

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
до лабораторних робіт з дисципліни
«Інженерія програмних продуктів та сервісів»**

Одеса: Одеська політехніка, 2025

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
до лабораторних робіт з дисципліни
«Інженерія програмних продуктів та сервісів»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
121 – Інженерія програмного забезпечення**

Затверджено
на засіданні кафедри ІПЗ
Протокол № 1 від 30.08.2024 р.

Одеса: Одеська політехніка, 2025

Методичні рекомендації до лабораторних робіт з дисципліни «Інженерія програмних продуктів та сервісів» для здобувачів першого (бекалаврського) рівня вищої освіти за спеціальністю 121 – Інженерія програмного забезпечення (освітньо-професійна програма: Інженерія програмного забезпечення – 2022) / Укл.: *В. В. Любченко*. – Одеса: Одеська політехніка, 2025. – 22 с.

Укладач: **Любченко В. В.**, докт. техн. наук, проф.

Методичні вказівки містять теоретичні відомості, які необхідні і достатні для виконання лабораторних робіт з дисципліни «Інженерія програмних продуктів та сервісів».

Призначаються для студентів денної форми навчання.

ЗМІСТ

Вступ.....	4
Лабораторна робота 1. Визначення ідеї продукту або сервісу.....	5
Лабораторна робота 2. Бізнес-планування.....	8
Лабораторні роботи 3-4. Специфікація вимог до програмного продукту.....	11
Лабораторна робота 5. Проектування досвіду користувача.....	15
Лабораторна робота 6. Планування бета-тестування.....	17
Лабораторна робота 7. Дорожня карта програмного продукту.....	19
Рекомендована література.....	21

Вступ

Зміст лабораторних робіт пов'язаний з питаннями розробки програмних систем. Основна увага приділяється питанням розуміння та аналізу потреб клієнтів, а також їх використання в процесі подальшого проєктування та розробки.

Лабораторні роботи виконуються командно, склад команди – до 3 студентів. Кожна команда самостійно визначає продукт або сервіс, який планується розробляти в ході лабораторних робіт. Продукт або сервіс має розроблятися від початку.

Лабораторна робота 1

Визначення ідеї продукту або сервісу

Мета роботи: формування навичок специфікації ідеї програмного продукту та його поведінки.

Теоретичні відомості

Одним з методів визначення ідеї продукту є дизайн-мислення (design thinking).

Дизайн-мислення – метод розробки продуктів та сервісів, орієнтованих на споживача. Дизайн-мислення завжди ставить до центру запит споживача і лише потім можливості технічної реалізації та економічні можливості.

Особливості методу дизайн-мислення:

- занурення в досвід споживача;
- сторонній підхід до визначення проблем;
- фокусування на персональних сценаріях поведінки і дій.

Щоб вирішити проблему, знайти ефективне рішення, потрібно спочатку провести дослідження і точно визначити проблему, потім сфокусуватися на конкретних «больових точках», згенерувати ідеї, вибрати кращі, створити прототип і протестувати його. Метод дизайн-мислення описує, як проходять ці стадії, на що потрібно звернути увагу і які використовуються інструменти на кожній з них.

Метод дизайн-мислення складається з шести етапів, кожен з яких проходить дві фази:

- дивергентну – пошук множини рішень однієї проблеми;
- конвергентну – точне використання інструкції для рішення задачі.

Етап 1: емпатія

Завдання етапу – розібратися, що роблять люди, навіщо вони це роблять, яким бачать навколишній світ, які мають цінності, а також дізнатися про їх емоційні і фізичні потреби. Неможливо зробити проєкт, якщо немає розуміння того, для кого він призначений.

Будь-яка розробка починається з болю і незадоволеності. Біль – привід поліпшити продукт або створити новий. Щоб знайти больові точки, необхідно провести дослідження. Тому стосовно дизайн-мислення, емпатію можна визначити як первинне дослідження, що передбачає безпосередню взаємодію з потенційними споживачами: інтерв'ю, спостереження, проведення експериментів.

Етап 2: фокусування

Єдиний спосіб знайти правильне рішення – правильно зрозуміти проблему. Фокусування відіграє критичну роль, так як дозволяє отримати явний вираз проблеми, яку потрібно вирішити, виходячи із зібраної інформації про життя людей. Визначення вузьконаправленої проблеми дозволяє розробити більше якісних рішень на етапі генерації ідей.

Режим фокусування також називають «точкою зору» (Point-of-View, POV). Це твердження, яке фокусує увагу на знахідках, зроблених в процесі спостереження і потребах певного споживача або образу персони, який складений з характеристик різних споживачів (композитний споживач). «Точку зору» отримують шляхом об'єднання трьох елементів: споживача, потреби і інсайту.

Хороша точка зору:

- фокусує увагу на проблемі;
- надихає команду;
- створює критерії оцінки різних ідей;
- допомагає команді приймати рішення незалежно і паралельно;
- допомагає розробляти конкретні концепції, уникаючи абстрактних.

Етап 3: генерація ідей

Генерація ідей – це режим, в якому потрібно зосередитися на розробці ідей і рішень. У плані процесу мислення цей етап виглядає як розгалуження на всілякі концепції та результати, які можуть бути застосовані як вирішення проблеми.

Для генерації ідей використовуються прототипування або побудова макета. Так само може використовуватися мозковий шторм, майндмеппінг, скетчі.

