

# ***ХАРАКТЕРИСТИКИ ВИМОГ. ІДЕНТИФІКАЦІЯ ТА КЛАСИФІКАЦІЯ ВИМОГ***

---

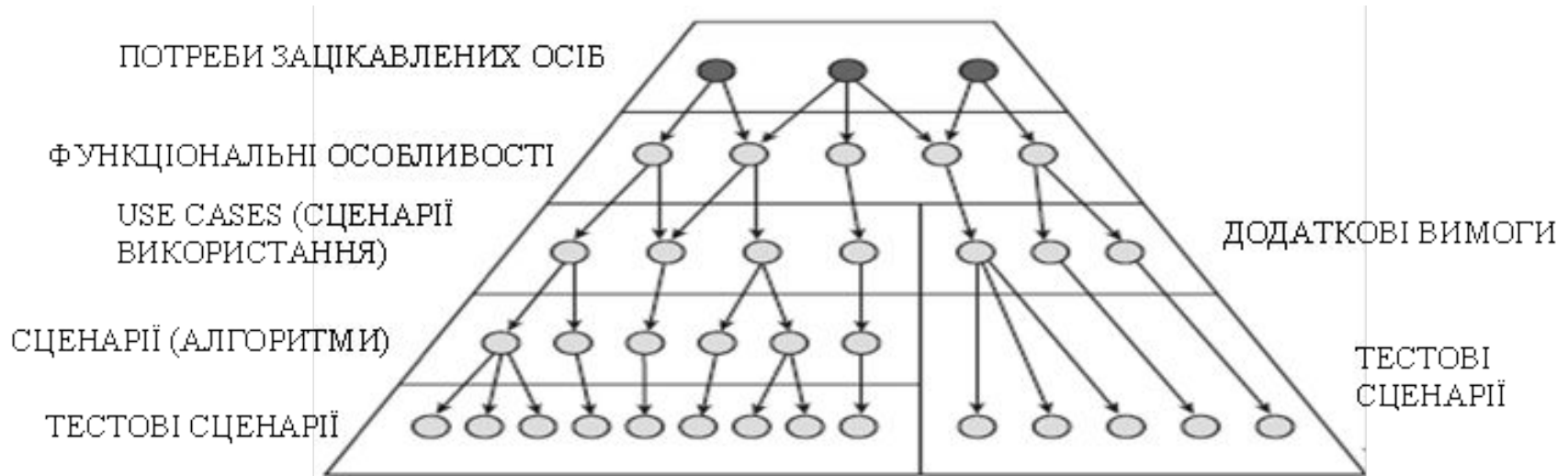
***Лекція 3.  
Ч.3.2***

***Розділ 2. Тема 2.2***

# Основні питання:

- Піраміда вимог
- Характеристики відмінної вимоги
- Ідентифікація, класифікація та організація вимог

# Піраміда вимог



# Характеристики відмінної ВИМОГИ

- ✓ Недвужначність
- ✓ Тестованість  
(можливість перевірки)
- ✓ Ясність (короткість, стислість, простота, точність)
- ✓ Коректність
- ✓ Зрозумілість
- ✓ Правдоподібність (реальність, виконуваність)
- ✓ Незалежність
- ✓ Елементарність
- ✓ Необхідність
- ✓ Незалежність від реалізації  
(абстрактність)

Крім цих критеріїв для окремих вимог і для набору вимог застосовуються три критерії. *Вимоги повинні бути:*

- ✓ Постійними.
- ✓ Небагатослівними.
- ✓ Завершеними.

# Недвужначність

Повинен існувати тільки один шлях інтерпретації вимоги. Іноді двозначність проявляється у вигляді невизначених акронімів.

## Наприклад 1:

**Вимога 1:** Система повинна бути реалізована з використанням ASP. Що значить ASP - Active Server Pages або Application Service Provider? Для вирішення цієї ситуації, нкжно вказати повне найменування та надати акронім в дужках:

**Вимога 1:** Система повинна бути реалізована з використанням Active Server Pages (ASP).

Наприклад 2:

*Вимога 1: Система не повинна приймати паролі більше 15-ти символів.*

Не зовсім ясно, що система повинна робити: Система не повинна дозволяти користувачеві вводити більш ніж 15 символів. Система повинна відсікати введений рядок до 15-ти символів. Система не повинна відображати повідомлення про помилку, якщо користувач вводить більш ніж 15 символів. виправлена вимога містить пояснення:

*Вимога 1: Система не повинна приймати паролі більше 15-ти символів. Якщо користувач вводить більш 15-ти символів при виборі пароля, повідомлення про помилку повинно інформувати користувача про необхідний виправленні пароля.*

# Тестованість (можливість перевірки)

Використання деяких слів може зробити вимогу неможливою для тестування:

*Деякі прикметники:* стійкий, безпечний, точний, ефективний, доцільний, гнучкий, що настроюється, надійний, доброзичливий, адекватний.

