

Лабораторна робота 1

Аналіз та формування вимог до програмного забезпечення

МЕТА: Навчитись аналізувати і формувати перелік вимог до програмного забезпечення інформаційної системи, що розробляється.

1. ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Основні визначення бізнес-архітектури

Бізнес-драйвер (Business Driver) – це внутрішня або зовнішня зміна чи вплив, що викликає реакцію бізнесу [2]. Цей вплив не обов'язково є негативним.

Приклади:

Збільшення податку в країні, конкуренція на ринку, зростання попиту на певну послугу, висока кількість тих, хто відмовляється від послуг тощо.

Бізнес-драйвер – це ресурс, процес або умова, життєво важливі для подальшого успіху та зростання бізнесу. Компанія повинна визначити драйвери свого бізнесу та намагатися максимізувати ті, що знаходяться під її контролем.

Завжди існують зовнішні чинники бізнесу, на які компанія не може вплинути, наприклад економічні умови чи торговельні відносини з іншими країнами.

Бізнес-ціль (Business goal) являє собою заяву про наміри, напрямок або бажаний кінцевий стан організації та її стейкхолдерів.

Приклади:

Збільшення клієнтів на 20% щороку протягом наступних 3 років.

Зменшення експлуатаційних витрат на 10% до кінця року.

Збільшення прибутку на 1 мільйон доларів цього року.

Цілі є фундаментальною частиною будь-якого хорошого бізнес-плану. Вони вказують, куди бізнес йде ваш і який шлях для того обирає.

Бізнес-завдання (Business Objectives) — це активний елемент для досягнення бізнес-цілей.

Приклад:

Business goal: збільшити прибуток на \$1 млн цього року:

- Business Objectives відділу продажів: збільшити продажі на 40% цього року
- Business Objectives виробничого відділу: знизити віддачу на 10%.

Успішний бізнес базується як на цілях, так і на завданнях, оскільки вони пояснюють мету бізнесу та допомагають визначити необхідні дії. Цілі — це загальні заяви про бажане досягнення, тоді як завдання — це конкретні кроки або дії, які ви робите для досягнення своєї мети (рис. 1).

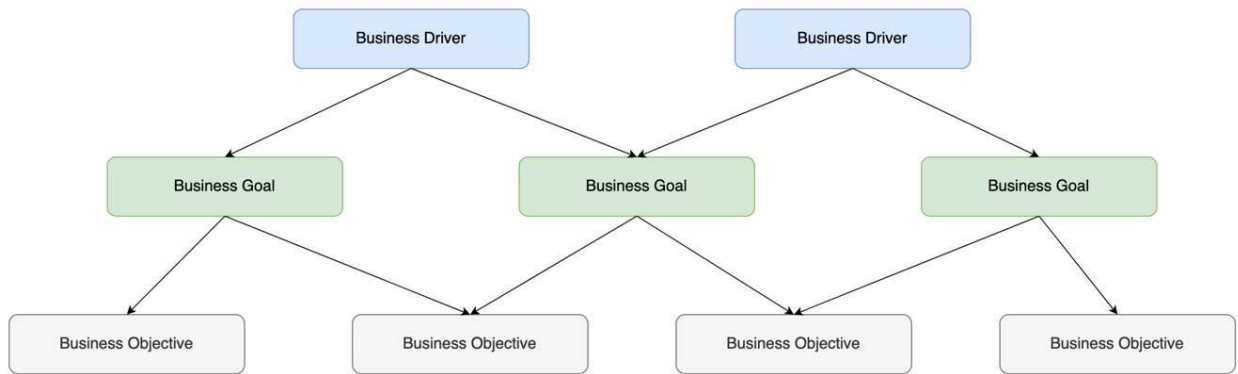


Рис. 1 – Взаємозв'язок Business Driver, Business Goal та Business Objective

Як цілі, так і завдання мають бути конкретними та вимірюваними.