Щоб набрати якомога більше ідей, потрібно:

1. Виключити критику. Коли людина чує критику, у нього включається захисна реакція і страх блокує творчий потік.
2. Створити позитивну атмосферу. Розслабтеся, дружите, підтримуйте один одного.
3. Зібрати різнопрофільних гравців. Добре, коли в обговоренні беруть участь люди з різним досвідом.
4. Використовувати принцип «так і ...». Прийом, коли ви безумовно приймаєте ідею колеги і розвиваєте її.

Етап 4: вибір ідей

Щоб визначити життєздатні ідеї, їх потрібно пропустити через фільтри. Як вибирати ідеї:

- Сформулювати критерії відбору. Якщо критеріїв кілька, то у кожного має бути вага. Критерії дозволяють не втратити інноваційний потенціал, накопичений в режимі генерації ідей.
- Проголосувати (небезпека голосування може бути в тому, що якщо 6 - «за» і 4 - «проти», то умови дотримані, але ентузіазм майже половини команди дуже слабкий).
- Створити прототип. Прототипи дозволяють оцінити працездатність ідей.

Етап 5: прототипування

Прототипування – ітераційне створення макетів, які допоможуть знайти правильне рішення. На ранньому етапі прототипи повинні бути максимально простими. У міру того, як проводиться тестування і відбувається підтвердження гіпотези, прототипи стають складніше і дорожче.

Навіщо потрібні прототипи

- Щоб генерувати ідеї і вирішувати проблеми. Будуйте, щоб думати.
- Щоб спілкуватись. Одна картинка коштує тисячі слів, один прототип – тисячі картинок.
- Щоб почати розмову. Взаємодія з споживачем корисніша, якщо будується навколо чого-небудь. Прототип – можливість провести плідну розмову.
- Щоб зазнати невдачі швидко і дешево. Чим менше витрат, тим менше жалю.
- Щоб протестувати можливості. Простота дозволяє протестувати відразу кілька ідей, не добираючи лише один напрямок.

–Щоб керувати процесом розробки рішення. Виявлення змінюваних умов ділить велику проблему на багато дрібних і більш зручних для тестування.

Крім цього, для створення прототипів можна використовувати:

1. Сторітеллінг. Зробити презентацію ідеї, але у вигляді розповіді, більш людяну.
2. Макети з картону і паперу. Дозволяє створювати багато рішень швидко і дуже дешево.
3. Сторіборди. Намалювати покадровий сценарій використання.
4. Стоп-моушн. Якщо продукт ще не існує, то можна зняти ролик, щоб передати атмосферу і умови, в яких він буде працювати.
5. Бодістормінг. Зобразити сервіс за допомогою людей.

Етап 6: тестування

Тестування – отримання зворотного зв'язку щодо створених прототипів. Воно допомагає зрозуміти споживача і заглибитися в проблему, над якою ви працюєте.

Два підходу до тестування

- просто спостерігати: дати споживачу прототип і дивитися за його діями;
- поставити завдання: зробити якусь цільове дію.

Зміст роботи

Поспілкуйтеся з однокурсниками щодо потреб в продукті, який ви збираєтеся розробляти. Сформулюйте щонайменше 5 потреб потенційних споживачів в форматі «need – insight». Сформулюйте щонайменше 3 ідеї та підготуйте будь-який прототип рішення. Поспілкуйтеся з однокурсниками для отримання зворотного зв'язку.

Звіт з роботи має містити:

1. Назву програмного сервісу та склад команди, яка його розробляє.
2. Опис потреб потенційних користувачів, з якого зрозуміло, в чому полягає потреба, і яким чином її можна задовольнити.
3. Опис ідей щодо задоволення потреб користувачів.

Контрольні питання

1. Що таке дизайн-мислення і чому воно важливе в розробці продуктів і сервісів?
2. Які шість етапів включає метод дизайн-мислення?
3. У чому полягає основна мета етапу емпатії?
4. Як проводиться дослідження на етапі емпатії? Наведіть приклади методів.
5. Що таке "точка зору" (Point-of-View, POV), і як її сформулювати?
6. Які критерії визначають хорошу "точку зору"?
7. Назвіть основні принципи ефективної генерації ідей.
8. Як визначаються життєздатні ідеї на етапі вибору?
9. Які переваги надає прототипіювання у процесі розробки продукту?
10. Які два основні підходи використовуються для тестування прототипів?

Лабораторна работа 2

Бізнес-планування

Мета роботи: формування навичок аналізу потреб споживача.

Теоретичні відомості

Після того, як ідея продукту сформульована, необхідно забезпечити документацію бізнес-моделі майбутнього продукту.

Lean canvas, або «ощадливий шаблон» – це засіб ефективного запису найбільш важливих гіпотез відносно розвитку бізнесу. Метод застосовується переважно для роботи з SaaS-стартапами (*software as a service*).

Коли ідея і концепція придумані, мета будь-якого проєкту – сформулювати вимоги до MVP (minimum viable product), зрозуміти, як буде виглядати продукт на початковому рівні, на чому заробляти тощо. У цьому і допомагає Lean canvas.

Lean canvas – частина Lean-філософії, коли не робиться нічого зайвого, а робиться лише мінімально необхідне, що дозволить краще зрозуміти, як зробити наступний крок.

