

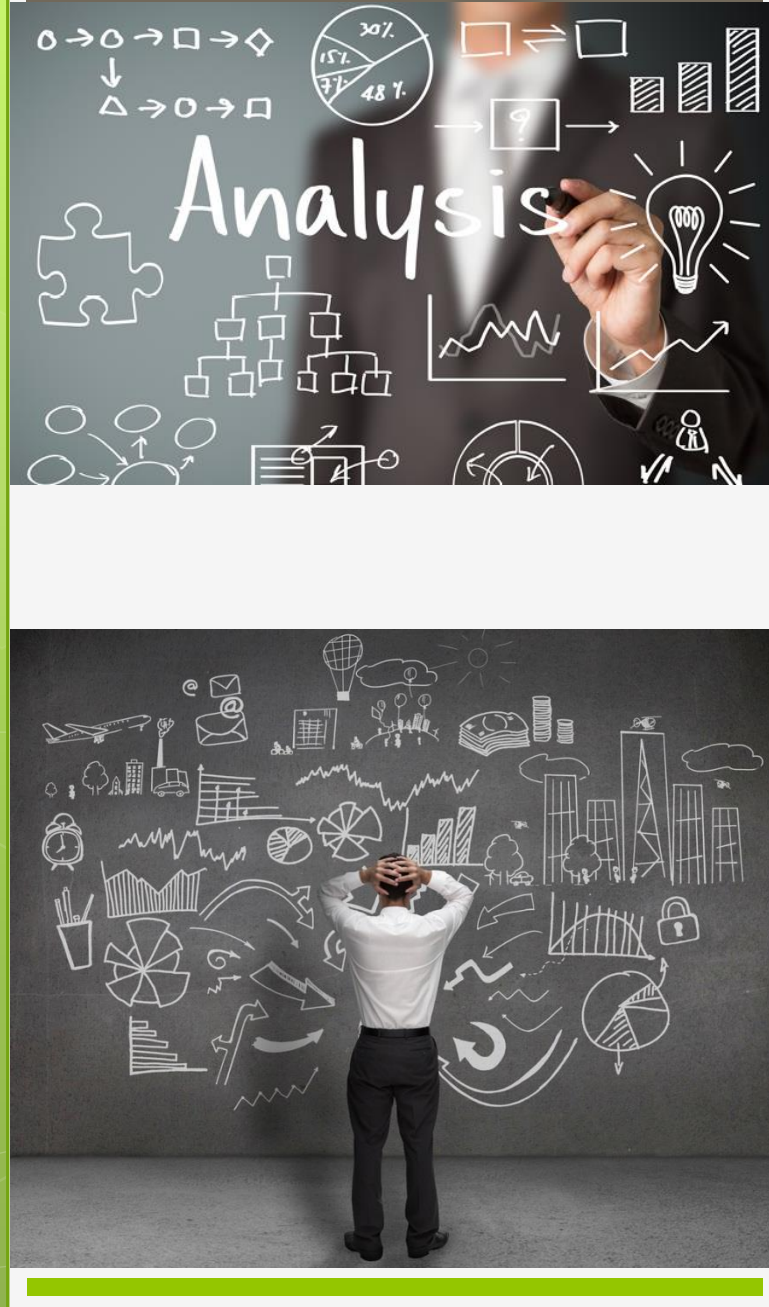
# Аналіз вимог до програмного забезпечення

Відповідальний за курс

к.т.н., доц.

Каплієнко

Тетяна Ігорівна



Who are we?



Clients!



What do we  
want?



We don't know!



When do we want  
it?



NOW!



**ТИ МОЖЕШ БУТИ  
БЕЗКІНЕЧНО ПРАВИМ.**

**ПРОТЕ ЯКИЙ У ЦЬОМУ  
СЕНС, ЯКЩО ТЗ БУЛО  
СКЛАДЕНО УСНО**

# Еволюція уяви про вимоги



How the customer explained it



How the Project Leader understood it



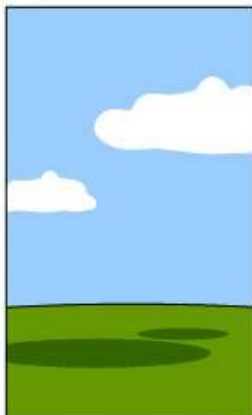
How the Analyst designed it



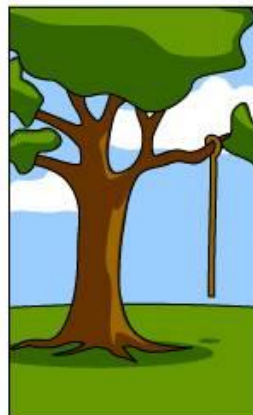
How the Programmer wrote it



How the Business Consultant described it



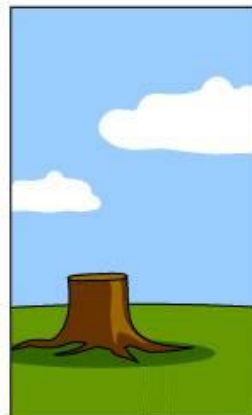
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed





