



A-Level
UKRAINE

REQUIREMENTS



AGENDA

- Вимоги
- Бізнес-аналіз
- Якість хороших вимог
- Техніки тестування вимог
- Багато практики 😊



План тестування (Test Plan)

документ, що описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку та закінчення тестування, до необхідного в процесі роботи обладнання, спеціальних знань, а також оцінки ризиків з варіантами їх вирішення.

Тест дизайн (Test Design)

процес проектування та створення тестових випадків для проведення надалі перевірки ПЗ з урахуванням специфікації проекту та вимог.

Набір тест кейсів і тестів (Test Case & Test suite)

це послідовність дій, якою можна перевірити чи відповідає тестована функція встановленим вимогам.

Дефекти / Баг Репорти (Bug Reports / Defects)

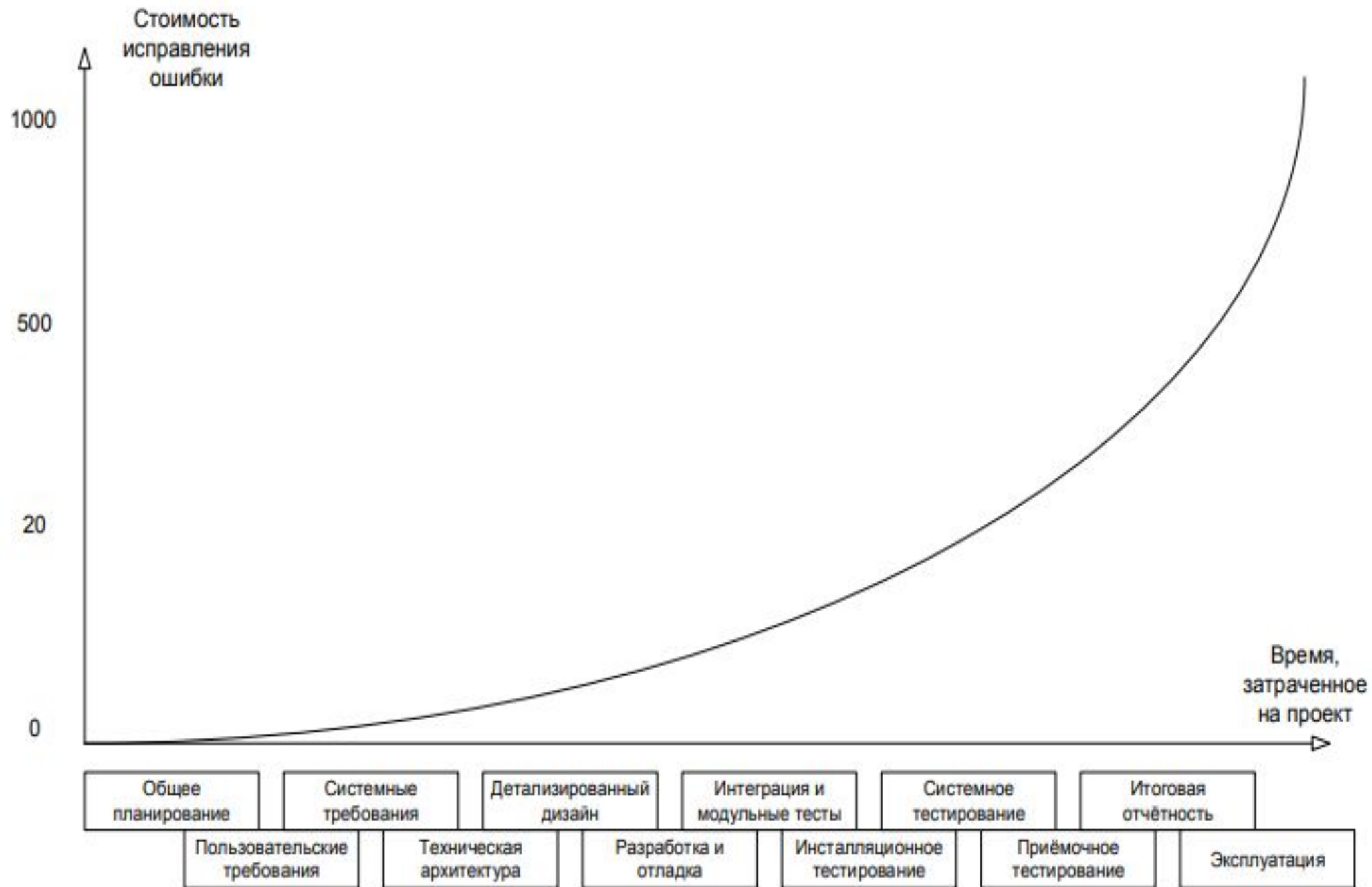
це документи, що описують ситуацію або послідовність дій, що призвела до некоректної роботи об'єкта тестування, із зазначенням причин та очікуваного результату.

Чек-лист (Check-list)

документ, який містить короткий опис функціональності всієї програми, для перевірки його (ПЗ). Створення та написання чек-листа зазвичай замінює написання тестових випадків, у більшості випадків прискорює підготовку до проведення тестування.