Схема Lean canvas та послідовність заповнення блоків показані на рис. 2.1.

Проблема Топ-3 проблеми	Рішення Топ-3 пропозиції	Унікальна пропозиція Чітка, зрозуміла, цікава, проста, цікава, яка пояснює, чому слід скористатися з продукту або послуги	Скрита перевага Неможна легко скопіювати чи купити	Споживачі Цільові клієнти
1	3	2	7	1
Ключові метрики Основні показники для вимірювання		Канали Шлях до споживача		
6		4		
Структура витрат Вартість залучення користувачів Вартість реалізації Хостінг ...			Потоки доходу Бізнес-модель (спосіб монетизації) Доход від одного користувача Валовий прибуток ...	
5			5	

Рисунок 2.1 – *Lean canvas*

Розглянемо кожний блок окремо.

1. Сегментація споживачів і визначення проблеми (Customer & Problem)

Головна увага Lean canvas сфокусована на споживачах (customer). Спочатку визначають всі сегменти споживачів, які приходять в голову, потім описують проблему. З усієї аудиторії необхідно виокремити групу ранніх клієнтів (early adopters). Це ті користувачі:

- для яких проблема є найгострішою;
- які вже намагалися вирішити проблему альтернативними способами.

Зверніть увагу, якщо на цьому етапі немає альтернатив на ринку або непрямих конкурентів – це не дуже добре. Можливо проблему перебільшено, раз ніхто ще не працював над вирішенням.

2. Унікальна пропозиція і рішення проблеми (UVP & Solution)

Унікальна пропозиція (UVP – Unique Value Proposition) – це фраза, яка визначає основну цінність продукту або сервісу, заради якої його будуть обирати для застосування.

Рішення (solution) – це набір характеристик, які забезпечують унікальну пропозицію. Наприклад, більш зручний user experience (досвід користувача), більше інформації і комфорту, тощо.

3. Канали контакту з користувачами (Channels)

Канал – це шлях, по якому клієнт приходить до вас. Як усталено, перший канал збуту – прямі контакти, коли ви домовляєтеся з клієнтами і показуєте їм продукт. Виходячи з суті продукту або сервісу, необхідно розуміти, якими будуть шляхи його просування.

Як правило, на початку застосовують 1-2 канали, найчастіше це контекстна реклама, SEO або SMM.

4. Прогнози бюджету: витрати і передбачуваний дохід (Cost Structure & Revenue)

Це своєрідний тест на розуміння бізнесу, для якого готується продукт або послуга. Як правило, тут мова іде про фіксовані витрати і про загальний регулярний дохід. Не враховуємо, скільки потрібно проінвестувати в розробку продукту, і не чіпаємо змінні витрати на кшталт комісій.

Важливо побачити, що потік доходу реалістичний і він не губиться на тлі витрат.

5. Ключові метрики

Є два підходи до ключових метрик (KPI).

Перший підхід – натуральний показник KPI. KPI повинен утворюватися сам по собі, як результат роботи бізнесу. Він не може бути грошовим, оскільки грошовий показник може регулюватися штучно – через підвищення цін або через урізання витрат.

Другий підхід – так звані піратські метрики, або концепція AARRR:

– Acquisition – залучення (вираховується вартість залучення клієнтів, щоб конкурувати на ринку, вона повинна бути мінімальною);

– Activation – активація (мотивація повторно зайти на сторінку; враховуються час, проведений за переглядом сайту, кліки і переглянуті сторінки);

– Retention – утримання (email-маркетинг, push-повідомлення);

– Referral – мотивація постійних клієнтів (репости, реферальні посилання);

– Revenue – дохід (фінансові метрики; ефективність перевірити просто: чим більше користувачів перетворюються в постійних, тим вище прибуток).

6. Прихована перевага (Unfair advantage)

Суть прихованої переваги (unfair advantage) в тому, що має бути закладено якусь довгострокову перевагу продукту або сервісу. Адже якщо він підтвердить базову гіпотезу і почне рости, це помітять великі гравці на ринку, яким нічого не варто скопіювати ідею.

Коли складено Lean canvas, у вас перед очима цілісна бізнес-модель. Щоб можна було рухатися далі, необхідно визначити ризиковані елементи. Існують три ключові траєкторії ризику:

–ризик споживача: невірно обрана аудиторія або сегмент ранніх клієнтів, невірно обрані канали, за якими повинні дізнаватися про продукт;

–ризик продукту: помилка у формуванні продукту, переоцінили проблему, придумали погане рішення, не так його позиціонують або неправильно оцінюють успіх;

–ризик ринку: недооцінили або переоцінили зовнішні параметри, альтернативи конкурентів занадто сильні і немає місця на ринку, не зароблять грошей або дорого коштуватиме вихід на ринок.

Зміст роботи

Складіть Lean canvas для свого продукту або сервісу. Визначте, який ризик є найбільш суттєвим, та перевірте гіпотези, на яких базуються припущення, що закладені до Lean canvas.

Звіт з роботи має містити:

1. Lean canvas сервісу, що розробляється, та опис найбільш суттєвого ризику.
2. Визначення ключового результату роботи сервісу, 2-3 найбільш важливих функціональних можливостей, кінцевого користувача, покупця (якщо відрізняється від користувача), ринкової ніши, способу залучення клієнтів та ключового повідомлення для них.