**The Client wanted a more realistic Fire.**



**I gave him some !!!**

# ЖИТТЄВИЙ ЦИКЛ ПЗ

- Концептуалізація системи;
- Специфікація вимог;
- Проєктування архітектури системи;
- Проєктування класів;
- Реалізація;
- Інтеграція;
- Супровід.

# Підходи до розробки ПЗ

- Структурний підхід:
  - діаграми потоків даних (*data flow diagrams*) для моделювання процесів
  - діаграми сутність-зв'язок (*entity relationship diagrams*) для моделювання даних
- Об'єктно-орієнтований підхід



# Способи пошуку концепції НОВИХ СИСТЕМ

- Нова функціональність
- Модернізація
- Спрощення
- Автоматизація ручних процесів
- Інтеграція
- Аналогії
- Глобалізація



# Концепція системи

- Для кого призначений застосунок?
- Яке завдання буде вирішувати застосунок?
- Де буде використовуватися система?
- Коли буде вимагатися система:
- Чому потрібна система
- Як буде працювати система?

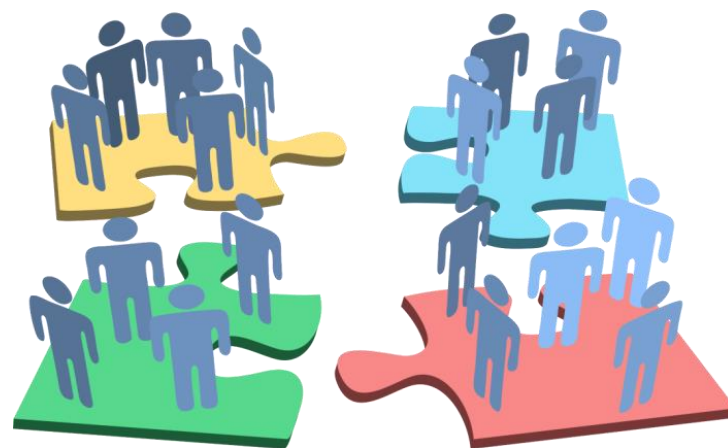


# Класи користувачів

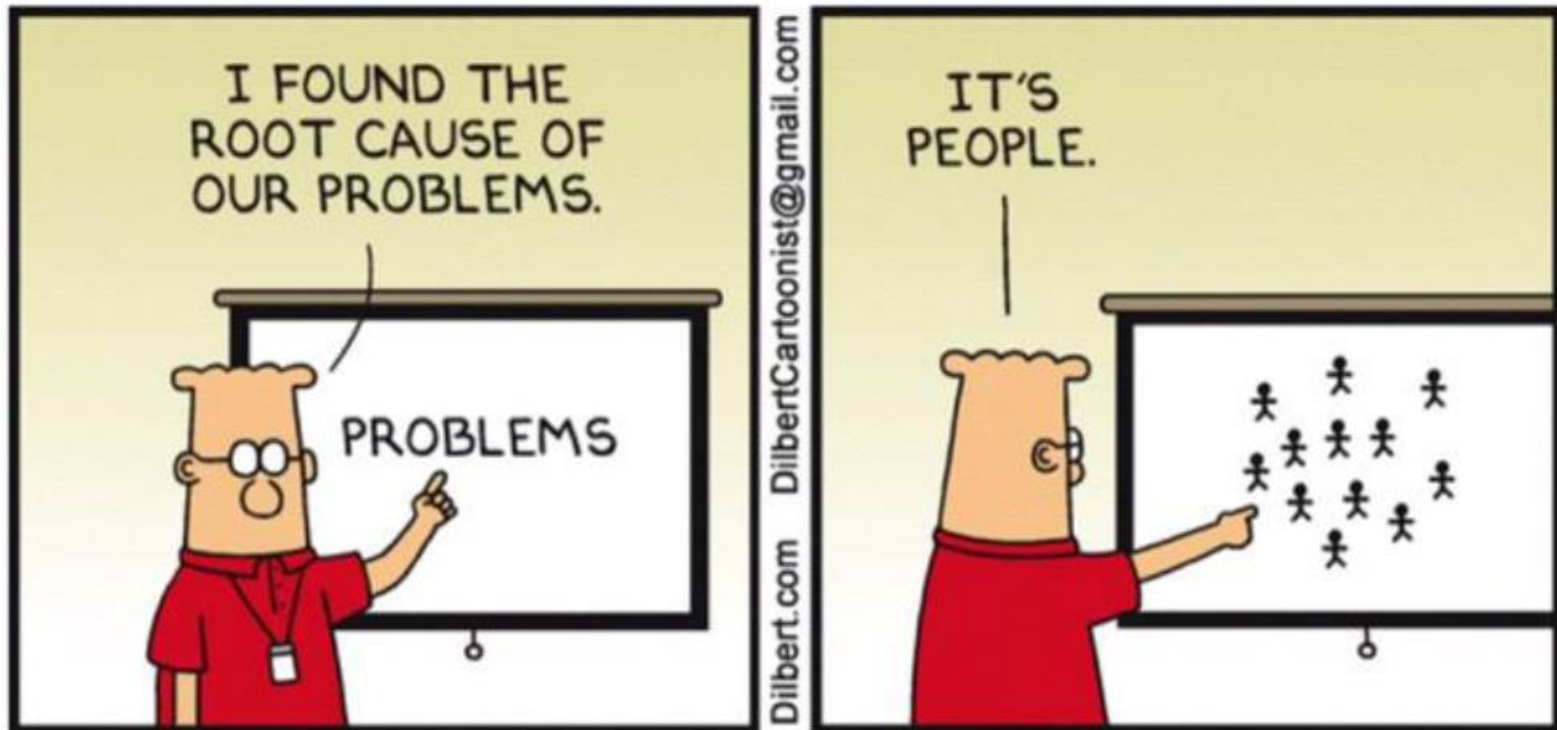
- За частотою використання продукту
- За досвідом в предметній області
- За функціональністю, яка потребується
- За задачами, що вирішуються
- За правами доступу до системи

# Зацікавлені особи

- Замовники
- Користувачі
- Аналітики вимог
- Розробники
- Тестувальники
- Технічні письменники
- Менеджер проєкту
- Співробітники юридичного відділу
- Представники промислових організацій
- Співробітники відділу продажів