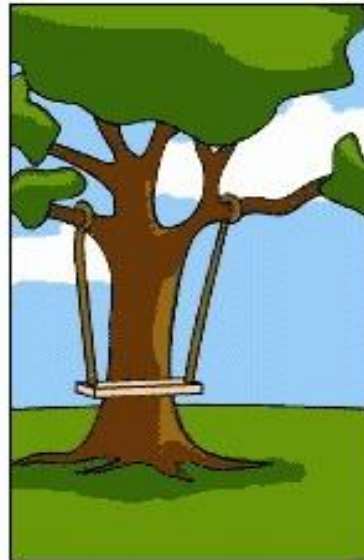
Специфікація вимог (Software Requirements Specification)

це повний, чіткий опис програми, що розробляється. Ключовими особливостями є повне опис всієї функціональності програми, чіткість вимог, відсутність неточностей у тому описі, простота понять, і досить чітка деталізація.





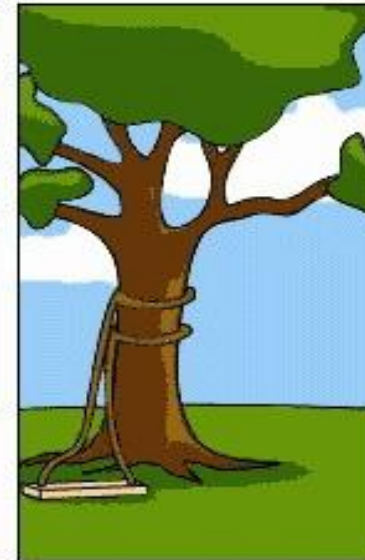
Как объяснил клиент
чего он хочет



Как понял клиента
начальник проекта



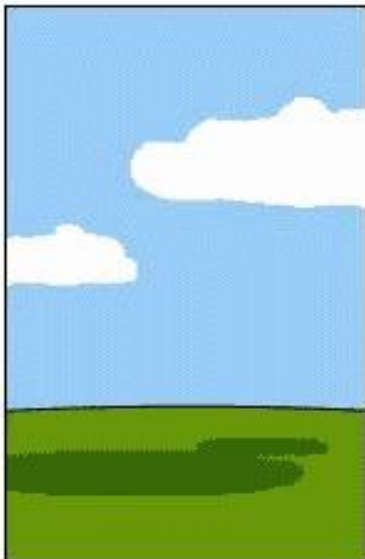
Как описал проект
аналитик



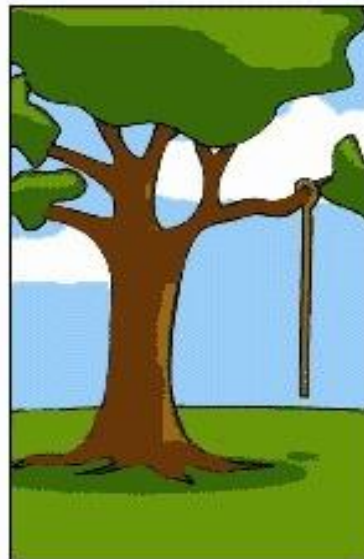
Как написал
программист



Как представил проект
бизнес-консультант



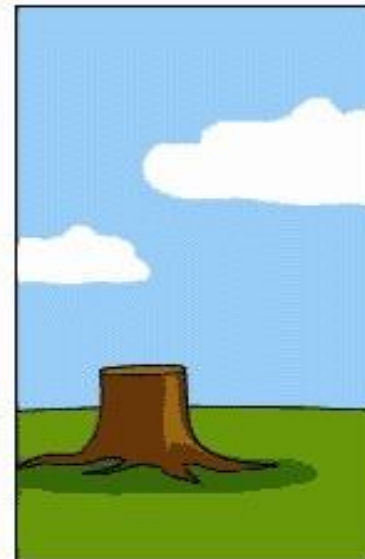
Как задокументировали
проект



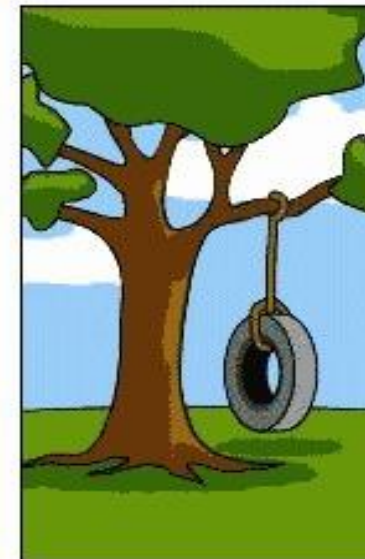
Какие фичи удалось
внедрить



Как заплатил клиент



Как работала
техническая поддержка



Что было нужно
клиенту

Business

Analyst

- де організація знаходиться зараз (as-is);
- який хоче стати/куди хоче прийти (to-be);
- яке рішення дозволить здійснити це перетворення найкращим способом.