Контрольні питання

1. Що таке Lean canvas, і які її основні переваги для розробки бізнес-моделі?
2. Яка роль сегментації споживачів у Lean canvas? Що таке «ранні клієнти» (early adopters)?
3. Як визначити основні проблеми споживачів під час роботи над Lean canvas?
4. Що таке унікальна пропозиція цінності (Unique Value Proposition, UVP), і як її сформулювати?
5. Яким чином характеристики продукту забезпечують вирішення проблем клієнтів?
6. Які види каналів контакту з клієнтами використовуються на ранніх етапах проєкту?
7. Як визначити структуру витрат і потоки доходів у Lean canvas?
8. Що таке піратські метрики AARRR, і як вони допомагають оцінити успішність бізнесу?
9. У чому полягає прихована перевага (Unfair advantage), і чому вона важлива для довгострокового успіху продукту?
10. Які три ключові траєкторії ризику можуть виникнути під час реалізації бізнес-моделі, і як їх враховувати?

Лабораторні роботи 3-4

Специфікація вимог до програмного продукту

Мета роботи: формування навичок формалізації вимог до програмних продуктів.

Теоретичні відомості

User Story (історія користувача) – це коротке формулювання наміру, який описує, що система повинна робити для користувача:

– вони не є детальним описом вимог (тобто того, що система повинна б робити), а являють собою скоріше обговорюване подання намірів (потрібно зробити щось на зразок цього);

– вони є короткими, легко читаються, зрозумілі розробникам, стейкхолдерам і користувачам;

– вони описують невеликі інкременти цінної функціональності, яка може бути реалізована в рамках декількох днів або тижнів;

– вони відносно легко піддаються оцінюванню, таким чином, зусилля, необхідні для реалізації, можуть бути швидко визначені;

– вони не займають громіздких документів, а скоріше організовані в списки, які легше впорядкувати і перевпорядкувати по ходу надходження нової інформації;

– вони не деталізовані на самому початку проєкту, а вже більш детально розробляються «точно в строк», уникаючи таким чином занадто ранній визначеності, затримок в розробці, нагромадження вимог і надмірного обмеження формулювання рішення.

Текст самої *user story* повинен пояснювати роль та дії користувача в системі, його потреби і вигоди, які він отримає після того, як історія трапиться:

Як <роль / персонаж користувача>, я <щось хочу отримати>, <з такою-то метою>.

Наприклад:

– Як покупцю мені зручно шукати книги по жанрам, щоб швидко знайти ті, які я люблю читати.

– Як покупець я, відбираючи книги для покупки, хочу класти відразу кожную в кошик.

– Як керуючий по випуску нової продукції я хочу мати можливість відстежувати покупки наших клієнтів, щоб бути в курсі, які книги їм можна пропонувати.

Важливо описувати історію на рівні «що робить», а не «як». Це головне в історії, опишіть проблему, а не її вирішення.

Відмовитися від формулювання "щоб". Це корінь зла. ТАК, для якихось історій можна вказати цінність історії в такому форматі, але не для більшості.

При формулюванні мети краще перейти з поняття цінності (value) на вплив (impact). Історія не обов'язкова повинна мати цінність, але обов'язково повинна впливати на актора, що зазначений в історії. А вже цей вплив веде в кінцевому підсумку до мети, яка має цінність.

Історію користувача можна оцінити за критеріями INVEST:

– Independent: скорочення (або запобігання) залежностей полегшує планування;

- Negotiable: деталі додаються через обговорення та співпрацю;
- Valuable;
- Estimable: занадто великі або занадто невизначені не можуть бути оцінені;
- Small: можуть бути реалізовані командою менше ніж за тиждень;
- Testable: критерії прийняття добре визначені.

Use Case (варіант використання) – це сценарна техніка опису взаємодії. Варіант використання фіксує угоду між учасниками системи про її поведінку. Він описує поведінку системи при її відповідях на запит одного з учасників, званого основною дійовою особою, в різних умовах. У загальному випадку, за допомогою *Use Case* може описуватися взаємодія двох або більше учасників, яка має конкретну мету.

У розробці програмного забезпечення (ПЗ) цю техніку часто застосовують для проектування і опису взаємодії користувача і системи, тому назву *Use Case* часто сприймають як синонім вимоги людини-користувача до вирішення певної задачі в системі. Опис вимог до системи у вигляді опису контексту і послідовності дій користувача допомагає сформулювати набір функціональних вимог, який буде забезпечувати повноту і ненадмірність вимог.

Наприклад, опишемо варіант використання, що відповідає процесу реєстрації на рейс.

UC1 Реєстрація пасажирів на рейс

Система: Система реєстрації пасажирів на рейс

Основна діюча особа: Пасажир

Мета: Зареєструватися на рейс

Тригер: Пасажир вирішує зареєструватися на рейс і заходить на сторінку реєстрації сайту

Результат: Інформація про реєстрацію пасажирів збережена. У пасажирів є посадковий талон

Основний потік подій			
№ кроку	Дійова особа	Крок	Коментар
1	Система	Запитує прізвище і код бронювання	
2	Пасажир	Вводить прізвище і код бронювання	Код бронювання: 7 символів, заголовні латинські та цифри. Наприклад: MTDTCS1
3	Система	Перевіряє, що придбаний квиток на цей рейс на ім'я цього Пасажирів	
4	Система	Зберігає інформацію про реєстрацію Пасажирів на рейс	
5	Система	Підтверджує Пасажирів, що він зареєстрований на рейс	
6	Система	Виводить для друку посадковий талон	
7	Пасажир	Відправляє посадковий талон на друк	

Велику частину реального повного *Use Case* становить не основний потік, а альтернативні потоки, які дозволяють задавати розгалуження потоку, цикли і обробляти «неправильні» події.