# Складність процесу розроблення та керування вимогами



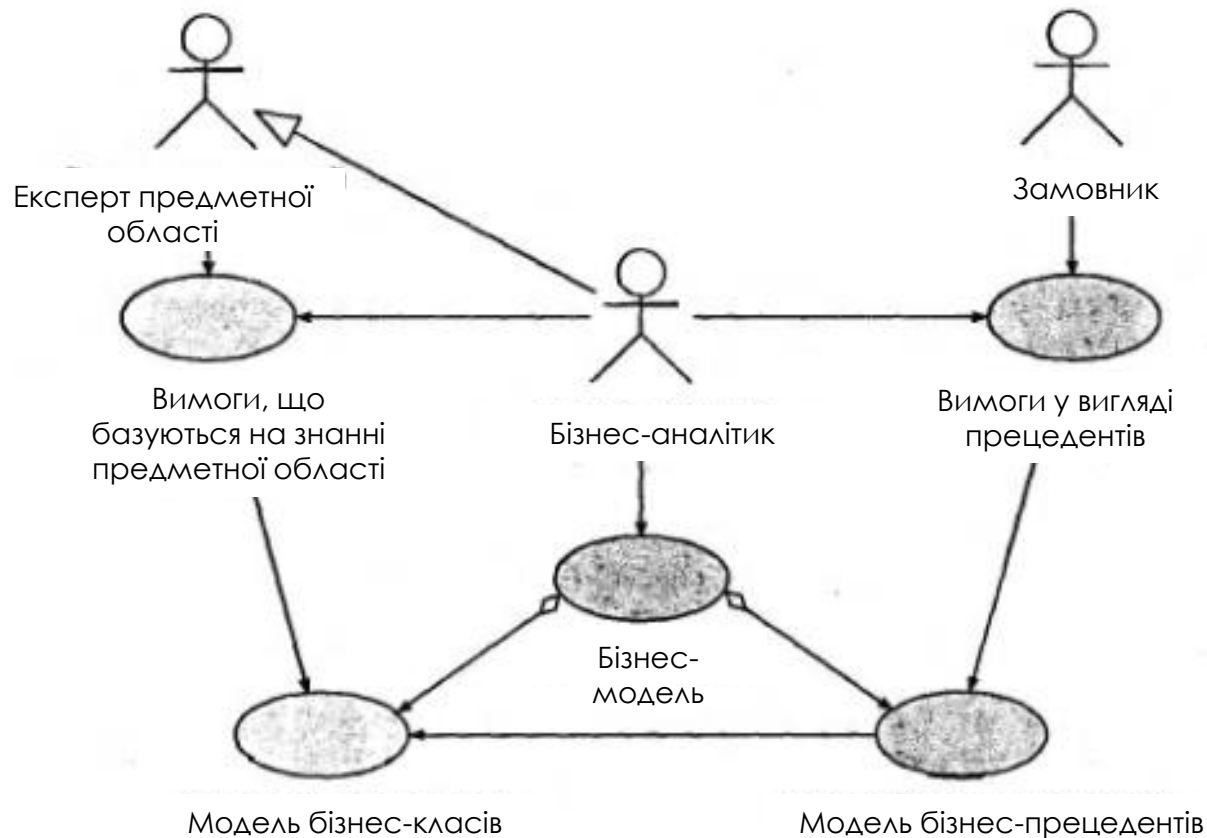


# Традиційні методи виявлення вимог

- Інтерв'ювання замовників і експертів в проблемної області
- Анкетування
- Спостереження
- Вивчення документів і програмних систем



# Взаємний вплив моделей, характерний для процесу виявлення вимог



# Типи інтерв'ю



- Неструктуроване інтерв'ю
- Структуроване інтерв'ю
  - Питання з відкритою множиною відповідей
  - Питання з замкнутою множиною відповідей

## Необхідно уникати:

! Особистих питань

«Чи повинні ми працювати так, як ми працюємо?»

! Упереджених питань

«Адже Ви не станете цього робити, чи не так?»

! Навідних питань

«Ви ж зробите саме так, вірно?»

# Анкетування

- Багатоальтернативні питання
- Рейтингові питання
- Питання з ранжуванням



# Спостереження

- Пасивне
- Активне





# Вивчення документів і програмних систем

## Організаційні документи:

1. Форми ділових документів (по можливості - заповнені)