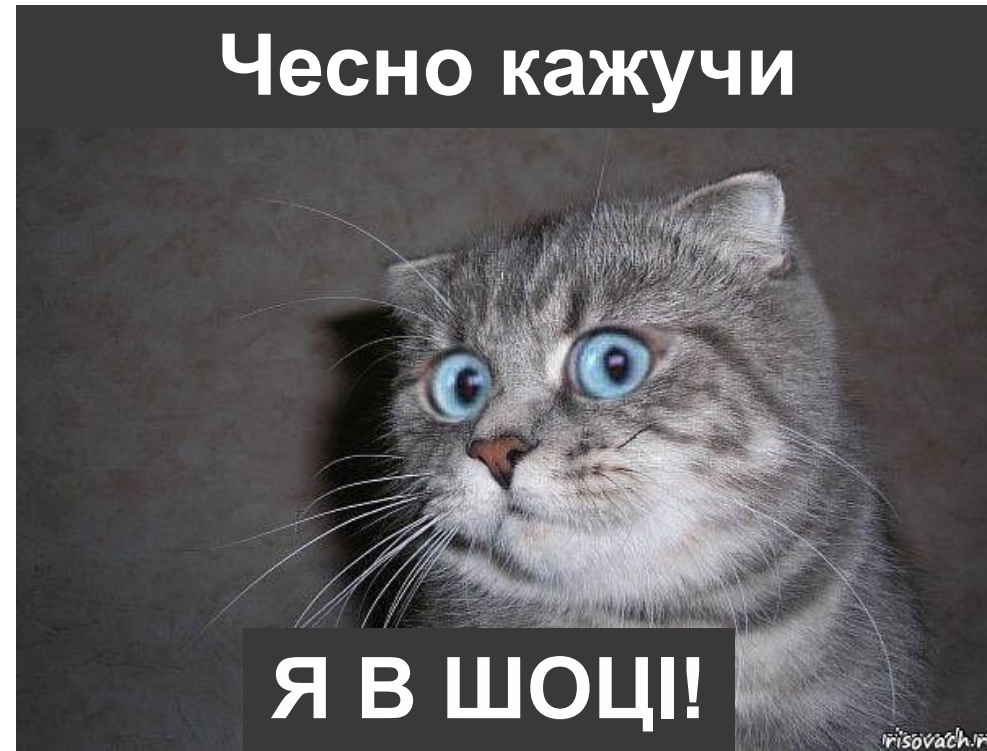




Функції

- зрозуміти, які є незакриті потреби у споживача;
- які у нього є очікування/уподобання, у тому числі приховані;
- який продукт/послуга могли б його зацікавити

ОСНОВНІ ПРИНЦИПИ ТЕСТУВАННЯ ВИМОГ



Ідеї?

1. Тестування вимог краще проводити до початку розробки. Для цього потрібно розрахувати необхідний час на перевірку та заморозити документацію, що тестується, до закінчення перевірки.





2. Проводити тестування вимог можуть як аналітики, і тестувальники. Однак для досягнення кращого результату опис та перевірку вимог слід доручати різним людям.

3. Заклад дефектів документації нічим не відрізняється від закладу дефектів продукту, створюються тікети в баг-трекінгову систему.





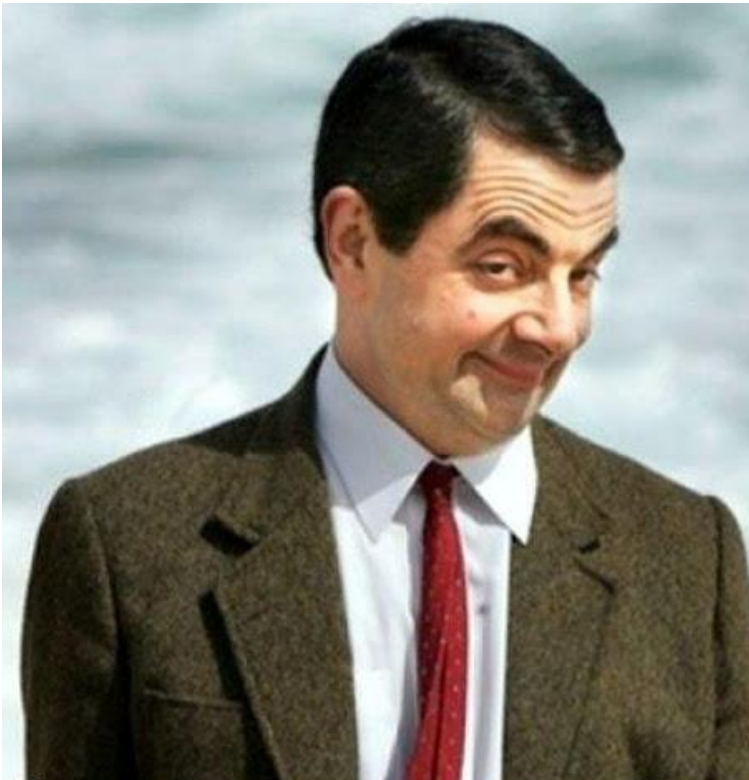
4. У тому випадку, коли перевірка вимог ведеться паралельно з розробкою, вкрай бажано попередити команду розробки знайдених дефектів відразу. Це дозволить призупинити розробку доти, доки вимоги не будуть з'ясовані повністю або щоб вони могли вчасно виправити помилку.

5. Рівень деталізації вимог (як і глибина тестування) залежить від проекту.