За допомогою *Use Case* можна виявити багато функціональних вимог: практично кожен рядок *Use Case* є окремою функціональною вимогою. Ми бачимо, які функції повинні виконуватися разом, а отже, у нас є можливість виставляти пріоритети реалізації цих вимог так, щоб вони були готові в один час.

Важливі моменти:

– Використовуйте мінімальну кількість слів і пунктів, необхідних для однозначного розуміння сценарію. З дуже довгими сценаріями, з великою кількістю розширень, працювати вкрай незручно.

– Якщо в двох або більше сценаріях повторюється однаковий набір кроків, є сенс винести ці кроки в окремий сценарій, і посилатися на них.

– Перелік повідомлень, які система видає користувачу, стандартні тексти електронних листів тощо зручно розташувати в єдиному місці в документі, і посилатися на потрібний пункт з різних Use Case, тому що повідомлення в сценаріях часто дублюються.

Під час формулювання вимог слід звернути увагу на вимоги до якості. *Будинок якості* є елементом технології розгортання функцій якості (Quality Function Deployment – QFD). Розгортання функцій якості є системним підходом до проектування, заснованим на чіткому розумінні бажань споживачів. Застосування цієї технології дозволяє перевести побажання споживача в технічні характеристики виробу.

Основною метою розгортання функцій якості є переклад суб'єктивних критеріїв якості в набір технічних характеристик, які можна виміряти і які можна застосовувати для проектування і розробки програмних продуктів та сервісів. QFD є одним з ефективних методів «розстановки пріоритетів» в процесі створення продукції. Будинок якості документує цей процес в зручній формі.

Процес складання будинку якості складається з п'яти кроків:

- 1) визначаються та записуються вимоги клієнтів;
- 2) вимоги перетворюються на технічні та вимірювані характеристики програмного продукту;
- 3) клієнти заповнюють кореляційну матрицю, визначаючи міцність зв'язку між різними вимогами та технічними характеристиками продукту;
- 4) на основі опитувань визначаються пріоритети для заявлених вимог клієнтів;
- 5) пріоритети технічних характеристик продукту визначаються шляхом підсумовування результатів множення пріоритетів вимог замовника та ваги технічних характеристик продукту.

Розглянемо приклад.

Технічна специфікація продукту					
1	2	Вимоги користувача			4
	Виконання за розумний час	Простота реалізації	Генерація інтересу користувачів	Пріоритети для вимог користувача	
	1 On-line допомога	9	6	9	6
	2 Об'єднання з каталогом	8	8	8	7
	3 Включення бібліотеки стандартів ASME	6	8	7	3
	4 Регулярна добре-визначена стратегія релізів	9	9	6	9
5		209	197	185	

Рисунок 3.1 – Будинок якості

Зміст роботи

Формалізувати вимоги рівня бізнесу за допомогою User Stories. Формалізувати вимоги рівня користувача за допомогою Use Cases. Визначити впливові технічні характеристики продукту та побудувати будинок якості для розроблюваного програмного продукту. Зробити висновки з отриманих оцінок щодо подальшої розробки продукту.

Звіт з роботи має містити:

1. Перелік User Stories програмного сервісу.
2. Перелік Use Cases програмного сервісу, які згруповано відповідно до User Stories з метою візуалізувати зв'язок між вимогами рівня бізнесу та вимогами рівня користувача.
3. Перелік важливих вимог за якістю до програмного сервісу з обґрунтуванням їх вибору.
4. Будинок якості програмного сервісу.
5. Висновки з аналізу бинку якості щодо подальшої розробки продукту.

Контрольні питання

1. Що таке User Story, і як вона відрізняється від детального опису вимог?
2. Які основні елементи повинен містити текст User Story?
3. Наведіть приклад правильно сформульованої User Story.
4. Що означає аббревіатура INVEST, і як ці критерії застосовуються до User Story?
5. Чому важливо, щоб User Story була незалежною та тестованою?
6. Що таке Use Case, і для чого він використовується у розробці програмного забезпечення?
7. Які основні елементи включає опис Use Case?
8. Чим відрізняються основний потік подій Use Case від альтернативних потоків?
9. У чому полягає різниця між User Story та Use Case?
10. В яких випадках краще використовувати User Story, а в яких Use Case?
11. Що таке Будинок якості, і як він пов'язаний з процесом розгортання функцій якості (QFD)?
12. Які п'ять кроків необхідно виконати для створення Будинку якості?

Лабораторна робота 5

Проектування досвіду користувача

Мета роботи: ознайомлення з підходами UI/UX дизайну.

Теоретичні відомості

Під час проектування інтерфейсу користувача вкрай важливо розуміти, хто саме буде користатися з вашого продукту або послуги. Один з ефективних інструментів для цього – виділення так званої персони або профілю представника цільової аудиторії.