2. Опис робочих процедур
3. Посадові обов'язки
4. Методичні посібники
5. Бізнес-плани
6. Схеми організаційних структур
7. Внутрішній кореспонденція
8. Протоколи нарад
9. Зовнішня кореспонденція
10. Нарікання клієнтів

Організаційні документи

## Системні форми і звіти:

1. Системні моделі аналізу і проєктування
2. Звіти разом з документацією
3. Системні керівництва по експлуатації
4. Призначена для користувача документація
5. Технічна документація
6. Копії екранів

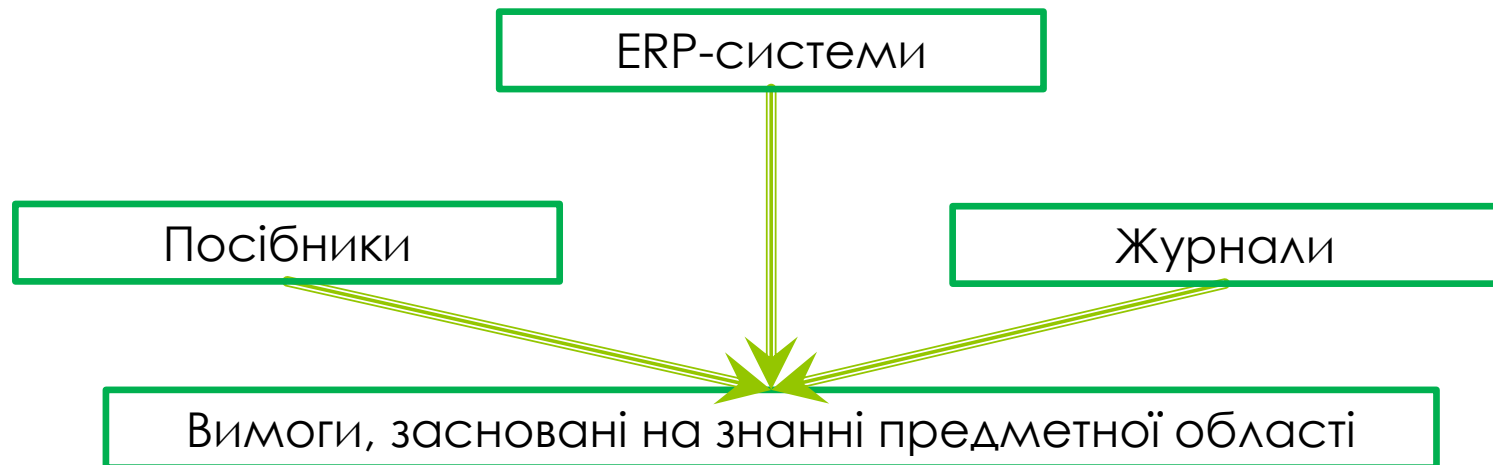
Системні форми і звіти

Вимоги, сформульовані у вигляді прецедентів, або прецедентні вимоги

# Вивчення документів і програмних систем

## Документи з предметної області:

1. Посібники
2. Журнали
3. ERP-системи



## Сучасні методи виявлення вимог

- Використання програмних прототипів
- JAD (Joint Application Development) – спільна розробка застосунків
- RAD (Rapid Application Development) – швидка розробка застосунків