ХАРАКТЕРИСТИКИ ХОРОШИХ ВИМОГ

Недвозначність;
Перевірюваність;
Чіткість (стислість);
Точність;
Зрозумілість;
Здійсненність;
Незалежність;
Атомарність;
Необхідність.





1. Недвозначне – має існувати лише одне трактування вимоги. Приміром, слід уникати у тексті вимоги скорочень.

Приклад поганої вимоги: R

EQ1 Система не повинна приймати пароль довше 15 символів.

Дана вимога не прояснює що повинна робити система:

- *Система не повинна дозволяти користувачеві вводити пароль довше 15 символів.*
- *Система має обрізати введений пароль до 15 символів.*
- *Система повинна виводити повідомлення про помилку, якщо користувач введе більше 15 символів.*

Приклад хорошої вимоги:

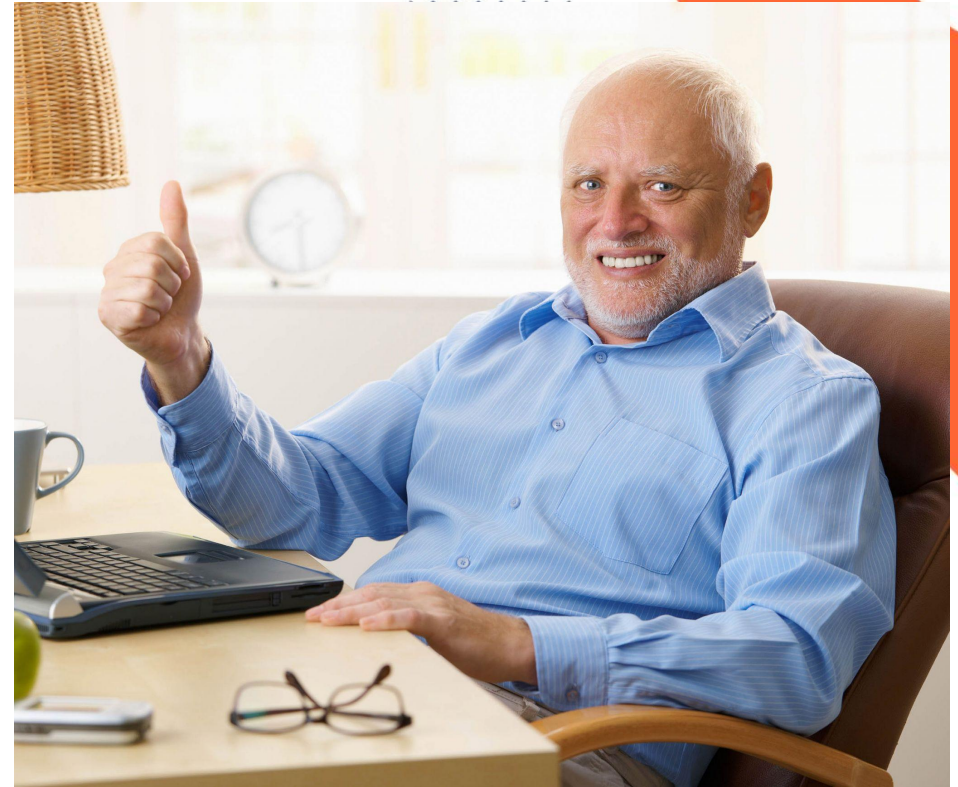
REQ1 Система не повинна приймати пароль довше 15 символів. Якщо користувач введе більше 15 символів, система повинна відобразити повідомлення про помилку з проханням виправити пароль.

2. Перевірене – тестувальники повинні мати можливість перевірити, чи правильно вимога реалізована в системі. Для цього вимога має бути чіткою, точною, недвозначною.

Приклад поганої вимоги:

REQ1 Пошук у системі повинен здійснюватися на ім'я, прізвище користувача тощо.

У цьому прикладі критерії пошуку мають бути розкриті повністю. Інакше не можна перевірити, чи виконується вимога в системі.





3. Чітка (коротка) – вимога не повинна містити зайвої інформації. Воно має бути викладене чітко та просто.

Приклад поганої вимоги:

REQ1 Іноді користувач може вводити код аеропорту, який повинен визначатися системою. Але іноді код аеропорту повинен замінюватися назвою найближчого міста, таким чином користувачеві не потрібно знати код аеропорту, але система як і раніше повинна знати код аеропорту.

Ця вимога може бути змінена так:

REQ1 Система повинна визначати аеропорт як за кодом аеропорту, так і за назвою міста.



4. Точне – вимога має містити справжні факти.