Критерії SMART допомагають у ефективному формуванні завдань та цілей. SMART - це аббревіатура, яка означає:

- **Specific** (Конкретна): мета вимоги має бути чітко визначеною та конкретною, відповідати на питання що, чому та як.
- **Measurable** (Вимірювана): вимога має бути вимірюваною з точки зору кількісних критеріїв або етапів, щоб можна було відслідковувати та оцінювати прогрес.
- **Achievable** (Досяжна): вимога має бути реалістичною та досяжною, враховуючи наявні ресурси, навички та час.
- **Relevant** (Релевантна): вимога має відповідати загальним цілям або пріоритетам проекту чи організації.
- **Time-bound** (Визначена у часі): вимога повинна мати конкретний часовий проміжок або кінцевий термін для досягнення.

Використання критеріїв SMART допомагає переконатися, що вимоги є чіткими, досяжними та значущими, а прогрес у досягненні цих вимог можна відстежувати й ефективно оцінювати. Критерії SMART можна застосовувати до особистих цілей, цілей команди чи цілей організації.

Приклад:

Business Goal: *система програмного забезпечення здатна обробляти принаймні 10000 одночасних сеансів користувача без зниження продуктивності до кінця року.*

Конкретність: ціль є чітко визначеною та конкретною — забезпечити, щоб програмна система могла обробляти щонайменше 10000 одночасних сеансів користувача без зниження продуктивності.

Вимірюваність: ціль піддається вимірюванню – команда може відстежувати прогрес, вимірюючи поточну здатність системи обробляти одночасні сеанси користувачів, встановлюючи ціль у 10 000 і відстежуючи продуктивність під час навантажувального тестування та перевірки прийнятності користувача.

Досяжність: ціль досяжна – команда визначила конкретні зміни, які можна внести в архітектуру або інфраструктуру системи, щоб покращити її здатність обробляти одночасні сеанси користувачів, і має необхідні технічні навички та ресурси, щоб завершити зміни протягом заданого терміну.

Релевантність: ціль відповідає загальній масштабованості системи та взаємодії з користувачем, оскільки повільний час відповіді або помилки під час періодів високого трафіку можуть призвести до розчарування та зниження рівня використання.

Кожне завдання включає перелік зацікавлених сторін (стейкхолдерів), які, ймовірно, братимуть участь у виконанні цього завдання або на яких воно вплине.

Стейкхолдери (Stakeholders) або зацікавлені сторони – особи, які мають легітимний інтерес у проєкті чи організації, тобто певною мірою залежать від неї або можуть впливати на її діяльність.[3]

Відповідно до PMBOK® 7 аналіз зацікавлених сторін – метод систематичного збору та аналізу кількісної та якісної інформації, щоб визначити, чиї інтереси необхідно враховувати впродовж проєкту.

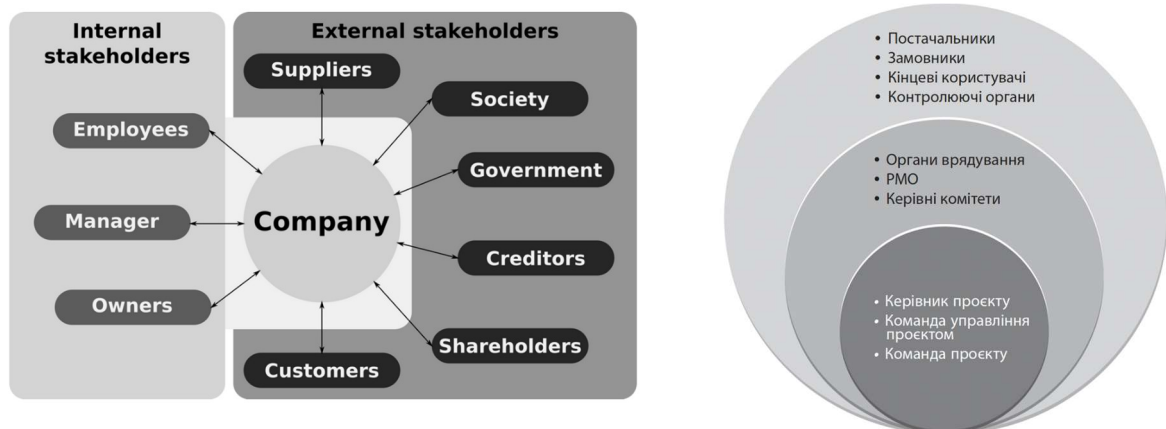


Рис. 2 – Приклади визначення стейкхолдерів [3]

