

Лабораторна робота 1. Дослідницьке тестування, оракули та схеми покриття

Тема. Дослідницьке тестування незнайомого продукту: схема покриття продукту, чартери та сесії, оракули, тестопридатність.

Мета. Провести на дисципліноване дослідження (явна модель продукту, сформульована причина кожного вибору, що тестувати, і названий оракул за кожною знахідкою).

Застосунок для тестування

OrderFlow 1.0 — невеликий вебсервіс замовлень та рахунків-фактур для оптового постачальника канцелярських товарів, з вебінтерфейсом і REST API, у вигляді архіву `OrderFlow_lab1.zip`. Ви тестуєте його ззовні, без доступу до коду.

— `README.md` пояснює, як запустити застосунок. Потрібен .NET 10 (runtime або SDK: <https://dotnet.microsoft.com/download/dotnet/10.0>).

Розпакуйте архів і запустіть застосунок зі своїм номером варіанта:

```
ORDERFLOW_VARIANT=<ваш номер у списку групи> dotnet  
app/OrderFlow.Api.dll
```

— Застосунок доступний за адресою `http://localhost:5000`; опис API — за адресою `http://localhost:5000/swagger`. Номер варіанта виводиться в нижньому колонтитулі сторінки і має бути видимим на ваших знімках екрана.

— `PRODUCT.md` — опис продукту, написаний продуктовою командою. Це єдина специфікація, яку ви маєте.

— `requests.http` — набір підготовлених запитів до API (VS Code «REST Client» або HTTP-клієнт JetBrains; `curl` чи Postman підходять так само).

— `templates/` — схема покриття, аркуш сесії та таблиця оракулів.

— Облікові записи: `manager / manager`; клієнти `alice / alice (VIP)`, `bob / bob`, `carol / carol`.

База даних — файл `app/orderflow.db`; видаліть його, щоб повернутися до демонстраційних даних. У кожному аркуші сесії зазначаєте, з яких даних ви починали.

Правила роботи

— Працюйте самостійно. Сесії обмежені в часі. Якщо сесія вийшла за межі відведеного часу, зупиніться, зафіксуйте час і почніть нову сесію з новим чартером.

— Ведіть нотатки під час тестування, а не після: чорнові нотатки під час сесії, потім п'ятнадцять хвилин наприкінці, щоб перетворити їх на архівні.

- Записуйте те, що спостерігали, окремо від того, який з цього висновок. «Показана сума становила 1050,00» — спостереження; «знижка неправильна» — висновок, якому потрібен оракул.
- Кожна знахідка потрапляє до таблиці оракулів із названим оракулом. Знахідка, для якої ви не можете назвати оракул, — це питання, а не дефект; тримайте її в списку питань.
- Звіти про дефекти поки що не пишуть. Одного рядка на дефект в аркуші сесії достатньо (звіти належать до лабораторної роботи 2).
- Використовуйте обидва інтерфейси. Вебсторінки показують продукт так, як його подають розробники; API показує, що продукт робить насправді.

Хід роботи

Завдання 0. Знайомство з продуктом (20 хвилин)

Запустіть OrderFlow зі своїм номером варіанта. Увійдіть як клієнт, потім як менеджер. Пройдіть **тур переваг**: користуйтеся продуктом так, як задумали його розробники, за основним сценарієм, і подивіться, що він робить добре. Перегляньте каталог, створіть замовлення через форму, оплатіть його, відкрийте рахунок-фактуру, перегляньте те саме замовлення через API.

Зафіксуйте: п'ять рядків перших вражень — для чого продукт, хто ним користується, що вас здивувало. Дефектів поки не шукайте.

Завдання 1. Схема покриття продукту (45 хвилин)

Побудуйте **схему покриття продукту** за шаблоном `templates/coverage_outline.md` — як список з відступами або як мапу думок, за сімома елементами продукту: структура, функції, дані, інтерфейси, платформа, операції, час.

- Почніть зі *знайдених* схем: `PRODUCT.md`, сторінка Swagger, меню вебінтерфейсу. Потім досліджуйте і додавайте те, чого в них немає.
- Позначте кожен запис: (F) — знайдено, (E) — виявлено дослідженням.
- Включайте елементи, які не збираєтеся тестувати. Схема — це перелік, а не план.
- Завершіть **ранжуванням за ризиком**: п'ять ділянок, які ви тестували б першими; для кожної — одна фраза про те, чому вона, ймовірно, відмовить, і одна про те, чому відмова матиме значення.

Зафіксуйте: схему (30 записів або більше, в усіх семи категоріях) і ранжування за ризиком.

Завдання 2. Запитання до власника продукту (25 хвилин)

Прочитайте `PRODUCT.md` так, як тестувальник читає вимогу на нараді, до появи будь-якого коду. Використайте питання проспективного

тестування з теоретичних відомостей: чи можете ви уявити кожне твердження на практиці; які слова вжито неточно; які приклади розв'язали б кожен неоднозначність; що найгірше може статися; як ви дізнаєтеся, що результат правильний.

- Складіть щонайменше вісім запитань. Кожне запитання називає речення опису, з якого воно походить, і елемент продукту або критерій якості, якого стосується.
 - Для кожного запитання зазначте, чи працюючий продукт уже відповів на нього тим чи іншим способом і як ви це з'ясували.
 - Тримайте цей список окремо від списку дефектів. Неоднозначна заява залишається питанням, доки на нього не відповів власник продукту.
- Зафіксуйте:* список запитань.