Персона – це опис типового користувача певного продукту. Портрет персони враховує:

- соціально-демографічні дані – вік, стать, посаду, освіту, сімейний стан;
- цілі, потреби, мотивація, очікування (допомагає пріоритезувати властивості продукту);
- больові точки (проблеми);
- мрії;
- аргументи, які утримують від покупки (заперечення);
- способи користування програмними продуктами та сервісами (де, коли і з яких пристроїв).

Використання персон дозволяє піти від абстрактного підходу і дати людям, які займаються розробкою нового продукту, детальний портрет кінцевого споживача. Після створення персони стає зрозуміло, які функції потрібні користувачам, чому вони їх хочуть, в якому вигляді їх подати, а також з якими труднощами вони можуть зіткнутися.

Дослідження показали, що однією з причин, через які споживач відмовляється від товару або послуги є труднощі з розумінням того, як користуватися цим товаром/послугою. Щоб зафіксувати досвід користувача під час взаємодії з продуктом застосовують карти подорожей користувачів (User Journey Map).

Карта подорожей користувачів – це візуалізація взаємовідносин людини з продуктом / брендом у часі та на різних каналах.

Карти подорожей користувачів зазвичай представлені як часова шкала, на якій зазначені всі точки контакту між користувачем і продуктом. Ця часова шкала містить інформацію про всі канали, які користувачі використовують для взаємодії з продуктом. Карта подорожей користувачів робить наочним те, як користувач взаємодіє з продуктом, і дозволяє розробникам бачити продукт з точки зору користувача. Це забезпечує орієнтований на користувача підхід до дизайну продукту або сервісу, що призводить до кращого досвіду користувача (user experience – UX). Шаблон карти подорожей показано на рис. 4.1.



Рисунок 4.1 – Шаблон карти подорожей користувача

Процес складання карти подорожей користувача містить 8 кроків:

1. Виберіть область застосування

Обсяг карти подорожі користувача може варіюватися від карти високого рівня, яка показує досвід роботи з продуктом в цілому, до більш детальної карти, яка фокусується на одній конкретній взаємодії (наприклад, оплаті рахунку).

2. Створіть персону користувача

3. Визначте сценарій та очікування користувачів

Сценарій описує ситуацію, на яку спрямована карта подорожі. Він може бути реальним або передбачуваним, як основу для вибору сценаріїв слід застосовувати варіанти використання. Також важливо визначити, що очікує персона щодо взаємодії.

Наприклад, можливий сценарій – замовлення таксі за допомогою мобільного застосування з очікуванням отримати машину за 5 хвилин або менше.

4. Створіть список точок контакту

Точки контакту – це дії користувачів та взаємодія з продуктом або сервісом. Важливо визначити всі основні точки контакту та всі канали, пов'язані з кожною точкою контакту.

5. Врахуйте намір користувача

Для кожної подорожі користувача важливо розуміти:

- мотивацію – чому користувач намагається це зробити;
- канали – де відбувається взаємодія;
- дії – реальна поведінка та кроки, які вживають користувачі;
- больові точки – з якими проблемами стикаються користувачі.

6. Ескіз подорожі

Зберіть разом всю наявну інформацію та намалюйте подорож у форматі покрокової взаємодії. Кожен крок демонструє досвід роботи користувача із послугою або продуктом чи іншою людиною.

7. Розгляньте емоційний стан користувача під час кожного кроку взаємодії

Візуалізуючи емоційні «злети» та «падіння» досвіду користувача, можна визначити сфери досвіду, які потребують доопрацювання.

8. Підтвердження та уточнення подорожі користувача

Зміст роботи

Описати цільову аудиторію за допомогою опису щонайменше однієї персони. Показати карти подорожей для описаної персони.

Звіт з роботи має містити:

1. Опис персони (персон), що представляє цільову аудиторію.
2. Карта подорожі для описаної персони.
3. Визначення ага-моментів для користувачів на карті подорожі.

Контрольні питання

1. Що таке персона, і які її ключові характеристики враховуються під час створення?
2. Як врахування больових точок і мрій користувача впливає на розробку продукту?
3. Що таке карта подорожей користувача, і яку роль вона відіграє в дизайні продуктів?
4. Які основні елементи входять до карти подорожей користувача?
5. Який обсяг може мати карта подорожей користувача, і від чого це залежить?
6. Наведіть приклад сценарію, який можна використати для створення карти подорожей.
7. Що таке точки контакту, і чому вони важливі у складанні карти подорожей користувача?
8. Яким чином врахування емоційного стану користувача допомагає покращити продукт?
9. Як візуалізація взаємодії користувача з продуктом дозволяє виявити слабкі місця в UX?
10. Чому важливо підтверджувати та уточнювати карту подорожей користувача після її складання?

Лабораторна робота 6

Планування бета-тестування

Мета роботи: формування умінь планування бета-тестування програмного продукту.

Теоретичні відомості

Створення якісного продукту або сервісу має бути метою розробників. Терміном *якість програмного забезпечення* позначаємо сукупність властивостей, що визначають спроможність забезпечення задовольнити потреби споживачів.

Якість програмного забезпечення – це відносне поняття, що має сенс тільки при врахуванні реальних умов його застосування. Тому надзвичайно цінним в комплексі заходів забезпечення якості є бета-тестування.