# Візуальні моделі

- Технічне завдання
- Каркасна уявлення
- Прототипування



# Прототипування

- Одноразовий прототип (throw-away prototype)
- Еволюційний прототип (evolutionary prototype)





# Спільна розробка застосунків (JAD-метод)

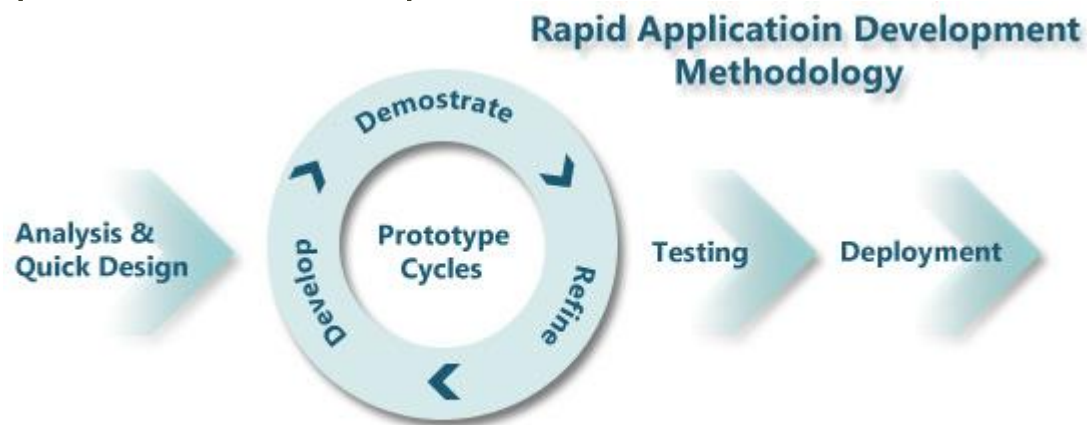
Групова динаміка (не більше 25-30 чоловік)

Круг осіб учасників:

- Ведучий
- Секретар
- Замовники
- Розробники

# Швидка розробка застосунків (RAD-метод)

- Еволюційне протипування
- CASE-засоби
- Спеціалісти, які володіють розвинутими інструментальними засобами
- Інтерактивний JAD-метод
- Жорсткі часові рамки

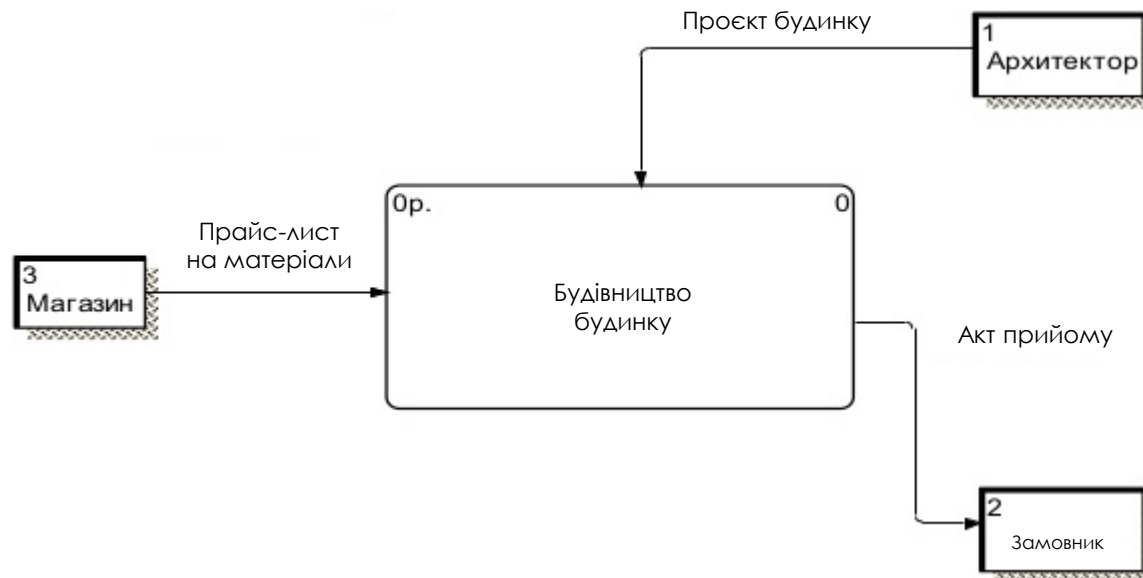


# Узгодження і перевірка обґрунтованості вимог

- Вимоги, що виходять за рамки проєкту
- Матриця залежності вимог
- Ризики та пріоритети вимог