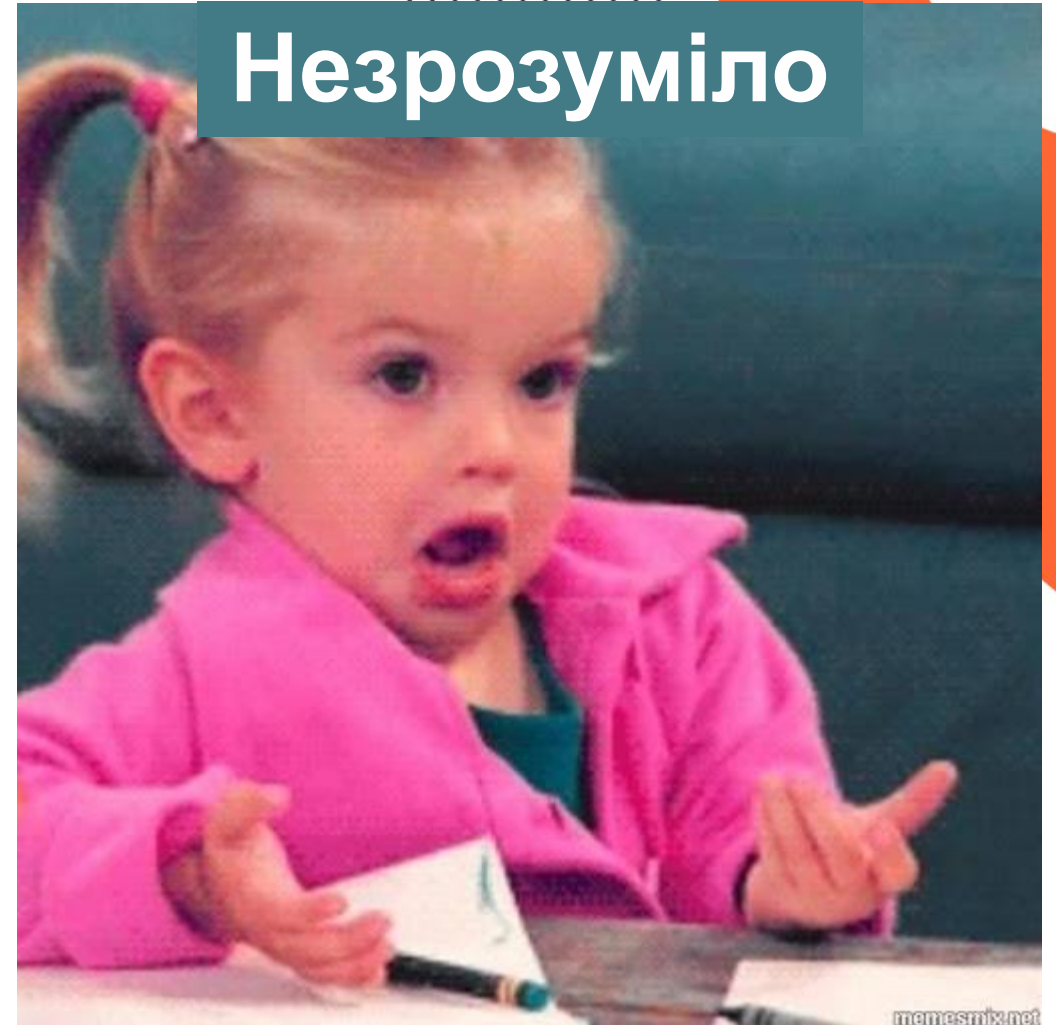
Приклад поганої вимоги: Система повинна швидко обробляти велику кількість користувачів.

Приклад поганої вимоги: Система повинна обробляти 20 користувачів не більше ніж за 3 секунди.

?

Незрозуміло

5. Зрозуміле – вимога має містити граматичних помилок, має бути викладено послідовно.



6. Здійсненна – вимога має бути здійсненою в рамках існуючих обмежень (час, гроші, існуючі ресурси).

Приклад?

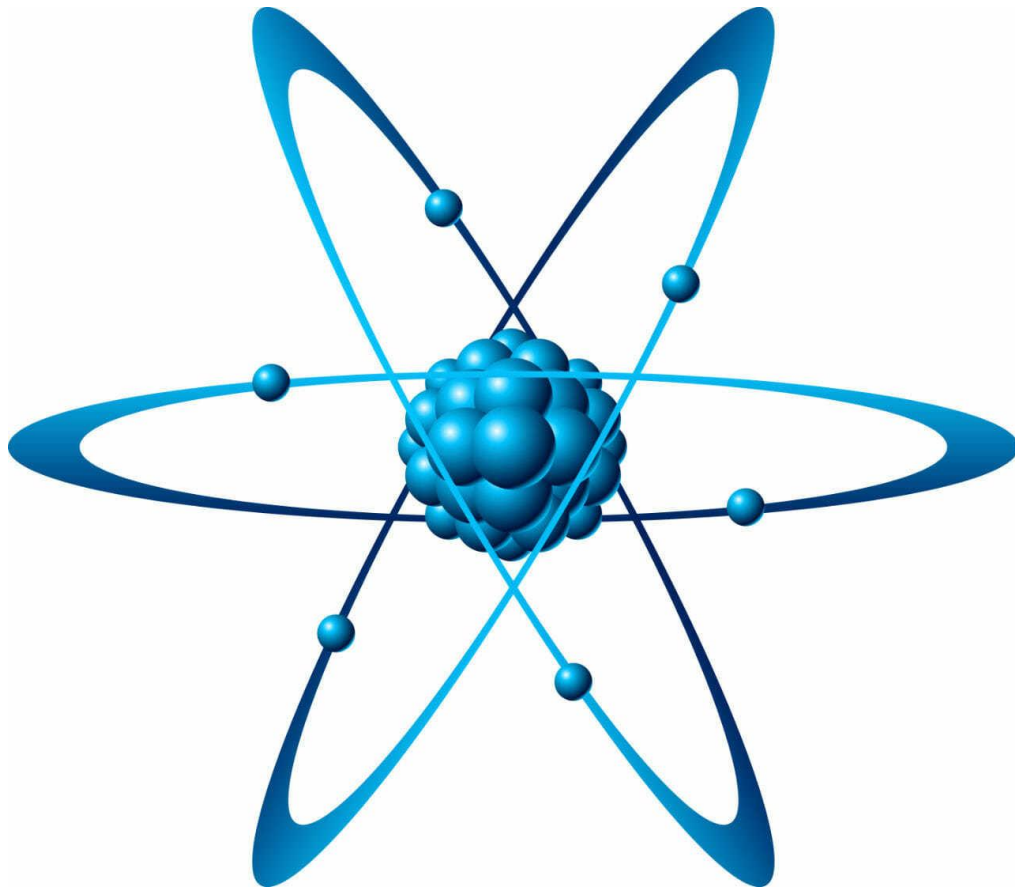
7. Незалежне – для розуміння вимоги не потрібно знати інших вимог.