BAВOK®v3 [4] у загальний перелік зацікавлених сторін включає наступні ролі: бізнес-аналітик (business analyst), замовник (customer), експерт предметної області (domain subject matter expert), кінцевий користувач (end user), експерт предметної області впровадження (implementation subject matter expert), операційна підтримка (operational support), керівник проєкту (project manager), регулятор (regulator), спонсор (sponsor), постачальник (supplier) і тестувальник (tester). Не обов'язково, щоб ці ролі були виконані для кожної конкретної ініціативи. Будь-яка зацікавлена сторона може бути джерелом вимог, припущень або обмежень.

1.2 Визначення вимог до розробки інформаційної системи

Вимоги – це властивості, якими має володіти ПЗ для адекватного визначення функцій, умов та обмежень виконання ПЗ, а також обсягів даних, технічного забезпечення та середовища функціонування. Насправді часто застосовується підхід, використовуваний у різних методологіях розробки, і базується на визначенні груп вимог до продукту. Такий підхід зазвичай включає групи (типи, категорії) вимог, наприклад: функціональні, нефункціональні (зокрема атрибути якості), обмеження тощо.

Специфікація вимог до програмного забезпечення (Software Requirements Specification SRS) — це документ, який описує функціональні та нефункціональні вимоги до програмної системи. Він надає детальний опис того, що програмне забезпечення має робити, як воно має поводитися та яким обмеженням воно має відповідати (рис. 2).

