспекті лекцій.

Завдання

1. Виберіть свій варіант завдання з лабораторної роботи №1 або №4.
2. Проаналізуйте виявлені Вами вимоги.
3. Проаналізуйте учасників процесу виявлення вимог.
4. Проаналізуйте учасників проекту, яким необхідна специфікація та як саме.

5. Повторіть матеріали по стандарту IEEE 830.
6. Ознайомтесь з запропонованим у теоретичних положеннях шаблоном специфікації вимог до ПЗ.
7. Складіть специфікацію вимог по запропонованому шаблону відповідно свого варіанту.
8. Побудуйте діаграми потоку даних, діаграми класів, діаграми переходу стану та діаграми «сутність-зв'язок» у відповідних розділах.
9. Коротко опишіть логіку побудови кожної із діаграм.
10. Оформіть лабораторну роботу відповідно вимог до звіту.

Лабораторна робота №6

Інспекція і атестація вимог до програмного забезпечення та виявлення конфліктів

Теоретичні положення

Аналіз вимог – це трансформація інформації, отриманої від користувачів (та інших зацікавлених осіб) в чітко і однозначно визначених вимогах, які подалі передаються інженерам для реалізації в програмному коді.

Результатом аналізу вимог повинні бути інтерпретовані вимоги, реалізація яких перевірена, а вартість і ресурси – передбачені.

Аналіз вимог передбачає їх деталізацію, яка гарантує, що вимоги розуміють всі зацікавлені особи, а також детальне дослідження вимог на предмет похибок та інших недоліків.

Крім того, аналіз включає створення прототипів, аналіз здійсненності і узгодження пріоритетів.

Ціль аналізу – достатньо якісно і детально описати вимоги, які дозволяють менеджерам реалістично оцінювати всі затрати на проект, а технічному персоналу – почати проектувати, збирати та тестувати.

Варто деякі вимоги представити та проаналізувати декількома способами в текстовій та графічній формі. Існує ряд практик для перевірки вимог, наприклад огляд вимог, прототипіювання, розробка моделі, приймальні тести та ін.

Процес аналізу вимог тісно пов'язаний, з однієї сторони, з аналізом проблемної області, а з іншої – з архітектурним аналізом і проектуванням. Часто на практиці буває складно розподілити границі компетенцій цих потоків робіт. Так, модель аналізу потоків даних, широко використовується в аналізі проблемної області.

Для визначення прийомів, методик, мов моделювання при аналізі вимог приведемо три практичні рекомендації:

1. Аналіз вимог покликаний вивчати взаємодію автоматизованої інформаційної системи і її середовища, так як користувачі, мережеві і системні компоненти знаходяться поза системою. Відповідно бізнес-моделі, які описують взаємодію між компонентами організаційної системи можна розглядати лише як «сирі» для виявлення вимог, але не як моделі, які описують вимоги.
2. Аналіз вимог повинен знаходити відповідь на питання *ЩО* робить система, а не *ЯК* вона це робить. *Діаграма взаємодії об'єктів*, яка реалізує той чи інший варіант використання, можна розглядати як ілюстрацію внутрішнього пристрою системи, корисну для програміста, ніж модель для замовника.
3. Для моделювання аналізу вимог слід використовувати моделі, найбільш адекватно пояснюючі функціональність системи і особливості її використання.

Моделі UML, які пояснюють функціональність системи:

1. *Діаграмми варіантів використання UML, Use Case Diagram* – це одне з найпростіших представлень системи. Діаграма задумана так, що дати найбільш загальне представлення про функціональність системи (її компонентів), не вдаючись в деталі взаємозв'язків функцій.
2. *Діаграма дій* дозволяє детально проілюструвати окремий варіант використання та його сценарії з необхідною ступінню деталізації.
3. *Діаграма станів* в аналізі вимог використовується, коли потрібно дослідити поведінку системи, як кінцевого автомату. Вона використовується для моделювання окремих ситуацій.

Моделі UML, які пояснюють внутрішній пристрій системи:

1. *Діаграми класів* показують статистичну структуру проблемної області. Для аналізу взаємодії об'єктів - екземплярів класів під час реалізації варіанта використання в UML передбачені дві діаграми взаємодії: діаграми кооперації і діаграма послідовності.

Більш детальна інформація представлена в конспекті лекцій.

Завдання

1. Виберіть свій варіант завдання з лабораторної роботи №1.
2. Проаналізуйте виявлені Вами вимоги.
3. Проаналізуйте учасників процесу виявлення вимог.
4. За допомогою діаграми варіантів використання UML надайте загальне представлення про функціональність системи. Опишіть діаграму.
5. За допомогою діаграми дій детально проілюструйте окремий варіант використання ПП та його сценарії з необхідною ступінню деталізації. Опишіть діаграму.
6. За допомогою діаграми станів змодельуйте окрему ситуацію для поведінки системи. Опишіть діаграму.
7. За допомогою діаграми класів поясніть внутрішній пристрій системи. Опишіть діаграму.
8. Напишіть ґрунтовний висновок до роботи.
9. Оформіть лабораторну роботу відповідно вимог до звіту.