Приклад поганої вимоги:

REQ1 Список доступних рейсів повинен містити інформацію про номер польоту, час посадки та приземлення.

REQ2 Вони мають бути відсортовані за ціною.





8. Атомарне – вимога має містити одну пов'язану суть.

Приклад поганої вимоги:

REQ1 В системі має бути передбачена можливість забронювати політ, купити квиток, зарезервувати номер у готелі, орендувати машину.

9. Необхідне – зацікавлені особи повинні потребувати цієї вимоги.





Також існують певні характеристики, якими має мати набір вимог:

- а. Постійне – не повинно бути конфліктів між вимогами.
- б. Не надмірна – кожна вимога має бути описана один раз і не повинна містити інших вимог.
- с. Повна – вимога має бути визначена для всіх можливих ситуацій.

Існують додаткові критерії, але вони містять описані вище, наприклад:

Модифікованість - якщо вимога атомарна і повна, вона модифікується.

Трасування (простежуваність) – якщо вимога атомарна і має унікальний ідентифікатор, вона простежується.





A-Level
UKRAINE

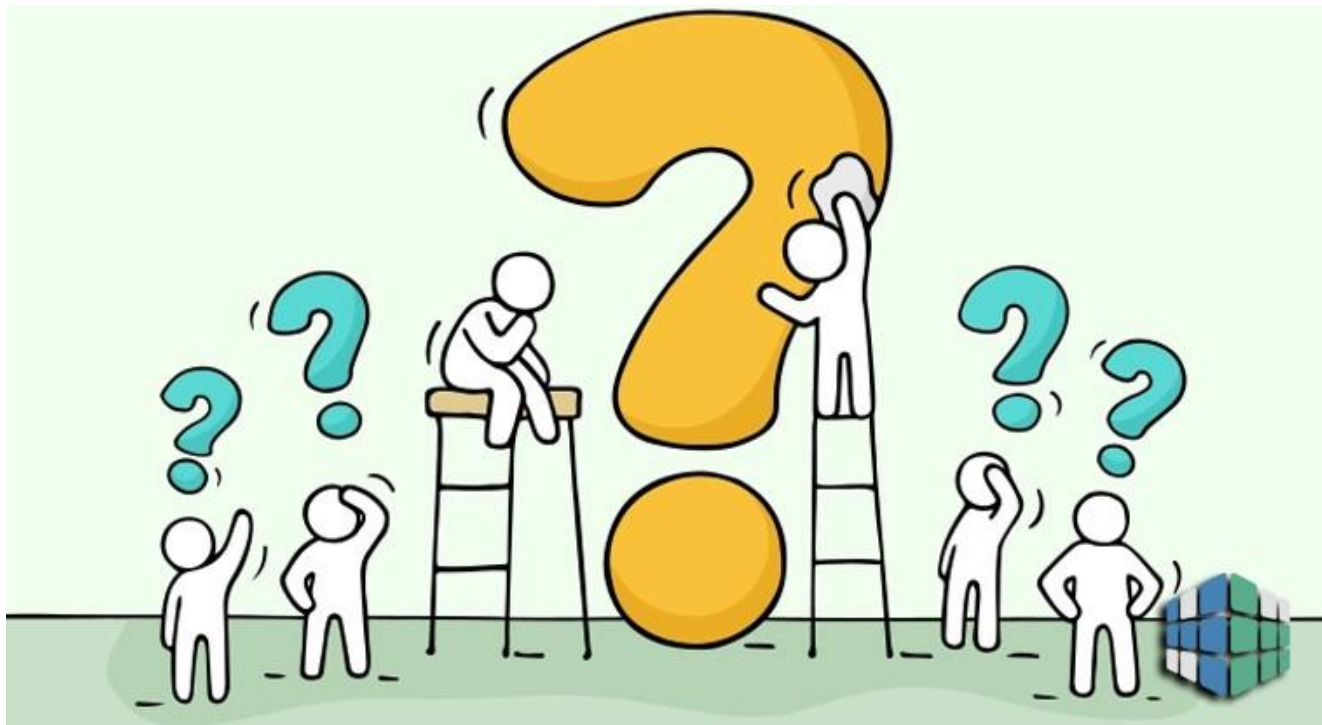
Техніки тестування вимог



Взаємний перегляд:

побіжний перегляд - автор показує свою роботу колегам, вони в свою чергу дають свої рекомендації, висловлюють свої питання та зауваження; технічний перегляд - виконується групою спеціалістів; формальна інспекція — залучається велика кількість фахівців, є структурованим, систематизованим та документованим підходом. Мінус такого варіанту – витрачається багато часу.