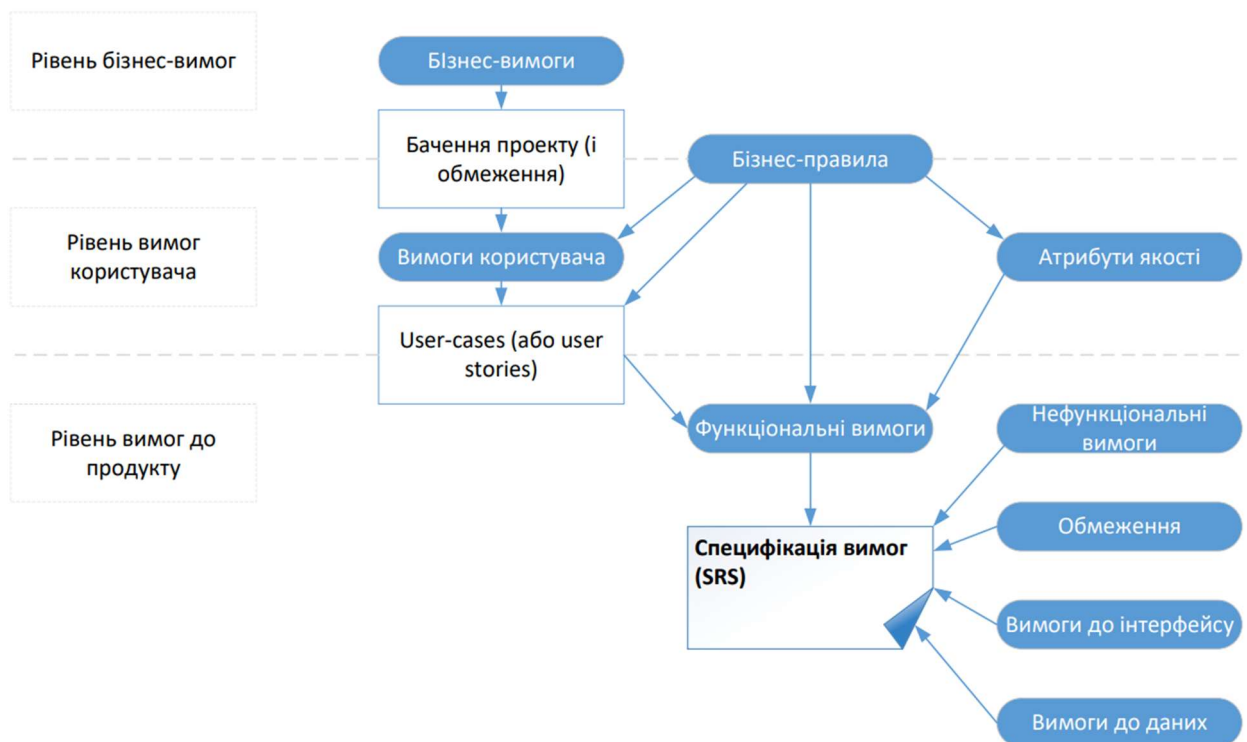


Рис. 2 – Типи вимог

SRS зазвичай містить опис загальної мети та цілей системи, а також перелік конкретних функціональних і нефункціональних вимог. *Функціональні* вимоги визначають, що має робити програмна система, тоді як *нефункціональні* вимоги визначають із врахуванням «яких умов», «наскільки добре» система повинна виконувати певні завдання (наприклад, час відгуку, масштабованість, надійність, безпека).

Вимоги користувачів (User Requirements) – описують цілі/завдання користувачів системи, які мають досягатися/виконуватися користувачами за допомогою створюваної програмної системи. *Обмеження* (Constraints) включають формулювання умов, що модифікують вимоги або набори вимог, звужуючи вибір можливих рішень щодо їх реалізації.

SRS також може включати діаграми або моделі, які ілюструють архітектуру системи, інтерфейс користувача або потік даних. Крім того, SRS має надавати чітке та недвозначне визначення всіх технічних термінів, акронімів та аббревіатур, які використовуються в документі.

SRS слугує основним довідковим документом протягом усього життєвого циклу розробки програмного забезпечення. Вона використовується для керівництва розробкою, впровадженням і тестуванням програмної системи, а також для забезпечення відповідності кінцевого продукту заданим вимогам. SRS також є важливим інструментом для спілкування із зацікавленими сторонами та управління очікуваннями від проекту, оскільки вона надає чіткий і вичерпний опис програмної системи, яка розробляється.

Вимоги до інтерфейсу стосуються специфікацій і критеріїв того, як різні частини системи або програмного забезпечення повинні спілкуватися або взаємодіяти одна з одною. Інтерфейси — це точки, де різні компоненти або модулі системи підключаються та

обмінюються інформацією, тому важливо визначити вимоги до того, як має відбуватися ця взаємодія.

Вимоги до інтерфейсу можуть включати такі деталі, як формат даних для обміну, протоколи або стандарти, які будуть використовуватися для зв'язку, частота або час взаємодії, заходи безпеки та контролю доступу, які мають бути реалізовані, а також процедури обробки помилок і відновлення.

Наприклад, у програмній системі, яка інтегрується зі сторонньою службою, вимоги до інтерфейсу можуть вказувати, що система повинна використовувати певний API для обміну даними зі службою, і що дані мають бути відформатовані певним чином. Вимоги також можуть визначати частоту обміну даними, максимальний час відповіді та заходи безпеки, які необхідно вжити для захисту конфіденційних даних.

Визначення чітких і детальних вимог до інтерфейсу є важливим для забезпечення ефективного та надійного обміну даними між різними компонентами системи, а також для мінімізації ризику помилок або несправностей, спричинених неправильним зв'язком або несумісністю інтерфейсів.

Вимоги до даних у SRS стосуються конкретних даних, які потрібні програмній системі для правильної роботи. Це може включати:

- **Вхідні дані:** це дані, які система очікує отримати від користувачів, інших систем або зовнішніх джерел. Приклади можуть включати інформацію про користувача, дані про продукт, фінансову інформацію або показання датчиків.
- **Вихідні дані:** це дані, які система генерує в результаті їх обробки. Приклади можуть включати звіти, сповіщення або оновлені записи даних.
- **Зберігання даних:** це дані, які система повинна зберігати або отримувати для належної роботи. Прикладами можуть бути профілі користувачів, каталоги продуктів або історії транзакцій.
- **Обробка даних:** це алгоритми або процедури, які система використовує для маніпулювання або перетворення даних. Приклади можуть включати фільтрацію, сортування або аналіз даних.