*Деякі прислівники і фрази з ними:* швидко, безпечно, своєчасно.

*Неспецифічні слова або акроніми:* і т.д., та / або, TBD (to be determined).

Такі вимоги можуть виглядати приблизно так:

**Вимога 1:** Функція пошуку повинна дозволяти користувачеві шукати замовлення на основі Прізвища, Імені, Дати, і т.д.

У цій вимозі всі критерії пошуку повинні бути детально перераховані. Дизайнер і розробник не можуть здогадатися, що користувач мав на увазі під скороченням «тощо».

# Ясність (короткість, стислість, простота, точність)

Вимоги не повинні містити непотрібних висловів чи інформації. Вони повинні бути викладені стисло і просто:

***Вимога 1:** Іноді користувач буде вводити Airport Code (Код Аеропорту), який система буде розпізнавати. Але іноді код можна замінити довколишнім містом, і тоді користувачеві не потрібно знати код аеропорту, так як система буде розуміти назву міста.*

Ця пропозиція може бути замінено на більш просте:

***Вимога 1:** Система повинна ідентифікувати аеропорт на підставі Airport Code (Кода Аеропорту) або City Name (Назви Міста).*



# *Корректність*

Якщо вимога містить факти, ці факти повинні бути достовірні:

*Вимога 1: Ціни на оренду машин повинні включати всі відповідні податки (включаючи податок штату - 6%)*

Податок залежить від штату, так що представлена цифра в 6% - некоректна.

# ***Зрозумілість***

Вимоги повинні бути граматично правильні, написані у відповідному стилі. Мають бути використані стандартні угоди. Слово «повинен» має бути використано замість «буде», «зобов'язаний» або «може».

# Правдоподібність (реальність, виконуваність)

Вимога повинна бути здійснено в рамках існуючих обмежень, таких як час, гроші та доступні ресурси.

*Вимога 1: Система повинна мати інтерфейс природною мовою, який буде розуміти команди англійською мовою.*

Ця вимога може бути нереальним через короткий період часу розробки.

# Незалежність

Щоб зрозуміти вимогу, не потрібно знати якусь іншу вимогу.

*Вимога 1: Список доступних рейсів повинен включати номер рейсу, час відправлення і час прибуття для кожного відрізка шляху.*

*Вимога 2: Він повинен бути відсортований за цінами.*

Слово «Він» у другому реченні відноситься до попередньої вимоги. І якщо порядок вимог змінити, ця вимога буде незрозумілою.

# Елементарність

Вимога повинна містити окремий трасований елемент (для якого можливе відстеження зв'язку):

*Вимога 1: Система повинна надавати можливість бронювати рейс, купувати квиток, бронювати номер в готелі, бронювати машину, а також надавати інформацію про розваги.*

Ця вимога містить п'ять елементарних вимог, які ускладнюють трасування. Пропозиції, що включають прийменники «і» або «але» мають бути переглянуті на предмет поділу їх на елементарні вимоги.

# Необхідність

У вимозі немає необхідності, якщо: жодній зацікавленій особі вимога не потрібна або видалення вимоги не вплине на систему.

***Вимога 1:** Всі вимоги, зазначені в документі Концепції мають бути реалізовані і протестовані.*

# Незалежність від реалізації (абстрактність)

Вимоги не повинні містити непотрібної інформації про дизайн і реалізації системи:

*Вимога 1: Інформація повинна зберігатися в текстовому файлі. Рішення того, яким чином інформація буде зберігатися, і потім передаватися користувачеві, має прийматися дизайнерами або архітекторами системи.*

# Постійність

Не повинно існувати конфліктів між вимогами. Конфлікти можуть бути прямі і непрямі. Прямі конфлікти виникають, коли очікується різна поведінка системи в одній і тій же ситуації:

*Вимога 1: Дата повинна відображатися у форматі мм/дд/рррр.*

*Вимога 1: Дата повинна відображатися у форматі дд/мм/рррр.*



# Небагатослівність

Кожна вимога має бути позначена тільки один раз, і не повинно перекриватися іншою вимогою:

***Вимога 1:** Для зручності введення дати рейсу в системі повинен бути доступний календар.*

***Вимога 2:** Система повинна відображати спливаюче вікно з календарем при введенні будь-якої дати.*

Перша вимога (що відноситься тільки до дати рейсу) є частиною другої вимоги (що відноситься до будь-яку дату, введеної користувачем).