Питання:

якщо виникають питання, то можна запитувати у представників замовника більш досвідчених колег.

Тест-кейси та чек-листи:

хороша вимога повинна бути перевіреною, щоб визначити можна використовувати чек-листи або повноцінні тест-кейси. Якщо можна швидко вигадати кілька пунктів чек-листа — це вже добрий знак.



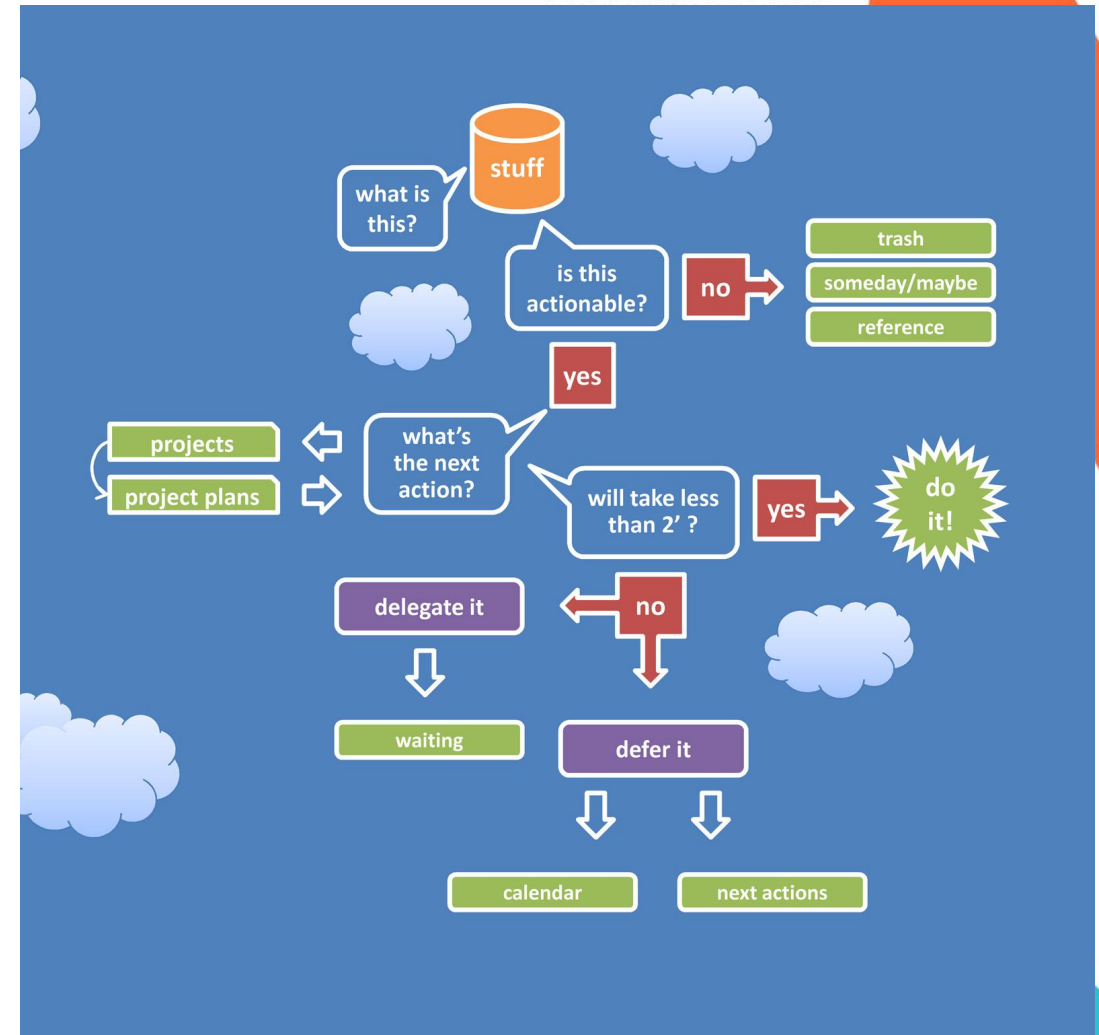


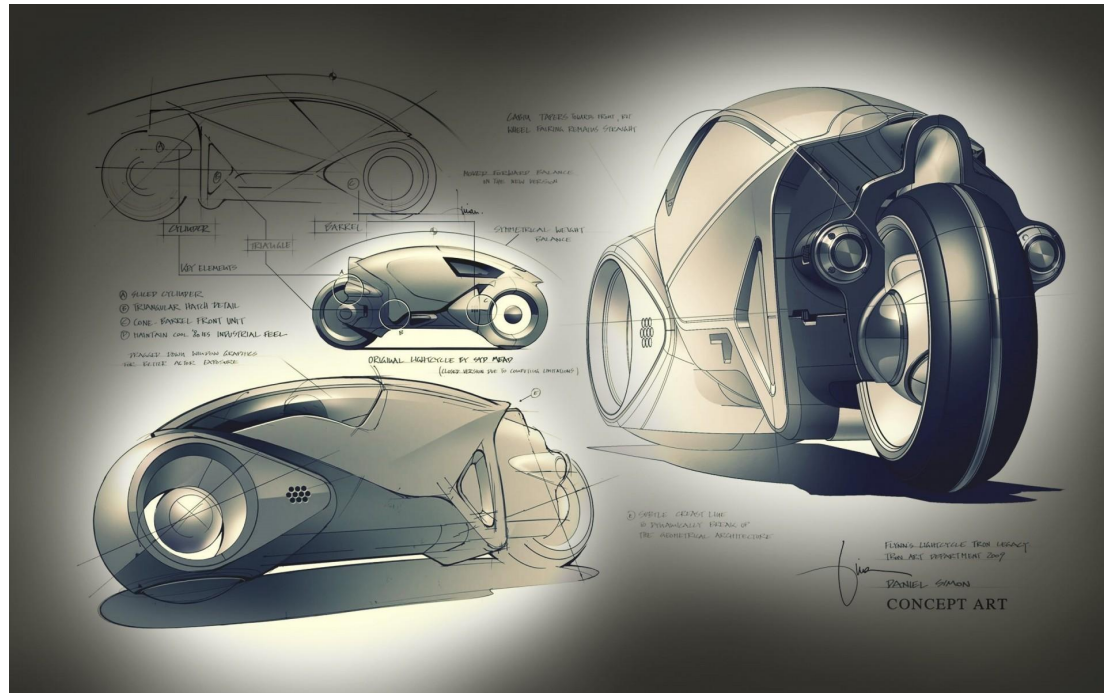
Дослідження поведінки системи:

необхідно подумки змодельювати процес роботи користувача з системою, створеною за тестованими вимогами, після цього визначити неоднозначні варіанти визначення системи.

Малюнки:

графічне уявлення дає наочне уявлення докладання, малюнку простіше побачити, що якісь елементи не стикуються, десь чогось бракує тощо.





Прототипування:

зробивши нариси інтерфейсу користувача, легко оцінити застосувати застосування тих чи інших користувацьких рішень.

Дедлайн:

День лекції

+ 3 повні дні (до 18:00)



| Приклади вимог | Ваші коментарі/ питання | * А як би ви їх переписали? |
|---|----------------------------|--------------------------------|
| 1. Система повинна дозволяти авторизацію користувачів за даними соціальних мереж та вимагати із соціальних мереж та публікувати на стіні користувача в соціальній мережі певну інформацію. | | |