Кожна організація використовує власний варіант процесу розробки вимог, що залежить від певних факторів: досвіду роботи колективу розробників, типу системи, що розробляється, стандартів, що використовуються, тощо.

Процес формування та аналізу вимог проходить через низку етапів.

1.Аналіз предметної галузі. Аналітики повинні вивчити предметну галузь, де експлуатуватиметься система.

2.Збір вимог. Це процес взаємодії з особами, які формують вимоги. Під час цього процесу продовжується аналіз предметної галузі.

3.Класифікація вимог. На цьому етапі безформний набір вимог перетворюється на логічно пов'язані групи вимог.

4.Вирішення протиріч. Безперечно, вимоги численних осіб, зайнятих у процесі формування вимог, будуть суперечливими. На цьому етапі визначаються та вирішуються протиріччя різного роду.

5.Призначення пріоритетів. У будь-якому наборі вимог одні з них будуть важливішими за інші. На цьому етапі спільно з особами, які формують вимоги, визначаються найважливіші вимоги.

6.Перевірка вимог. На цьому етапі визначається їхня повнота, послідовність і несуперечність.

Процес формування та аналізу вимог циклічний, із зворотним зв'язком від одного етапу до іншого. Цикл починається з аналізу предметної галузі та закінчується перевіркою вимог. Розуміння вимог предметної галузі збільшується у кожному циклі процесу формування вимог.

Існує три основні підходи до формування вимог: метод, що ґрунтується на безлічі опорних точок зору, сценарії та етнографічний метод.

Розглянемо метод на основі опорних точок зору.

Підхід з використанням різних опорних точок зору до розробки вимог визнає різні (опорні) точки зору на проблему та використовує їх як основу побудови та організації, як процесу формування вимог, так і безпосередньо самих вимог.

Різні методи пропонують різні трактування виразу "погляд". Точки зору можна трактувати в такий спосіб.

1. *Як джерело інформації про системні дані.* У цьому випадку на основі опорних точок зору будується модель створення та використання даних у системі. У процесі формування вимог відбираються такі погляди (і визначаються дані), які будуть створені чи використані під час роботи системи, і навіть способи обробки цих даних. Методи SADT та CORE використовують цю інтерпретацію точок зору.

2. *Як структура представлень.* У цьому випадку погляди розглядаються як особлива частина моделі системи. Наприклад, з урахуванням різних точок зору можуть розроблятися моделі "сутність-зв'язок", моделі кінцевого автомата тощо.

3. *Як отримувачі системних послуг.* І тут погляди є зовнішніми (щодо системи) одержувачами системних сервісів. Позиції допомагають визначити дані, необхідні для виконання системних сервісів або їх управління.

Кожна з цих інтерпретацій точок зору має сильні та слабкі сторони. Опорні точки зору як джерела інформації про системні дані і як подання мають цінність для виявлення протиріч у вимогах. Але використовувати їх у процесі структурного аналізу вимог важко, оскільки тут не фіксуються зв'язки між точками зору та типами учасників формування вимог.

Інтерактивні системи постачають послуги кінцевим користувачам. Тому найефективнішим підходом до аналізу таких систем є використання зовнішніх опорних точок зору. Ці точки зору взаємодіють із системою, одержуючи від неї послуги і продукуючи дані та керуючі сигнали.

Цей підхід корисний створення нефункціональних вимог, із якими можна пов'язати будь-який сервіс. Різні точки зору, що багаторазово повторюються, дозволяють сформулювати для сервісів різні нефункціональні вимоги.

На основі цього підходу розроблено **метод VORD** (Viewpoint-Oriented Requirements Definition – визначення вимог на основі точок зору) для формування та аналізу вимог. Основні етапи методу VORD:

1. Ідентифікація точок зору, які отримують системні послуги, та ідентифікація сервісів, що відповідають кожній точці зору.
2. Структурування точок зору - створення ієрархії згрупованих точок зору. Загальносистемні послуги надаються вищим рівням ієрархії та успадковуються точками зору нижчого рівня.