# Вимоги, що виходять за рамки проєкту

- Основний метод аналізу - контекстна діаграма методу потоків даних DFD (*Data Flow Diagrams*)



# Матриця залежності вимог

Вимога	B1	B2	B3	B4
<b>B1</b>	X	X	X	X
<b>B2</b>	конфлікт	X	X	X
<b>B3</b>			X	X
<b>B4</b>		перекриття	перекриття	X



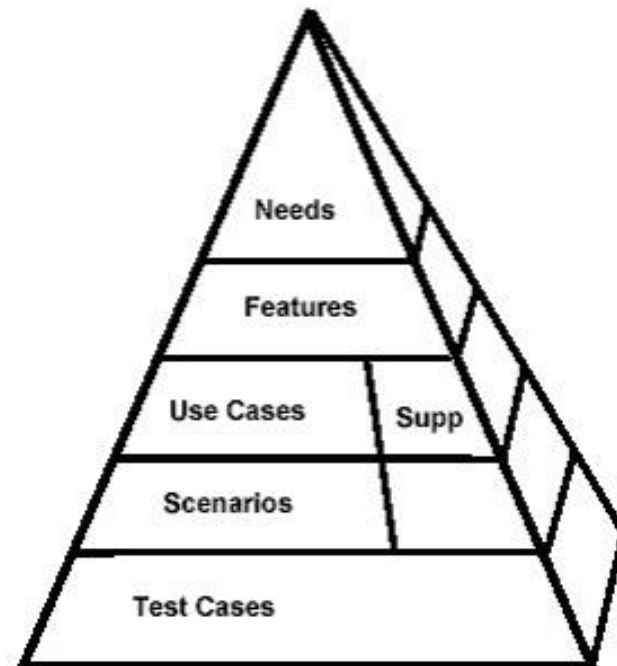
# РИЗИКИ ВИМОГ



1. Технічний ризик
2. Ризик, пов'язаний зі зниженням продуктивності
3. Ризик, пов'язаний з порушенням безпеки
4. Ризик, пов'язаний з процесом розробки
5. Ризик, пов'язаний з порушенням цілісності баз даних
6. Політичний ризик
7. Ризик, пов'язаний з порушенням законності
8. Ризик, пов'язаний з мінливістю