Бета-тестування є одним з типів тестування приймальних випробувань, що додає цінність продукту, оскільки кінцевий користувач перевіряє продукт на функціональність, зручність використання, надійність і сумісність. Оскільки бета-тестування відбувається на стороні кінцевого користувача, це не може бути контрольованою дією.

Бета-тестування надає такі переваги:

1. Бета-тестування забезпечує повний огляд істинного досвіду, накопиченого кінцевими користувачами при роботі з продуктом.
2. Він виконується широким колом користувачів, і причини, через які продукт використовується, сильно розрізняються. Фактичне сприйняття кінцевих користувачів ясно показує, чому вони мають потребу в цьому продукті, і як вони будуть його використовувати.
3. Бета-тестування забезпечують перевірку сумісності продукту в реальних умовах, оскільки тут використовується велика кількість реальних платформ для тестування на широкому спектрі пристроїв, операційних систем, браузерів тощо.
4. Оскільки широкий діапазон платформ, які фактично використовують кінцеві користувачі, може бути недоступний для внутрішньої групи тестувальників, бета-тестування також допомагає виявити приховані помилки і прогалини в кінцевому продукті.

Бета-тестування виконується, коли продукт буде готовий не менше ніж на 90-95%. Зазвичай тривалість бета-тестування – це один або два тестових цикли з 4-6 тижнями в циклі. Вона збільшується тільки при додаванні нової функції або при зміні основного компонента.

Бета-тестування може бути виконано кількома способами, але в цілому існує п'ять різних етапів.

1. Планування

Визначте цілі заздалегідь. Це допомагає в плануванні кількості користувачів, що беруть участь в тестуванні, і тривалості, необхідної для завершення і досягнення цілей.

2. Набір учасників

В ідеалі участь в тестуванні може брати будь-яка кількість користувачів, але через бюджетні обмеження проєкту слід встановити мінімальну та максимальну кількість залучених користувачів. Як правило це 50-250 користувачів в середньому.

3. Запуск продукту

–Інсталяційні пакети повинні бути поширені серед учасників. В ідеалі, поділіться посиланням, звідки їх можна завантажити і встановити.

–Поділіться посібником користувача, відомими проблемами, об'ємом тестування тощо.

–Ознайомте учасників зі способами реєстрації помилок.

4. Збір і оцінка відгуків

–Помилки, відгуки та пропозиції учасників обробляються.

–Відгуки аналізуються і оцінюються потреби клієнтів при використанні даного продукту.

–Вважається, що рекомендується поліпшити продукт в наступних версіях.

5. Закриття

–Як тільки зафіксовано, що всі функції працюють, помилок не виникає, критерії виходу виконуються, можна завершити етап бета-тестування.

–Розподіліть винагороди / заохочення учасникам відповідно до плану і офіційно подякуйте їм (це допомагає в подальшому бета-тестуванні продукту, ви отримуєте набагато більше відгуків, пропозицій тощо).

Зміст роботи

Спланувати виконання бета-тестування. Підготувати форму для збору фідбеків (з використанням Google Forms). Визначити спосіб висловлювання подяки учасникам бета-тестування.

Звіт з роботи має містити:

1. Сформульовану мету виконання етапу бета-тестування
2. План виконання бета-тестування
3. Посилання на форму для збирання відгуків залучених тестувальників.
4. Опис форми подяки та розрахунок витрат (часових та грошових) на неї.

Контрольні питання

1. Що таке якість програмного забезпечення, і чому вона є відносним поняттям?
2. Які властивості визначають якість програмного забезпечення?
3. Що таке бета-тестування, і яку роль воно відіграє в забезпеченні якості програмного забезпечення?
4. Які основні переваги надає бета-тестування?
5. Чому важливо визначити цілі бета-тестування на етапі планування?
6. Які фактори впливають на кількість учасників бета-тестування?
7. Які ключові дії виконуються на етапі запуску продукту під час бета-тестування?
8. Як забезпечується збір та обробка відгуків від учасників бета-тестування?
9. Як аналіз відгуків кінцевих користувачів допомагає вдосконалити продукт?
10. Чому важливо оцінювати потреби клієнтів у реальних умовах використання продукту?
11. Які критерії свідчать про успішне завершення етапу бета-тестування?
12. Як винагородження учасників тестування впливає на їх залученість у майбутніх проєктах?

Лабораторна робота 7

Дорожня карта програмного продукту

Мета роботи: формування умінь складати дорожньої карти для продукту.

Теоретичні відомості

Дорожня карта продукту – це стратегічний план на високому рівні, який забезпечує довгостроковий прогноз розвитку продукту. Вона забезпечує безперервність мети, встановлює очікування, вирівнює інформованість зацікавлених сторін та сприяє визначенню пріоритетів.

Слід звернути увагу на те, щоб дорожня карта не була переобтяжена деталізованим описом функцій продукту, що перетворить її на тактичний інструмент планування. Для тактичного планування слід застосовувати інші інструменти (наприклад, беклог продукту).

Типова форма дорожньої карти показана на рис. 6.1.

