1**)**

Я обрала робот-пилосос, тому що це побутовий пристрій, який поєднує апаратну частину та програмне забезпечення, це робить його цікавим об'єктом для тестування.

Перевірки, які я провірила:

Запуск пристрою: перевірила чи вмикається кнопкою та через мобільний додаток.

Спостерігала, як орієнтується в квартирі — чи уникає перешкод, чи правильно будує маршрут.

Налаштувала прибирання на певний час і перевірила, чи запускається вчасно.

Рівень шуму суб’єктивно оцінила,наскільки гучно працює в різних режимах.

Повернення на базу перевірила, як повертається на зарядну станцію після прибирання або при низькому заряді.

Робота на різних поверхнях: килим, плитка, ламінат.

Чому саме ці перевірки:

Вони допомагають оцінити, наскільки пристрій стабільний, зручний та ефективний у реальних умовах.

2)

Валідація - ми перевіряємо чи продукт відповідає вимогам, очікуванням користувачів. Відповідає на питання “ чи правильний продукт ми робимо “

Верифікація - ми перевіряємо чи продукт правильно зроблений з точки зору документації, внутрішніх процесів чи відповідає вимогам та критеріїв стандарту

Відповідає на питання “чи правильно ми робимо продукт “

Наприклад :

Верифікація це рецепт по якому ми повинні спекти пиріг

Валідація - це сам пиріг, який ми куштуємо і розуміємо це те що ми хотіли , і це те що від нас вимагали

Верифікація

У технічних характеристиках зазначено, що пилосос працює 90 хвилин без підзарядки. Тестування показує, що він дійсно працює 90 хвилин — отже, верифікація пройдена.

В інструкції вказано, що він повинен повертатися на базу при низькому заряді — перевірка показує, що він це робить.

Валідація

Користувач очікує, що робот не застрягатиме під диваном — перевіряємо, як він справляється з меблями.

Очікування, що підлога буде чистою після першого проходу — тестуємо якість прибирання.

Користувач хоче, щоб пилосос працював тихо, бо вдома є дитина — оцінюємо рівень шуму.

| Вид компанії | № | Плюси | Мінуси |
| --- | --- | --- | --- |
| Product | 1 | Можливість сконцентруватися на одному завданні, великі проекти , стабільність | Ти можеш бути вузькоспеціалізованим, такі компаніі можуть розпадатися і потім можливо треба перевчатися на інше направлення |
| Academ | 2 | можливість навчати інших, підтримка новачків, робота з людьми | обмежені можливості технічного росту. |
| Startap | 3 | інновації, швидке зростання, можливість впливати на продукт і процеси. | високий ризик, нестабільність, часте перевантаження. |
| Outsourse | 4 | різноманіття проєктів, швидке набуття досвіду, робота з іноземними замовниками. | менше впливу на кінцевий продукт, часто робота під тиском дедлайнів. |
| Outstaf | 5 | гнучкість, можливість працювати в міжнародних командах, досвід у різних доменах. | відсутність залученості до продукту, можлива нестабільність у проектах. |
| Progect | 6 | Допомагають знайти роботу, мають великий пул клієнтів | Відсутність стабільності  Менше розвитку як спеціаліста в середині компанії  Обмежений вплив на сам продукт або процеси |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

4)

Приклад невдалої валідації (з мого досвіду):

Працюючи в аптеці, я стикнулася з ситуацією, коли впровадили нову програму лояльності. За нею покупці мали отримувати знижку після сканування спеціальної картки. Програму встановили, ми почали працювати — але коли покупець давав картку, ми її сканували, а знижка не застосовувалась.

Тобто технічно система працювала: картка зчитувалась, програма відкривалась — але кінцева потреба користувача (отримати знижку) не виконувалась.

Це приклад невдалої валідації, бо продукт не відповідав очікуванням користувачів — знижка мала нараховуватись автоматично, але цього не відбувалось.

Приклад невдалої верифікації (з мого досвіду):

У фармацевтичній сфері буває так, що певні препарати (наприклад, БАДи або неефективні лікарські засоби) просуваються дуже активно — завдяки рекламі, хорошому дизайну упаковки та довірі з боку покупців.

Такі препарати можуть не проходити достатню кількість наукових досліджень або не мати доказаної ефективності. Тобто технічно вони не відповідають медичним стандартам і вимогам до лікувального засобу, але продаються як «корисні» чи навіть «лікувальні».

Це і є проблема верифікації — продукт не відповідає вимогам і стандартам, не має реальної дії, але потрапляє на ринок і виконує лише психологічну функцію (ефект плацебо).

Люди можуть вірити, що їм краще, але з точки зору документації та процесів — продукт не був достатньо перевірений і не виправдав заявлену функцію.

Раннє тестування

помилки дешевше й легше виправити, якщо знайти їх на ранньому етапі.

Приклад з досвіду: коли ми в аптеці отримали нову систему знижок, ніхто не протестував її до запуску — у результаті система не застосовувала знижки, що викликало проблеми з клієнтами. Якби перевірили раніше — уникнули б негативного досвіду .

