
Управління проектами

ДЕНЬ 3



Вимоги та їх аналіз

Вимоги

Вимога - це умова чи можливість (capability) якій повинна відповідати система

Вимоги прийнято поділяти на функціональні та нефункціональні

Нефункціональні вимоги

Нефункціональні вимоги, регламентують внутрішні і зовнішні умови або атрибути функціонування системи

Виділяють наступні основні групи нефункціональних вимог:

- Зовнішні інтерфейси (External Interfaces),
- Атрибути якості (Quality Attributes),
- Обмеження (Constraints).

Зовнішні інтерфейси

Зовнішні інтерфейси – це взаємозв'язки системи з зовнішнім світом.

Не варто зводити зовнішні інтерфейси тільки до інтерфейсу користувача.

До них також включають зв'язки із зовнішніми системами. Наприклад:

«Інтерфейс користувача Системи повинен бути витриманий у корпоративних кольорах Замовника.»

або

«Система бухгалтерського обліку повинна отримувати дані про співробітників з кадрової системи. Дані отримують шляхом імпорту з періодом один раз на тиждень.»

Атрибути якості

Атрибути якості встановлюють пріоритети у розробці системи з точки зору якісних характеристик

Обмеження

Обмеження вказують на умови, які накладають на систему.

Наприклад:

«Загальна кількість користувачів системи - до 50 000.»

«Кількість користувачів, які працюють з системою одночасно, - до 3 000.»

або

***«Система дозволяє будувати тільки стандартні звіти, перелік яких, наведено у ...
Система не будує нестандартні звіти.»***

Основні атрибути якості

Застосовуваність (Usability)

Надійність (Reliability)

Продуктивність (Performance)

Підтримуваність (Supportability).

Основні атрибути якості.2

Застосовуваність (Usability – часто дають просто трансліт юзабіліті) – це зручність системи до застосування. Прикладом таких вимог може бути:

«Будь-яка функція системи повинна бути доступною за допомогою не більше, ніж трьох кліків мишею»

Основні атрибути якості.3

Надійність – це здатність системи працювати безвідмовно певний проміжок часу.

Надійність вказують у годинах безвідмовної роботи, або, частіше, у відсотках безвідмовної роботи по відношенню до відмов.

Наприклад «три дев'ятки» - 99,9%. Або «п'ять дев'яток» - 99,999%.

Розраховують надійність, виходячи з графіку роботи системи.

- Якщо система повинна працювати тільки у робочий час (8x5, або 8 годин на тиждень 5 днів тижня), то відмова системи у неробочий час не враховується. Якщо ж система повинна працювати цілодобово (24x7), то враховуються всі відмови. Не вважаються також відмовами планові зупинки системи для її обслуговування. Для розрахунку надійності ділять загальний допустимий час відмов (на рік) на загальний час роботи системи та віднімають результат від 100%. Наприклад, при цілодобовій роботі системи надійність - 99,999% означає, що загальний час непрацездатності системи не повинен перевищувати приблизно 40 хвилин на рік.

Основні атрибути якості.4

Продуктивність системи вимірюють за різними параметрами. Прикладами відповідних вимог можуть бути:

«Час відповіді системи на будь-який запит не повинен перевищувати 3 сек.»

або

«Система повинна обробляти не менше 3 000 tps (транзакцій на секунду)»

Підтримуваність - це здатність системи до підтримки. Вимоги до підтримуваності системи можуть встановлювати умови підтримки, кваліфікацію персонала, або термін підтримки розробником. Ці вимоги також можуть включати час відновлення після аварії. Наприклад:

«Час відновлення після аварії не повинен перевищувати 4 год.»

Процес управління вимогами

Виявлення вимог (Requirements Elicitation),

Аналіз вимог (Requirements Analysis),

Специфікація вимог (Requirements Specification),

Валідація вимог (Requirements Validation).

Виявлення вимог

Як правило, замовник не може самостійно повністю та чітко сформулювати вимоги до створюваної системи

Розробники повинні допомогти замовнику виявити та описати вимоги

При цьому, чим чіткіше будуть описані вимоги, тим краще для всіх: розробників та замовника

Стратегії виявлення вимог:

- Інтерв'ю
- Анкетування
- Стеження
- Спільні семінари

Виявлення вимог. Інтерв'ю

Інтерв'ю полягає у проведенні бесід розробників з представниками замовника.

