

Бізнес-аналіз та аналіз вимог

№ уроку: 6 **Курс:** Manual QA

Засоби навчання: Браузер, Microsoft Office

Огляд, мета та призначення уроку

Метою даного уроку є знайомство з документацією, як із нею працювати, щоб не робити помилок самому і не допускати помилок з боку колег.

Вивчивши матеріал даного заняття, учень зможе:

- розбиратися у проектній документації;
- самостійно працювати з усіма видами документації;
- самостійно створювати усі необхідні види документації.

Зміст уроку

1. Процес тестування. SDLS та STLS.
2. Що таке «Вимога».
3. Важливість вимог.
4. Як виявляти вимоги.
5. Рівні та типи вимог.
6. Характеристики якісних вимог.
7. Методи тестування вимог.

Резюме

Життєвий цикл розробки програмного забезпечення (SDLC) спрямований на створення високоякісної системи, яка відповідає очікуванням клієнтів або перевершує їх, ефективно та дієво працює в існуючій та запланованій інфраструктурі інформаційних технологій, недорогої в обслуговуванні та рентабельній для покращення.

Життєвий цикл тестування програмного забезпечення (STLC) визначає, які дії при тестуванні виконувати та коли виконувати ці дії з тестування. Попри те що, що тестування у різних організаціях відрізняється, існує життєвий цикл тестування.

Бізнес-аналіз – це дослідницька наука про виявлення потреб бізнесу та визначення рішень для виявлених бізнес-завдань.

Аналіз – це процес розкладання чого-небудь на складові. Аналітик збирає інформацію, дані, технології і т. д., а потім за допомогою різних засобів, інструментів, у тому числі і свого власного мозку, знань, умінь, аналітичного мислення, методологій, розкладає все по полицях.

Вимоги до ПЗ складаються з трьох рівнів – бізнес-вимоги, вимоги користувачів та функціональні вимоги. На додачу кожна система має свої нефункціональні вимоги. Модель на мал. нижче ілюструє спосіб представлення цих типів вимог.

Бізнес-вимоги (business requirements) містять високорівневі цілі організації або замовників системи. Як правило, їх висловлюють ті, хто фінансує проект, покупці системи, менеджер реальних користувачів, відділ маркетингу. У цьому документі пояснюється, чому організації потрібна така система, тобто описані цілі, яких організація має намір досягти з її допомогою. Мені подобається записувати бізнес-вимоги у формі документу про образ та межі проекту, який ще іноді називають статутом проекту (project charter) або документом ринкових вимог (market requirements document). Визначення меж проекту являє собою перший етап управління загальними проблемами збільшення обсягу робіт.

Вимоги користувачів (user requirements) описують цілі та завдання, які користувачам дасть система. До відмінних способів подання цього виду вимог належать варіанти використання, сценарії та таблиці «подія – відгук». Таким чином, у цьому документі зазначено, що клієнти зможуть робити за допомогою системи.

Функціональні вимоги (functional requirements) визначають функціональність ПЗ, яку розробники повинні побудувати, щоб користувачі змогли виконати свої задачі у рамках бізнес-вимог. Іноді вони називаються вимогами поведінки (behavioral requirements), вони містять положення з традиційним «повинен» або «повинна»: «Система повинна електронною поштою надсилати користувачеві підтвердження про замовлення». Функціональні вимоги документуються у специфікації вимог до ПЗ (software requirements specification, SRS), де описується настільки повно, наскільки необхідно очікувана поведінка системи.

Системні вимоги (system requirements) – це високорівневі вимоги до продукту, які містять багато підсистем. Говорячи про систему, ми маємо на увазі програмне забезпечення або підсистеми ПЗ та обладнання. Люди – частина системи, тому певні функції системи можуть поширюватися і на людей.