Завдання 3. Три чартери, три сесії (3 × 45 хвилин)

Напишіть три чартери у формі *Дослідити <ціль> за допомогою <ресурсів або техніки>, щоб з'ясувати <інформацію>* і проведіть по одній 45-хвилинній сесії для кожного. Використовуйте `templates/session_sheet.md`. Зафіксуйте розподіл часу сесії на Т-час, В-час і S-час. Якщо сесія відхилилася від чартера, після сесії перепишіть чартер так, щоб він описував зроблене насправді, і поясніть чому.

- **Сесія 1 — за ризиком.** Візьміть першу ділянку з вашого ранжування за ризиком у завданні 1.
- **Сесія 2 — за туром.** Оберіть один тур: *тур помилок* (усі помилки, яких може припуститися користувач, у вебформі та в API), *тур даними* (усі види даних, які приймає продукт, із межовими значеннями, повторюваними значеннями, порожніми та екстремальними значеннями) або *тур бізнес-правил* (усі правила, які ви можете знайти або вивести: знижки, доставка, ПДВ, статуси, залишки на складі). Проведіть тур і в API, і на вебсторінках.
- **Сесія 3 — за персоною.** Оберіть одну персону і залишайтеся в ролі протягом усієї сесії: *нетерплячий клієнт на повільному з'єднанні* (обмежте мережу до 3G у DevTools, натискайте двічі, оновлюйте сторінку під час очікування); *бухгалтер, який читає рахунок-фактуру* (чи кожне число в рахунку узгоджується із замовленням, записом клієнта та каталогом?); або *недобросовісний клієнт, який редагує URL-адреси* (що один клієнт може побачити або зробити з даними іншого клієнта через сторінки та через API?).

Зафіксуйте: три заповнені аркуші сесій. Нотатки мають дати читачеві змогу зрозуміти, що ви робили і чому: що покрили, що побачили, що вирішили пропустити.

Завдання 4. Таблиця оракулів (30 хвилин)

Перенесіть кожен знахідку з трьох сесій до `templates/oracle_table.md`. Для кожного рядка: спостереження своїми словами; патерн або патерни оракула; форма оракула (живий, явне знання, неявне знання, паралельний продукт, специфікація); чому це важливо і для кого; дефект це чи питання; сесія, з якої походить знахідка.

— Знахідки, що спираються лише на неоднозначну заяву, переходять до списку **курйозів і питань** із приміткою, яке було б їх розв'язання і хто міг би відповісти.

— Знахідка може мати більше одного оракула. Найсильніший називайте першим.

Зафіксуйте: таблицю оракулів і список курйозів. Кожен рядок таблиці має називати принаймні один оракул.

Завдання 5. Тур проти персони (15 хвилин)

Порівняйте сесію 2 і сесію 3. Що знайшов тур, чого персона знайти не могла, і навпаки? Який із двох підходів ви застосували б першим на наступному незнайомому продукті і чому?

Зафіксуйте: один абзац.

Завдання 6. Тестопридатність (25 хвилин)

Перелічіть перешкоди, на які ви натрапили під час тестування, — те, що коштувало S-часу або зробило спостереження неможливим, — і класифікуйте кожен словом-підказкою тестопридатності: спостережуваність, керованість, доступність інформації, доступність оракулів, пісочниця тощо. Потім запропонуйте **три конкретні поліпшення**, які розробник міг би реалізувати за день, кожне сформульоване як дія, і зазначте, яке тестування кожне з них уможливило б.

Зафіксуйте: таблицю перешкод і три пропозиції.

Вимоги до оформлення звіту

Подайте один документ `Lab1_<Прізвище>_<Варіант>.pdf` (або `.docx`), що містить у такому порядку:

1. Номер варіанта, використані облікові записи і спосіб скидання даних між сесіями.
2. Перші враження (завдання 0).
3. Схему покриття продукту з позначками (F)/(E) і ранжування за ризиком (завдання 1). Мапу думок подають як читабельне зображення.
4. Запитання до власника продукту (завдання 2).
5. Три аркуші сесій (завдання 3).
6. Таблицю оракулів і список курйозів (завдання 4).

7. Абзац «тур проти персони» (завдання 5).
8. Перешкоди тестопридатності та три пропозиції (завдання 6).
9. Висновок: яким видається стан продукту, наскільки ви в цьому впевнені і чого ви не тестували.

Використовуйте знімки екрана як доказ, а не як оформлення: додавайте знімок там, де він підтверджує спостереження (з видимим номером варіанта в нижньому колонтитулі), а не по одному на кожну сторінку. Спостереження через API подають як запит і відповідну частину відповіді.

Контрольні питання

1. Чому схема покриття продукту — не список тест-кейсів, і що було б втрачено, якщо б її було перетворено на такий список?
2. Які записи вашої схеми походять зі знайденого джерела, а які — з дослідження? Що пропустили знайдені джерела?
3. Яка з ваших знахідок спирається на найслабший оракул? Що його посилює б?
4. Оберіть один рядок вашого списку запитань. Чому це питання, а не дефект?
5. Що таке S-час, чому його звітують окремо і з чого складався ваш?
6. Що знайшла ваша сесія за персоною, чого не знайшов би тур?
7. Яка перешкода тестопридатності коштувала вам найбільше, і що саме ви попросили б у розробників?
8. Наскільки ви впевнені в стані продукту після трьох сесій, і що ви тестували б наступним?