При виявленні вимог за допомогою інтерв'ю треба дотримуватись таких правил:

- З'ясувати, кого опитувати
- Узгодити план опитувань
- Підготувати перелік запитань та передати їх тим, кого опитують
- Фіксувати запитання та відповіді під час інтерв'ю
- Описати результат проведення інтерв'ю
- Проаналізувати інтерв'ю та уточнити відкриті питання

Виявлення вимог. Інтерв'ю. 2

Опитувати треба тих представників замовника, які є зацікавленими особами у створенні системи.

Порядок проведення опитування також має значення. Можна проводити опитування згори донизу (від головного замовника до користувачів) або ж знизу догори (від користувачів до головного замовника).

Перед проведенням інтерв'ю треба підготувати перелік запитань. Не варто сподіватись, що ви вдало зімпровізуєте та задасте всі потрібні питання без підготовки.

Звідки треба брати питання? По перше - з того опису майбутньої системи, який є на початку розробки. По друге - треба заздалегідь запросити у замовника загальні відомості про його організацію: структуру, основні інформаційні потоки, дані, з якими доведеться працювати, приклади роботи існуючої системи (якщо така є) тощо.

Підготовлений перелік запитань треба передати тим, кого будуть опитувати, заздалегідь.

Виявлення вимог. Інтерв'ю. 3

Обов'язково треба фіксувати питання та відповіді під час проведення інтерв'ю.

Якщо під час інтерв'ю можуть виникати додаткові запитання, їх треба з'ясувати невідкладно.

Після проведення інтерв'ю необхідно розшифрувати їх результат у вигляді документа з питаннями, відповідями, відкритими питаннями (які залишились без відповіді).

Після проведення всіх інтерв'ю документи з їх описами порівнюють, виявляють протиріччя (якщо є) та відкриті питання. В результаті отримують зведений документ, який містить основні вимоги до системи.

Виявлення вимог. Анкетування

Анкетування полягає у проведенні дистанційного опитування зацікавлених осіб з боку замовника з використанням підготовлених анкет.

Анкетування варто робити, коли кількість майбутніх користувачів є достатньо великою, щоб утворити статистичну вибірку.

При виборі цієї стратегії корисною буде наявність подібної готової системи, яку треба оновити або замінити.

Анкети включають здебільшого закриті питання з вибором однієї або декількох альтернатив із запропонованих варіантів.

Після проведення анкетування виконують обробку результатів шляхом аналізу наданих відповідей та документують виявлені вимоги.

Виявлення вимог. Стеження

Стратегія стеження (spying) полягає у тому, що представники розробників слідкують за виконанням співробітниками замовника своєї щоденної роботи.

Як правило, стеження займає якийсь визначений часовий період.

Під час стеження питання не задають. Натомість фіксують поточні дії співробітників.

Після завершення стеження складають документ, у якому описують поточні бізнес-процеси а також потенційні вимоги до системи

Виявлення вимог. Спільні семінари

Спільні семінари (JAD - Joint Application Design) проводять за участю розробників та (майбутніх) користувачів системи

Цю стратегію часто використовують для систем великої серії

Розробники презентують прототип майбутньої системи та акцентують увагу на нових або змінених функціях

Користувачі оцінюють презентовані функції а також можуть вносити свої пропозиції щодо майбутньої системи

Аналіз вимог

Головним результатом виявлення вимог є перелік (каталог) вимог

Цей перелік складають у вигляді таблиці, в якій спочатку заповнюють два стовпці: ідентифікатор та суть (короткий опис) вимоги

Ідентифікатор слугує для однозначного маркування кожної вимоги, яка може проходити різні стадії за час розробки системи

Стандартної системи ідентифікації вимог не існує. Якщо каталог вимог складають вручну, то варто ідентифікувати вимоги так, щоб їх можна було природньо впорядкувати за ідентифікаторами. Треба також зважати на можливість виникнення нових вимог поруч з виявленими раніше

Аналіз вимог.2

Ідентифікатор вимоги	Опис вимоги
R1.5	СИСТЕМА повинна дозволяти задавати / змінювати нові власні статуси завдань і вказувати життєвий цикл статусів (допустимі попередники / послідовники)
R1.6	СИСТЕМА повинна дозволяти виконавцю задавати / змінювати фактичні працевитрати / вартість завдання. При цьому автоматично повинні переобчислюватись працевитрати / вартість батьківських завдань.
R1.7	В СИСТЕМІ має бути можливість створення груп користувачів і розмежування прав доступу.
R1.8	В СИСТЕМІ повинні бути реалізовані оповіщення про події (виконання завдання, зміни в параметрах завдання, тощо)
R1.9	Відправка оповіщень про події повинна налаштовуватися для зацікавлених сторін.