| 2. Система дає відгук при маленькому навантаженні | | |
| 3. Програма повинна швидко запускатися | | |
| 4. Користувач може завантажувати файли у різних форматах. | | |
| 5. Дуже важливо, щоб UI був сучасний, запропонуйте свої варіанти для "Login" форми | | |
| 6. Всі повідомлення про успішну спробу повинні відображатися зеленим кольором, усі повідомлення сповіщувального характеру мають відображатися жовтим, всі повідомлення про помилку повинні відображатися зеленим. | | |



Практика



Звідки беруться вимоги?

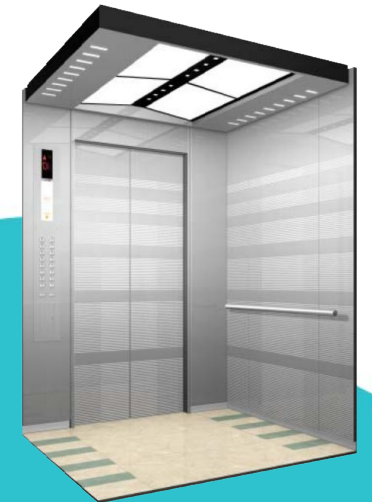
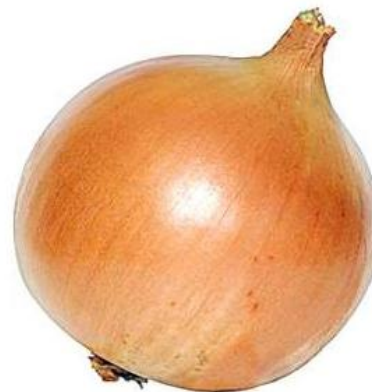


У якій формі можуть бути подано вимоги?



ХАРАКТЕРИСТИКИ ХОРОШИХ ВИМОГ

Недвозначність;
Перевірюваність;
Чіткість (стислість);
Точність;
Зрозумілість;
Здійсненність;
Незалежність;
Атомарність;
Необхідність.



Проаналізувати вимоги:

1. Програма повинна працювати на найпопулярніших браузерах та телефонах.
2. Доступ до лекцій користувач повинен мати лише після авторизації.
3. Користувач повинен мати можливість залогінитись через соціальні мережі.
4. Програма повинна працювати швидко.
5. Система дозволяє користувачеві працювати відразу, як тільки буде запущена.

Визначити, який критерій не виконано

Переписати вимогу, щоб вона стала якісною