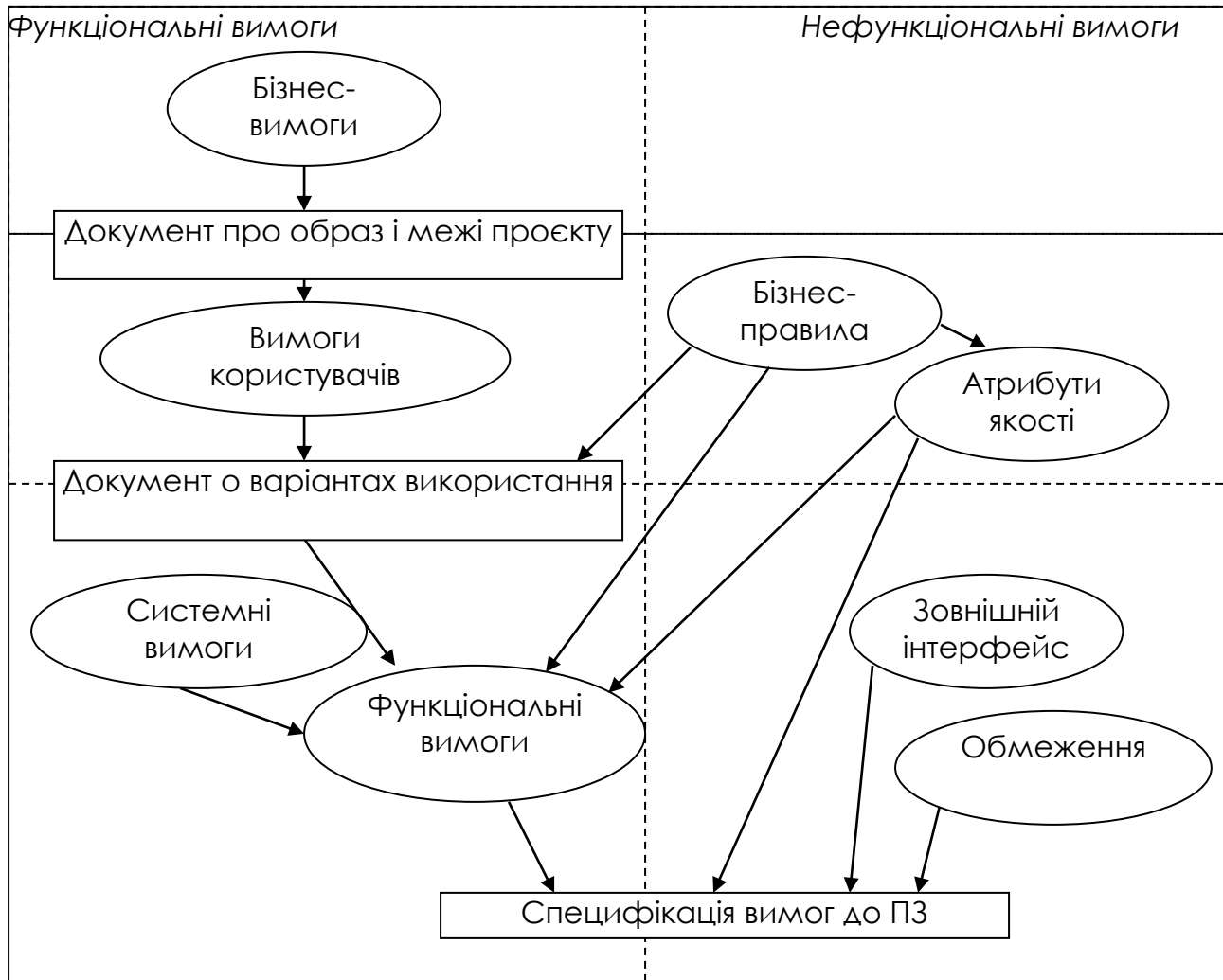
# Рівні вимог

- Бізнес-вимоги
- Функціональні вимоги
- Нефункціональні вимоги



# Класифікація вимог

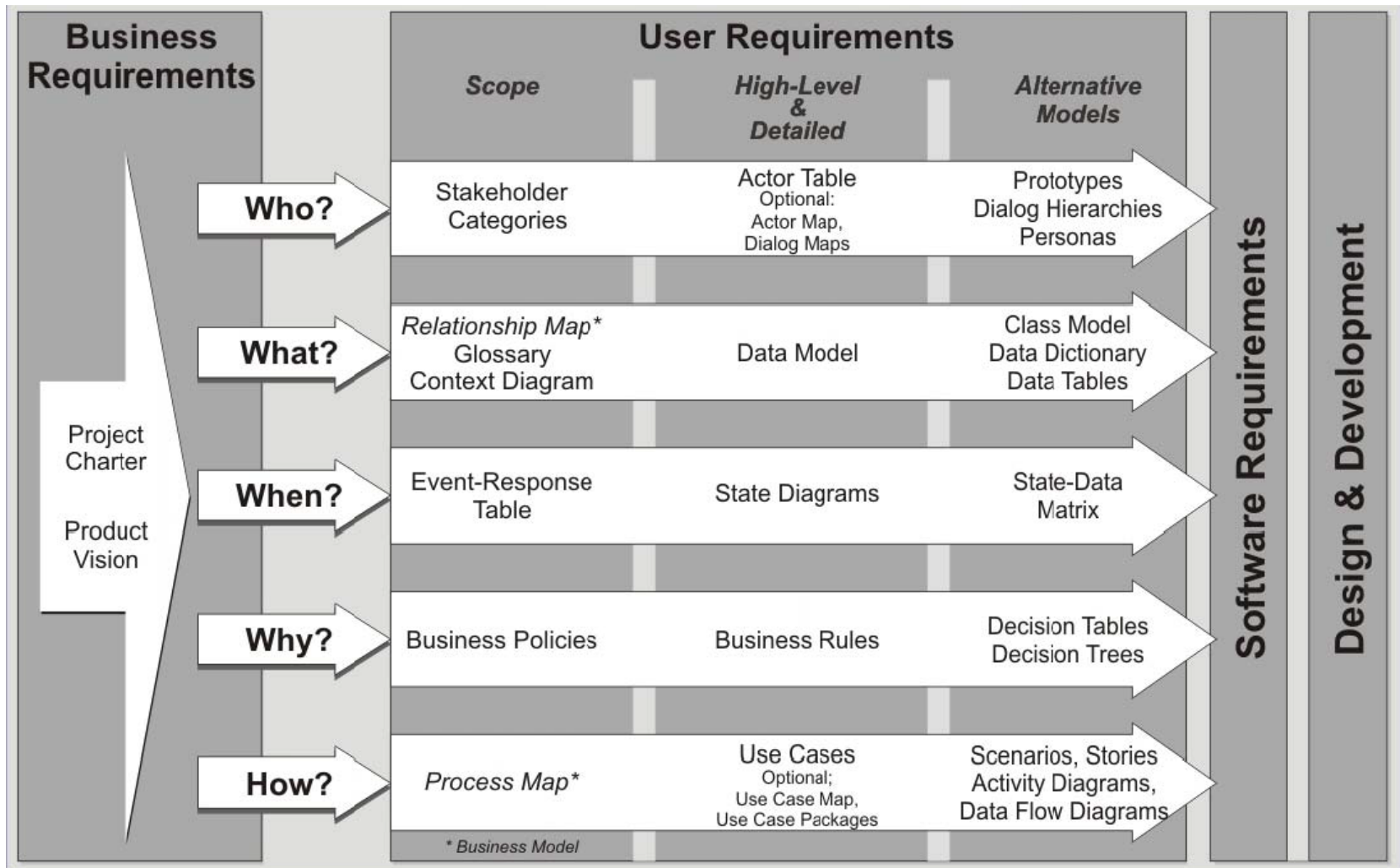
**К.ВИГЕРС**



# ЯКИХ ВИМОГ НЕ ПОВИННО БУТИ!

- Деталей дизайну або реалізації
- Даних о плануванні проєкту
- Відомостей про тестування





# Розробка та керування вимогами



# Розробка вимог

- Ідентифікація класів користувачів для даного продукту;
- З'ясування потреб тих, хто представляє кожен клас користувачів;
- Визначення завдань і цілей користувачів, а також бізнес-цілей, з якими ці завдання пов'язані;
- Аналіз інформації, отриманої від користувачів, щоб відокремити завдання від функціональних і нефункціональних вимог, бізнес-правил, передбачуваних рішень і даних, що надходять ззовні;
- Розподіл високорівневих вимог по компонентах ПЗ, розподілених в системній архітектурі;
- Встановлення відносної важливості атрибутів якості;
- Встановлення пріоритетів реалізації;
- Документування зібраної інформації і побудова моделей;
- Перегляд специфікації вимог, який дозволяє впевнитися в тому, що запити користувачів усіма розуміються однаково, і усунення проблем, що виникли до передачі документа розробникам.



# Керування вимогами

- **Визначення основної версії вимог** (моментальний зріз вимог для конкретної версії продукту);
- **Перегляд передбачуваних змін вимог** і оцінка ймовірності впливу кожної зміни до його прийняття;