Дата Дата випуску або інтервал часу	Дата/інтервал	Дата/інтервал	Дата/інтервал	Дата/інтервал
	Коли буде доступним реліз?			
Назва Назва продукту або версія релізу	Назва/версія	Назва/версія	Назва/версія	Назва/версія
	Як він буде називатися?			
Мета Причина випуску нової версії	Мета	Мета	Мета	Мета
	Чому він має бути розроблений?			
Властивості Функції високого рівня необхідні для досягнення мети	Функції	Функції	Функції	Функції
	Які в нього ключові властивості?			
Метрики Метрики/KPI для оцінювання ступеня досягнення мети	Метрики	Метрики	Метрики	Метрики
	Як ми будемо знати, що мета досягнута?			

Рисунок 6.1 – Дорожня карта продукту

Перший рядок дорожньої карти містить дату або часовий інтервал майбутніх випусків (релізів). Можна зазначати як певну дату, наприклад, 1 вересня, так і період, наприклад, перший чи другий квартал.

У другому рядку зазначають ім'я або версію випусків, наприклад, iOS 12.3.1 або Windows 10.

В третьому рядку зазначають мету кожного випуску, причину, чому його варто розробити та запустити. Прикладами цілей можуть бути придбання або активація користувачів, утримання користувачів шляхом покращення користувацького досвіду або прискорення розвитку шляхом усунення технічних труднощів. Робота з цілями зсуває увагу від окремих функціональних можливостей до узгодження стратегічних продуктових рішень. Команда розробників та зацікавлені сторони мають узгодити цілі.

Обов'язково цілі випусків мають бути визначені до визначення функцій.

В четвертому рядку описують функції, необхідні для досягнення мети. Функції є засобом досягнення мети, але не самоціллю: вони служать для створення цінності. Спробуйте обмежити кількість функцій для кожного випуску до трьох, але не вказуйте більше п'яти. Утримуйтеся від деталізації особливостей та зосередьтеся на можливостях продукту, необхідних для досягнення мети. Дорожня карта продукту має бути планом високого рівня.

В останньому рядку зазначаються метрики або ключові показники ефективності (KPI), які допомагають визначити, чи була досягнута мета. Слід переконатися, що вибрані показники

дозволяють вимірювати, в якій мірі досягнуто мету.

Дорожня карта продукту є потужним способом зробити фокусування на стратегічних цілях, а не тактичних задачах (функціях продукту). Це робить її інструментом стратегічного планування. Показники, що надаються інструментом, гарантують, що цілі є реальними і вимірюваними, а не високими і нечіткими ідеями.

Також слід постійно тримати фокус на меті під час розробки свого програмного продукту або сервісу та виведення його на ринок. З цією метою застосовують короткий опис стратегії виведення програмного продукту на ринок, який зазвичай містить визначення:

- основного результату продукту SaaS;

- найважливіших 2-3 функцій;

- ага-моменту;

- другорядних 4-7 функцій;

- кінцевого користувача;

- Ппокупця (якщо інший);

- ринку;

- ніши;

- маркетингових каналів збуту;

- канал конверсії клієнтів;

- заходів з утримання клієнтів;

- основного повідомлення бренду.

Як можна побачити, всі компоненти стратегії відпрацьовуються в життєвому циклі програмного продукту. Тому явне формулювання стратегії та забезпечення її доступності для всіх учасників процесу дає можливість перевіряти, чи не втрачене головне в процесі роботи над деталями.

Зміст роботи

Розробити дорожню карту продукту на наступний рік. Пересвідчитися в тому, що зазначені в карті цілі узгоджуються з потребами споживачів, які були виявлені на початку виконання лабораторних робіт. Сформулювати стратегію виведення продукту на ринок.

Звіт з роботи має містити:

1. Дорожню карту продукту
2. Стратегію виведення продукту на ринок

Контрольні питання

1. Що таке дорожня карта продукту, і яке її основне призначення?
2. Чому важливо уникати деталізації функцій у дорожній карті продукту?
3. Які основні елементи має містити дорожня карта продукту?
4. Як у дорожній карті визначаються цілі випусків, і чому це важливо?
5. Як зазначаються часові інтервали майбутніх випусків у дорожній карті?
6. Чому рекомендується обмежувати кількість функцій для кожного випуску до трьох-п'яти?
7. Що таке KPI, і яку роль вони відіграють у дорожній карті продукту?

8. Як обрати метрики, які дозволяють вимірювати досягнення цілей?
9. Які основні компоненти включає стратегія виведення продукту на ринок?
10. Чому важливо забезпечити доступність стратегії для всіх учасників процесу розробки?
11. Як дорожня карта продукту допомагає зосереджуватись на стратегічних цілях, а не на тактичних задачах?
12. Яким чином узгодження цілей між розробниками та зацікавленими сторонами впливає на успішність проєкту?

Рекомендована література

1. Dayle Miracle. Software Engineering: Principles and Practice (Kindle Edition). 2023. 649 p.
2. Wiegers K., Hokanson C. Software Requirements Essentials: Core Practices for Successful Business Analysis. Addison-Wesley Professional. 2023. 208 p.
3. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. – O'Reilly Media, 2020. – 432 p.
4. Anthony Conta. The Art and Science of UX Design: A step-by-step guide to designing amazing user experiences. New Riders. 2023. 464 p.

Навчальне видання

Методичні рекомендації до лабораторних робіт з дисципліни «Інженерія програмних продуктів та сервісів» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 121 – Інженерія програмного забезпечення

Укладач: Віра Вікторівна Любченко