Пріоритезація вимог

Після побудови каталогу вимог виконують їх пріоритезацію, тобто визначають ступінь важливості вимог. Для пріоритезації використовують різні методи. Один із них - це MoSCoW

У ньому використовуються чотири пріоритети - Must (M), Should (S), Could (C) і Won't (W)

Елементи, позначені як Must зобов'язані бути включені в продукт

Елементи Should також є важливими для продукту, але, вони можуть бути виключені з поточної версії продукту з об'єктивних причин

Елементи Could є некритичними, але здатні підвищити задоволення користувачів

Елементи Won't є найменш критичними для продукту. Вони можуть вважатися цікавими і перспективними для майбутніх версій продукту, але точно не будуть реалізовані в поточній версії

Пріоритезація вимог.2

Ідентифікатор вимоги	Опис вимоги	Пріоритет
R1.5	СИСТЕМА повинна дозволяти задавати / змінювати нові власні статуси завдань і вказувати життєвий цикл статусів (допустимі попередники / послідовники)	M
R1.6	СИСТЕМА повинна дозволяти виконавцю задавати / змінювати фактичні працевитрати / вартість завдання. При цьому автоматично повинні переобчислюватись працевитрати / вартість батьківських завдань.	M
R1.7	В СИСТЕМІ має бути можливість створення груп користувачів і розмежування прав доступу.	M
R1.8	В СИСТЕМІ повинні бути реалізовані оповіщення про події (виконання завдання, зміни в параметрах завдання, тощо)	S
R1.9	Відправка оповіщень про події повинна налаштовуватися для зацікавлених сторін.	S

Моделювання та ОПИС ВИМОГ

Пріоритезовані вимоги повинні бути формалізовані.

Здебільшого це стосується функціональних вимог.

Така формалізація передбачає більш докладний опис.

Розглянемо два популярних підходи:

- моделювання за допомогою прецедентів
- написання користувацьких історій

Моделювання за допомогою прецедентів

Моделювання вимог за допомогою моделі прецедентів використовується у об'єктно-орієнтованому підході до аналізу вимог.

Прецедент (use case або блок використання або варіант використання) - це частина функціональності системи, яка надає питомий результат користувачу.

Наприклад: **"сформувати рахунок"** або **"змінити ціну товару"**.

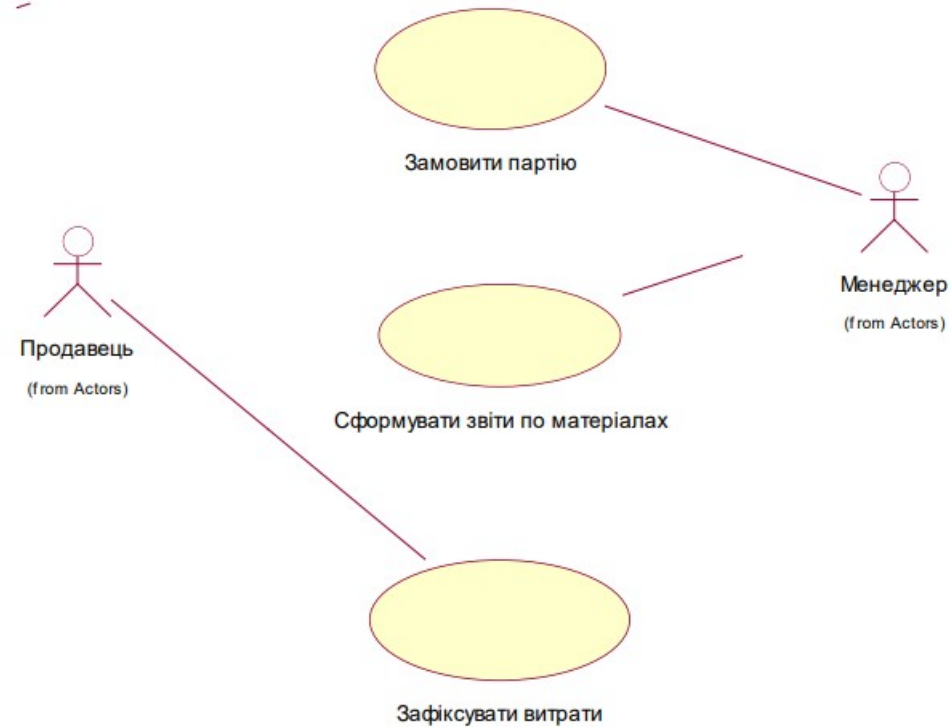
Прецеденти можуть бути зв'язані між собою.