- **Включення схвалених змін вимог** до проекту встановленим способом;
- **Узгодження плану проекту з вимогами;**
- **Обговорення нових зобов'язань**, заснованих на оціненому впливі зміни вимог;
- **Відстеження окремих вимог** до їх дизайну, вихідного коду та варіантів тестування;
- **Відстеження статусу вимог** та дій зі зміни протягом усього проекту.

# Керування вимогами

- Ідентифікація, класифікація, організація і документування вимог
- Зміна вимог
- Відстеження вимог



# Ідентифікація вимог

- Унікальний ідентифікатор (SUPL, FEAT, UC)
- Послідовний номер всередині ієрархії документу(1; 1.1)
- Послідовний номер в межах категорії вимог(FEAT 1; FEAT 2)

Ієрархічні відносини дозволяють ввести додатковий рівень класифікації вимог.

# Документ опису вимог

1. Попередні зауваження до проєкту
  - Мета та рамки проєкту
  - Діловий контекст
  - Учасники проєкту
  - Ідеї відносно рішень
  - Огляд документу
2. Системні сервіси
  - Рамки системи
  - Функціональні вимоги
  - Вимоги до даних
3. Системні обмеження
  - Вимоги до інтерфейсу
  - Вимоги до продуктивності
  - Вимоги до безпеки
- Експлуатаційні вимоги
  - Політичні та юридичні вимоги
  - Інші обмеження
4. Проєктні питання
  - Відкриті питання
  - Попередній план-графік
  - Попередній бюджет
5. Додатки
  - Глосарій
  - Ділові документи та форми
  - Посилання