3. Документування опорних точок зору, що полягає у точному описі ідентифікованих точок зору та сервісів.
4. Відображення системи точок зору, що показує системні об'єкти, визначені на основі інформації, яка міститься в опорних точках зору.

Розглянемо **приклад** використання методу VORD на перших трьох кроках аналізу вимог для системи підтримки замовлення та обліку товарів у продуктовій крамниці. У продуктовій крамниці для кожного товару фіксується місце зберігання (певна полиця), кількість товару та його постачальник. Система підтримки замовлення та обліку товарів повинна забезпечувати додавання інформації про новий товар, зміну або видалення інформації про наявний товар, зберігання (додавання, зміну та видалення) інформації про постачальників, що включає назву фірми, її адресу та телефон. За допомогою системи складаються замовлення постачальникам. Кожне замовлення може містити декілька позицій, у кожній позиції вказуються найменування товару та його кількість у замовленні. Система на вимогу користувача формує та видає на друк наступну довідкову інформацію:

- список усіх товарів;
- список наявних товарів;
- список товарів, кількість яких потрібно поповнити;
- список товарів, які постачаються даним постачальником.

Першим кроком у формуванні вимог є ідентифікація опорних точок зору. У всіх методах формування вимог, заснованих на використанні точок зору, початкова ідентифікація є найважчою задачею. Один із підходів до ідентифікації точок зору — метод "мозкового штурму", коли визначаються потенційні системні сервіси та організації, що взаємодіють із системою.

Організовується зустріч осіб, які беруть участь у формуванні вимог, які пропонують свої погляди. Ці точки зору подаються у вигляді діаграми, що складається з низки кругових областей, що відображають можливі точки зору (рис. 3). Під час "мозкового штурму" необхідно ідентифікувати: потенційні опорні точки зору, системні сервіси, вхідні дані, нефункціональні вимоги.



Рис. 3. Діаграма ідентифікації точок зору

Наступною стадією процесу формування вимог буде ідентифікація опорних точок зору (на рис. 3 показані у вигляді овалів без фону) та сервісів (показані у вигляді затінених областей). Сервіси повинні відповідати опорним точкам зору. Але можуть бути послуги, які

не поставлені їм у відповідність. Це означає, що на початковому етапі "мозкового штурму" деякі опорні точки зору не були ідентифіковані.

У таблиці 1 показаний розподіл сервісів для деяких ідентифікованих на рис.3 точок зору. Той самий сервіс може бути співвіднесений з декількома точками зору.

Таблиця 1 - Сервіси, співвіднесені з точками зору

| Клієнт | Покупець | Постійний покупець | Товар | Постачальник | Продавець | Адміністратор |
|---|--|--|--|--|-------------------------------|-------------------------------------|
| Перевірка наявності товару | Занесення до списку постійних клієнтів | Отримання знижки | Прийом товару | Занесення у базу даних (назва, телефон, адреса тощо) | Продаж товару | Доступ до бази даних |
| Покупка товару | | Отримання інформації про нові постачання | Занесення до бази даних (постачальника, кількості, місця зберігання, тощо) | | Друк чека | Перевірка статистики |
| Отримання чека | | | | | Доступ до каталогу | Перевизначення ціни |
| Замовлення товару | | | | | Перевірка наявності товару | Формування замовлення постачальнику |
| Занесення покупця та суми покупки до бази даних | | | | | Оформлення замовлення покупця | Друк замовлення |

Інформація, витягнута з точок зору, використовується для заповнення форм шаблонів точок зору та організації точок зору в ієрархію спадкування. Це дозволяє побачити загальні точки зору та повторно використовувати інформацію в ієрархії спадкування. Сервіси, дані та керуюча інформація наслідуються підмножиною точок зору. На рис. 4 показано частину ієрархії точок зору для системи підтримки замовлення та обліку товарів.