Також з прецедентами взаємодіють актори.

Актор - це будь-яка сутність, яка взаємодіє з системою. Актором може бути людина, інша система або час (таймер).

Прецедент, як правило, відображає одну або декілька вимог до системи.

Моделювання за допомогою прецедентів.2



Моделювання за допомогою прецедентів.3

Модель прецедентів не обмежується тільки зображеннями прецедентів, акторів та їх зв'язків. Кожен прецедент має бути описаний.

У описі визначаються:

- Початкова умова - умова початку дії прецедента;
- Кінцева умова - умова, яка виконується після завершення прецедента;
- Нормальний хід подій - сценарій, який відповідає нормальному ходу подій прецедента;
- Альтернативні ходи подій - сценарії, які описують, що відбувається при виникненні альтернатив, помилок або виключних ситуацій.

Опис прецеденту. приклад

1.1. Найти ПАК

Предусловие: Пользователь авторизовался в системе. Пользователю назначена роль «Менеджер» или «Супервизор»

1. Пользователь нажимает на кнопку «ПАК» на главной форме (MainManagerForm.frm)
2. Система открывает окно поиска ПАК (PackListView.frm)
3. Пользователь вводит критерии поиска и нажимает кнопку «Поиск»
4. Система выводит список найденных ПАК (PackListView.frm)

Исключения:

4а Данные не найдены. Система выводит пустой список

4б Пользователь не ввел критерии для поиска. Система выводит сообщение о необходимости введения критериев для поиска

Написання користувацьких історій

Користувацька історія - це варіант використання майбутнього продукту в конкретній ситуації з метою досягнення вимірюваного результату

Кожна користувацька історія повинна приносити користь - не повинно бути історій, які виконують дії заради дій

Користувацькі історії можуть містити як складні інструкції з розгалуженнями, так і конкретні приклади

Якщо дії користувачів продиктовані скоріше здоровим глуздом, ніж інструкцією, то користувацька історія повинна містити приклад реальної ситуації, а їх систематизація та оптимізація повинна бути залишена на потім

Якщо ж бізнес процеси, які автоматизує продукт, суворо формалізовані, то користувацька історія повинна містити алгоритми дій людей, що працюють з нею

Написання користувацьких історій

Користувацькі історії є доволі короткими та конструюються за шаблоном:

"Як <роль> я хочу <що робити> з метою <що отримати>"

Наприклад, користувацька історія для входу до системи отримання контенту за підпискою:

"Як зареєстрований користувач я хочу увійти до системи з метою отримання доступу до контенту за підпискою".

Специфікація вимог

Специфікація - моделями, документами

Підтвердження (валідація) вимог

Перевірка - неформальна або формальна

Відстеження вимог

Стан	Визначення
Proposed (Запропоновано)	Вимогу запитано авторизованим джерелом
Approved (Схвалено)	Вимогу проаналізовано, її вплив на проект прораховано, і вона була розміщена в базовій версії певної версії. Ключові зацікавлені в проекті особи погодилися з цією вимогою, а розробники ПЗ зобов'язалися реалізувати її.
Implemented (Реалізовано)	Код, який реалізує вимога, розроблений, написаний і протестований. Вимогу відстежено до відповідних елементів дизайну та коду.
Verified (Перевірено)	Коректне функціонування реалізованої вимоги підтверджено у відповідному продукті. Вимогу відстежено до відповідних варіантів тестування. Тепер вимога вважається завершеною.
Deleted (Вилучено)	Затверджену вимогу видалено з базової версії. Опишіть причини видалення і назвіть того, хто прийняв це рішення
Rejected (Відхилено)	Вимогу запропоновано, але не заплановано для реалізації в жодній з майбутніх версій. Опишіть причини відхилення вимоги і назвіть того, хто прийняв це рішення