Відтворюваність

Тести повинні давати однаковий результат при повторному виконанні, інакше складно зрозуміти, чи справді проблема вирішена.

Приклад з досвіду: одного разу система знижок «працювала через раз» — іноді скидка застосовувалась, іноді ні. Це ускладнювало пошук проблеми й змушувало сумніватися в її надійності.

Домашня робота номер 3

1)

| Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільно |
| --- | --- | --- | --- |
| Waterfall (Каскадна модель) | Сильні сторони: чітка структура, кожен етап виконується послідовно; підходить для проєктів із заздалегідь визначеними вимогами. | неможливість легко вносити зміни; тестування відбувається лише в кінці, тому помилки можуть виявитися пізно | у державних або інженерних проєктах, де важлива документація, точність і передбачуваність (наприклад, будівництво, банківські системи). |
| Agile | висока гнучкість; швидка адаптація до змін; постійний зворотний зв’язок з клієнтом; швидкий випуск результатів. | вимагає активної участі замовника; важко чітко спрогнозувати строки та бюджет | у стартапах, веб- і мобільній розробці, де проєкт постійно змінюється або розвивається. |
| Scrum | чіткий процес; швидке реагування на зміни; прозорість завдяки щоденним зустрічам і звітності; підвищена командна залученість. | : потребує досвідченої команди; можливе перевантаження мітингами; не завжди підходить для нестабільних команд. | у продуктових IT-компаніях, які активно розвивають нові функції, або працюють над складними системами з регулярними релізами. |
| Kanban | проста у впровадженні; візуальний контроль за роботою; гнучке управління пріоритетами; підходить для підтримки. | відсутність чітких термінів; може виникати плутанина без чіткого контролю над завантаженістю. | у службах підтримки, технічному обслуговуванні, DevOps — там, де задачі надходять постійно, а не пакетами |
|  |  |  |  |
|  |  |  |  |

2

Waterfall

Сильна сторона: Чіткість етапів – легко планувати.

Слабка сторона: Важко змінювати – все вже зафіксовано.

Галузь: Будівництво, держпроєкти – потрібна стабільність і документація.

Agile

Сильна сторона: Гнучкість – легко змінювати курс.

Слабка сторона: Важко прогнозувати строки – все часто змінюється.

Галузь: Стартапи, IT – часто змінюються вимоги.

Scrum

Сильна сторона: Командна робота – всі в курсі процесу.

Слабка сторона: Мітинги – займають багато часу.

Галузь: Продуктові команди – потрібно швидко оновлювати продукт.

Kanban

Сильна сторона: Візуальний контроль – видно, на якому етапі задача.

Слабка сторона: Немає чітких строків – складно контролювати терміни.

Галузь: Підтримка, обслуговування – задачі приходять постійно.

3

На мою думку, Agile-маніфест з’явився тому, що розробникам набридло працювати за жорсткими, довгими і складними методами, як Waterfall. Там усе було дуже формально: спочатку план, потім довго писати код, а тестували вже наприкінці. І якщо щось ішло не так — усе приходилось переробляти з нуля. Це займало багато часу і нервів.

4

Agile створили, щоб зробити розробку гнучкішою. Щоб можна було швидше показати результат, отримати фідбек, і одразу ж щось покращити. Він мав вирішити проблему того, що клієнти отримували продукт тільки в самому кінці, а не бачили, як усе розвивається.

На мою думку, це спрацювало. Agile і досі використовують у багатьох компаніях, і він реально допомагає працювати швидше, адаптуватися до змін і краще взаємодіяти з командою та замовником.Agile дав свободу командам — більше не треба чекати місяцями, щоб щось протестувати чи змінити. Замість цього команди можуть працювати маленькими кроками й одразу бачити результат. Це робить процес більш живим і зрозумілим для всіх: і для розробників, і для замовників. Крім того, Agile зробив команду більш згуртованою — бо всі постійно спілкуються, обговорюють задачі, вирішують проблеми разом. Раніше такого не було — кожен робив свою частину і не бачив повної картини. А ще Agile дозволив створювати продукти, які реально потрібні людям, а не просто “за технічним завданням

5

Для розробки мого стартапу я обираю методологію Kanban. Це гнучкий підхід, який ідеально підходить для невеликої команди, яка швидко реагує на зміни. Оскільки ми запускаємо новий мобільний застосунок для обміну світлинами котиків, завдання можуть постійно змінюватися або додаватися в процесі — наприклад, нові ідеї від користувачів чи баги.

Kanban дозволяє візуально бачити, що вже зроблено, що в роботі і що ще в черзі. Це допомагає краще організувати роботу, не перевантажуючи команду. А ще ми можемо почати з мінімального функціоналу (MVP) і поступово додавати нові фічі без жорстких дедлайнів.

Саме тому Kanban — найзручніший для нас варіант на старті