Бізнес-правила (business rules) включають корпоративні політики, урядові постанови, промислові стандарти та обчислювальні алгоритми. Бізнес-правила не є вимогами до програмного забезпечення, тому що вони знаходяться зовні меж будь-якої системи програмного забезпечення. Однак вони часто накладають обмеження, визначаючи, хто може виконувати конкретні ВІ, або диктувати, які функції повинна мати система, що підпорядковується відповідним правилам. Іноді бізнес-правила стають джерелом атрибутів якості, що реалізуються у функціональності. Отже, ви можете відстежити походження конкретних функціональних вимог аж до відповідних їм бізнес-правил.

Нефункціональні вимоги описують цілі та атрибути якості. Атрибути якості (quality attributes) є додатковим описом функцій продукту, виражених через опис його характеристик, важливих для користувачів або розробників.

До таких характеристик належать:

- * легкість та простота використання;
- * легкість переміщення;
- * цілісність;
- * ефективність та стійкість до збоїв;
- * зовнішні взаємодії між системою та зовнішнім світом;
- * обмеження дизайну та реалізації.

Обмеження (constraints) стосуються вибору можливості розробки зовнішнього вигляду та структури продукту.

Атрибути вимог:

- **коректність** – точний опис функціоналу, що розроблюється.
- **перевірюваність** – формулювання вимог таким чином, щоб можна було виставити однозначний вердикт, чи виконано все відповідно до вимог, чи ні.
- **повнота** – у вимозі повинна міститися вся необхідна для реалізації функціональності інформація.
- **недвозначність** – вимога має містити однозначні формулювання.
- **несуперечність** – вимога не повинна містити внутрішніх протиріч та протиріч іншим вимогам та документам.
- **пріоритетність** – у кожній вимозі має бути пріоритет (кількісна оцінка ступеня значущості вимоги). Цей атрибут дозволить грамотно керувати ресурсами на проекті.
- **атомарність** – вимогу не можна розбити на окремі частини без втрати деталей.
- **модифікованість** – у кожному вимозі можна внести зміну.
- **простежуваність** – кожна вимога повинна мати унікальний ідентифікатор, за яким на неї можна послатися.

Техніки тестування вимог:

1. **Взаємний перегляд**
 - a. **швидкий перегляд** - автор показує свою роботу колегам, вони у свою чергу дають свої рекомендації, висловлюють свої питання та зауваження;
 - b. **технічний перегляд** - виконується групою спеціалістів;
 - c. **формальна інспекція** – залучається велика кількість фахівців, є структурованим, систематизованим та документованим підходом. Мінус такого варіанту – витрачається багато часу.
2. **Питання** — якщо виникають питання, то можна запитувати у представників замовника, досвідченіших колег.
3. **Тест-кейси та чек-листи** - хороша вимога має бути перевіреною; щоб це визначити, можна використовувати чек-листи або повноцінні тест-кейси. Якщо можна швидко вгадати кілька пунктів чек-листу - це вже добрий знак.
4. **Дослідження поведінки системи** - необхідно подумки змодельовати процес роботи користувача з системою, створеною за тестованими вимогами, після цього визначити неоднозначні варіанти визначення системи.
5. **Малюнки** — графічне уявлення дає наочне представлення застосунку, на малюнку простіше побачити, що якісь елементи не стикуються, деось чогось не вистачає тощо.
6. **Прототипування** - зробивши нариси інтерфейсу користувача, легко оцінити застосування тих чи інших користувацьких рішень.

Закріплення матеріалу

1. Які бувають рівні та типи вимог до документації?
2. Чим відрізняється бізнес-рівень від користувацького рівня вимог?
3. Які джерела з'ясування вимог можна назвати?

Самостійна діяльність учня

Завдання 1

Напишіть функціональні вимоги для кавоварки. Не менше, ніж 5 штук.

Завдання 2

Допишіть ще 5 нефункціональних вимог.

Завдання 3

Напишіть по 2-3 "гарні" питання для кожної вимоги.

Рекомендовані ресурси

Що таке аналіз та збір вимог у SDLC?

<https://uk.myservername.com/what-is-requirement-analysis>

Характеристики якості вимог

https://uk.wikipedia.org/wiki/Якість_програмного_забезпечення

Чек-ліст тестування вимог

<https://training.gatetestlab.com/blog/technical-articles/work-with-checklist/>