# Завершеність

Вимога повинна бути описано для всіх можливих умов:

***Вимога 1:** Країна призначення не повинна відобразитися для рейсів в межах США.*

***Вимога 2:** Для рейсів через море система повинна відобразити країну призначення.*

Гарна вимога повинна задовольняти якомога більшій кількості критеріїв. Однак вони можуть бути виражені у вигляді комбінації перерахованих вище критеріїв:

***Змінюваний:** Якщо вимога елементарна і небагатослівна, то воно зазвичай змінювана.*

***Трасуємий:** Якщо воно коротке і має унікальний ідентифікатор (ID), то воно зазвичай трасуємий (здатне до відстеження зв'язку).*

# Основні питання управління

*Воно пов'язане з трьома основними питаннями:*

1. Ідентифікація, класифікація, організація та документування вимог.
2. Зміна вимог, тобто формулювання, узгодження, перевірка достовірності та документування неминучих уточнень.
3. Трасування вимог, тобто встановлення залежності між вимогами, з одного боку, і рештою системними артефактами - з іншого, а також між самими вимогами.

# ***Вимоги - ідентифікація та класифікація***

*Вимоги описуються природньою мовою, наприклад, наступним чином:*

«Система повинна запланувати наступний телефонний дзвінок клієнтові за запитом оператора».

«Система повинна автоматично набирати запланований телефонний номер і одночасно відобразити на екрані оператора інформацію, що включає телефонний номер, номер клієнта, ім'я абонента».

«У разі успішного з'єднання система повинна відобразити вступний текст, з яким оператор може звернутися до абонента для того, щоб зав'язати розмову».

Типова система може складатися із сотень або тисяч формулювань вимог, подібних тим, які наведені вище. Для належного управління такою величезною кількістю вимог їх необхідно пронумерувати за допомогою деякої схеми *ідентифікації*. Ця схема може мати на увазі класифікацію вимог по групах, які легше піддаються управлінню.

# ***Методи ідентифікації і класифікації вимог***

*Існує декілька методів ідентифікації і класифікації вимог:*

1. Унікальний ідентифікатор
2. Порядковий номер в ієрархії документів
3. Порядковий номер у категорії вимог

## Унікальний ідентифікатор

*Унікальний ідентифікатор* - це, як правило, порядковий номер, присвоєний вручну або згенерований з використанням бази даних CASE - інструментів, тобто бази даних (або репозиторію) в якій CASE - інструмент зберігає артефакти, створені в результаті аналізу чи проектування.

Порядковий номер в ієрархії документів

*Порядковий номер в ієрархії документів* - номер, присвоєний з урахуванням місця вимоги в технічному завданні.



## Порядковий номер у категорії вимог

*Порядковий номер у категорії вимог* - номер, присвоєний на додаток до мнемонічному імені, яке визначає категорію вимог (де під підкатегорією вимог розуміють функціональні вимоги, вимоги до даних, вимоги до продуктивності, вимоги до безпеки і т.д.).

## *Переваги та недоліки*

*Кожен з методів ідентифікації володіє своїми перевагами і недоліками.*

Найбільшою гнучкістю і захищеністю від помилок відрізняють *унікальні ідентифікатори*, які генеруються за допомогою бази даних. Бази даних володіють вбудованими можливостями генерації унікальних ідентифікаторів для кожного нового запису даних в умовах одночасного доступу до даних з боку багатьох користувачів.

Деякі бази даних здатні додатково підтримувати супровід декількох версій однієї і тієї ж записи за рахунок розширення значення унікального ідентифікатора за допомогою номера версії. Нарешті, бази даних можуть володіти можливостями підтримки цілісності на рівні посилань між артефактами моделювання, включаючи вимоги, і можуть, таким чином, забезпечити необхідну підтримку змін і трасування вимог.

# Ієрархія вимог

Вимоги можна упорядкувати у вигляді *ієрархічно впорядкованої структури*, що представляє відносини батько-потік.

Батьківська вимога складена з дочірніх вимог, дочірня вимога, по суті, являє собою частину батьківського вимоги.

Ієрархічні відносини дозволяють ввести додатковий рівень класифікації вимог. Іноді цей факт відображається в ідентифікаційному номері.

*Наприклад*, вимога, пронумерована як 4.9, може бути дев'ятим «нащадком» «батька» з ідентифікаційним номером, рівним 4.

Наведений нижче набір формулювань являє собою ієрархічно впорядковані вимоги.

1. Система повинна запланувати наступний телефонний дзвінок клієнтові за запитом оператора.

1.1. Система повинна активувати клавішу Next Call (Наступний дзвінок) при відкритті форми Telemarketing Control (Керування телемаркетингом) або по завершенні попереднього виклику.

1.2. Система повинна видалити дзвінок з початку черги запланованих дзвінків та надати йому статус поточного дзвінка.

1.3. І так далі.

**Ієрархії вимог** дозволяють визначати їх на різних рівнях абстракції. Це узгоджується з загальним принципом моделювання, відповідно до якого при переході до більш низького рівня абстракції моделі систематично збагачуються все новими деталями. В результаті для батьківських вимог можна сконструювати узагальнені вимоги, а деталізовані моделі можна пов'язати з дочірніми вимогами.