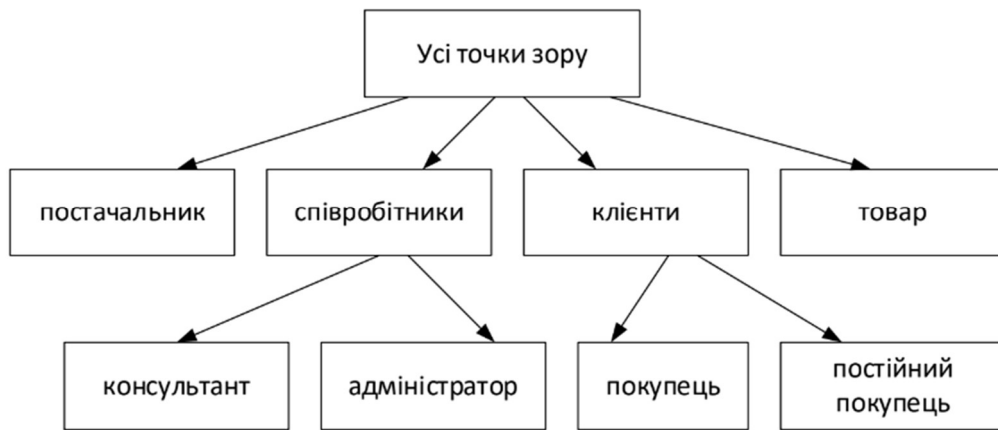


Рис.4 – Ієрархія точок зору

З отриманих моделей будуються вимоги користувачів, тобто, опис природною мовою функції, що виконуються системою, та обмежень, що накладаються на неї. Вимоги користувачів повинні описувати зовнішню поведінку системи, основні функції та послуги, що надаються системою, її нефункціональні властивості. Необхідно виділити опорні точки зору та згрупувати вимоги відповідно до них. Користувацькі вимоги можна оформити як простим перерахуванням, так і використовуючи нотацію варіантів використання use case.

2. ПОРЯДОК ВИКОНАННЯ РОБОТИ

Заняття 1:

1. Розподілитися по групах по 3 студенти у кожній. Список груп подається викладачу.
2. Ознайомитись з теоретичним матеріалом розділу «1.1 Основні визначення бізнес-архітектури».
3. Разом у команді визначити предметну область для розробки інформаційної системи.
4. Визначити бізнес-драйвери та бізнес-цілі, які відповідають створенню обраної інформаційної системи.
5. Визначити зовнішніх стейкхолдерів та їх зацікавленість у проєкті.

Заняття 2:

1. Ознайомитись з теоретичним матеріалом підрозділу «1.2 Визначення вимог до розробки інформаційної системи».
2. Побудувати опорні точки зору на основі методу VORD для формування та аналізу вимог. Результатом мають бути дві діаграми: діаграма ідентифікації точок зору та діаграма ієрархії точок зору.
3. Сформувані функціональні та нефункціональні вимоги і обмеження.
4. До кінця заняття 2 у викладача має бути список групи з розподілом ролей (студент 1, студент 2, студент 3 – дивись наступний пункт) за яку частку презентації результатів відповідатиме кожний студент.

Заняття 3 та 4: презентація результатів виконання роботи по командах.

Команда разом представляє загальну ідею та призначення інформаційної системи.

Далі представляються отримані результати, при цьому кожен студент відповідає за презентацію певної частини результатів:

1. Студент 1: Презентація бізнес-драйверів створення обраної інформаційної системи.
2. Студент 2: Презентація бізнес-цілей створення обраної інформаційної системи.
3. Студент 3: Презентація зовнішніх стейкхолдерів та визначення їхньої зацікавленості у проєкті.
4. Студент 1: Презентація функціональних вимог до розробки інформаційної системи.
5. Студент 2: Презентація нефункціональних вимог до розробки інформаційної системи.
6. Студент 3: Презентація обмежень до розробки інформаційної системи.

У звіті окрім вищенаведених результатів також має бути наведена діаграма ідентифікації точок зору та діаграма ієрархії точок зору.

3. Джерела:

1. Матеріали лекцій
2. <https://www.techopedia.com/definition/28013/business-driver>
3. <https://www.maxzosim.com/stakeholders/>
4. https://ieaf.ir/Public_Resource/Learn/BABOK_Guide_v3_Member.pdf