## ДЗ після уроку 1

:росток:Beet Seed — відпрацюй навички на базовому рівні.

* Вибери предмет з твого оточення (наприклад, чашку, ноутбук, машину, футболку тощо) і спробуй його протестувати. Поясни, чому ти обрав/ла саме таку перевірку.
* Своїми словами поясни визначення валідації та верифікації.

**Настільний календар**

1. Перевіряємо, чи може календар самостійно стояти на столі. Чи зручно зроблена його опора.
2. Перевіряємо пружину, на яку кріпляться листки. Чи можуть листки вільно перевертатися, чи ніщо їм не заважає. Чи зручно користувачу перекидати листи? (ергономічна складова)
3. Ще одна частина ергономічної складової - який папір на дотик. Чи не занадто слизький, чи є можливість легко перегорнути усього один листок?
4. Чи безпечно використовувати календар? Чи немає у ньому замалих предметів, що може з’їсти дитина, чи немає занадто гострих кутів.
5. Перевіряємо наявність титульного листа. Чи все на ньому заявлено без помилок?
6. Перевіряємо останній лист. Чи вся необхідна інформація вказана.
7. Перераховується кількість листів. Перевіряємо, чи співпадає воно з очікуваннями користувача.
8. Перевіряємо, чи на потрібний рік надруковано календар.
9. Перевіряємо логіку порядку листів. Якщо це календар, то спочатку має бути січень, потім лютий и т.д.
10. Переходимо до тестування дизайну. Наскільки читабельный текст на листах? Чи зможуть люди зі слабким зором розібрати всю інформацію, що очікують побачити?

*Верифікація* - робота зі списком. У нас є список вимог до предмету тестування. І ми перевіряємо, чи всім цим вимогам він відповідає.

*Валідація* - перевірка, чи вирішує потреби (бізнес-цілі) клієнта предмет тестування. Наскільки зручно клієнту використовуючи предмет тестування вирішувати свої потреби.

Тут гарним прикладом можна навести додаток у смартфоні - калькулятор. Він створений для того, щоб користувач міг швидко порахувати. Тобто вже на першому екрані має бути поле для ввода даних та частіше використовувані функції (+ , -, \*, / ). Кнопки із цифрами мають бути великими і т.д.

:травы: Mighty Beet — детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

2. Склади порівняльну таблицю різних видів компаній. Вкажи плюси та мінуси кожної з них (з точки зору працівника).

3. Наведи приклади невдалої валідації або верифікації продукту, з якими довелося зіткнутися в житті.

| № | Вид компанії | Плюси | Мінуси |
| --- | --- | --- | --- |
| 1 | Product | 1. Маєш змогу краще вивчити продукт. Бо працюєш із одним і тим самим довгий період часу. (“я знаю все твои трещенки…” (с) ) 2. Ти стаєш профі у своїй сфері. Бо тестуючи одне й те саме доведеться копати все глибше і глибше, щоб уникнути парадоксу пестициду. | 1. Є вірогідність уповільнитись у розвитку як спеціалісту. (бо працюєш з одним і тим самим). 2. При зміні компанії ти можеш бути менш універсальним працівником, ніж інші претенденти. |
| 2 | Startup | 1. Якщо ідея кльова, то можна отримати надприбуток. 2. За рахунок невеликого штату, у таких компаніях можуть встановлюватись більш теплі відносини у колективі. 3. Такі компанії можуть швидко реагувати на зовнішні зміни. | 1. Є ризик, що продукт не вийде на самоокупність. 2. Як правило, тут багато не заробиш 🙂 3. Скоріш за все, бізнес-процеси у компанії будуть налагоджені не зовсім добре. 4. Є вірогідність, що тут будуть заощаджувати на персоналі, тож доведеться самостійно виконувати функції кількох відділів. (он же Жора, он же Гоша, он же Гриша) |
| 3 | Outsource | 1. Працюєш в одній команді. 2. Між тобою і замовником є менеджер, котрий буде “переводити” забаганки клієнта на зрозумілу для програмістів мову. 3. Різноманітна робота. 4. Ти за залежиш від успішності компанії замовника. Якщо продукт замовника провалиться, на твоїх прибутках це не позначиться. | 1. Може втрачатися оперативність прийняття рішень. 2. При роботі аутсорс команди вона може недостатньо розбиратися у всіх тонкощах продукту. 3. Замовник не завжди розуміє можливості команди. Із-за цього може пропонувати менш ефективні та більш ресурсозатратні рішення. |
| 4 | Outstaff | 1. Більш широке коло спілкування. 2. Більше потенційно корисних знайомств. | 1. Найбільшим мінусом, наскільки я розумію, тут є факт, що буде потреба ділитися коштами з компанієї у якій ти оформлений. 2. Персонал, що працює у штаті може зверхньо ставитися до тебе. Типу - вони тут боги, а ти - чорнорабочій. 3. На кожний новий проєкт у тебе буде новий колектив. |
| 5 | Academy | 1. Широке коло спілкування. 2. Спілкування з молоддю. | 1. Неперервний потік не завжди розумних питань. 2. Типові помилки студентів. |
| 6 | Recruitment Agency | 1. Більш широке коло спілкування. 2. Більше потенційно корисних знайомств. 3. Прокачка софт скилов. | 1. Токсичні претенденти. |

**Приклад невдалої верифікації:**

Ситуація - вибір ноутбука.

У переліку вимог до нового ноутбука бів пункт: тиха робота. Продавець стверджував, що техніка працює тихо. Тестування показало, що ця вимога не виконується.

:лиственное_дерево:Mighty Beet — різнобічно опануй тематику уроку.

| 1. Виконай завдання двох попередніх рівнів.  2. Поясни важливість 2-3 принципів тестування на власний вибір. Наведи приклади з власного досвіду. |
| --- |

Один із принципів тестування гласить, що знайти всі помилки неможливо.

Вважаю, що потрібно витримувати рівновагу і пам'ятати, що тести бувають зайвими. Ми витратимо лише зайвий час та кошти клієнта на тести, які якщо і виявляють проблеми, то вони будуть настільки несуттєві, що не окуплять затрат часу та коштів, що були витрачені на їх пошук.

*Якось я стільки часу витратила на вичитку тексту ексклюзивної новини, що інші ЗМІ встигли опублікувати мій ексклюзив першими.*

Ще один принцип нагадує нам, що раннє тестування - завжди гарна ідея.

Чим раніше буде виявлено баг, тим дешевше буде його виправити. А ми маємо раціонально використовувати як свій час, так і кошти клієнта. Такий підхід забезпечить компанії гарний розвиток та зробить її більш конкурентоспроможною.

Із нещодавнього: я витратила декілька годин на дизайн візуалу для соц мереж (хвилин 20 тільки розміщувала елементи декору, щоб вони гармонійно дивилися та підкреслювали основну ідею візуалу). І вже тільки під час публікації виявилось, що потрібно було робити візуал іншого розміру (а це значить, що всі мої рівняння та масштабування потрібно переробляти).

## ДЗ після уроку 2

:росток:Beet Seed — відпрацюй навички на базовому рівні.

Склади порівняльну таблицю найбільш поширених методологій:

| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall |  |  |  |

|  | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | 1. те, що немає потреби повертатися до минулих етапів, є плюсом для розробників з психологічної т.з. Бо один раз роботу зробив і забув про неї. 2. всі відразу чітко знають, що і як мають робити. Можна краще планувати час. 3. Вона більш проста у використанні. | 1. не можна повернутися на попередній етап, якщо баг було виявлено пізно, або якщо ідея поліпшення продукту прийшла запізно. 2. ми маємо чітко розуміти чого бажаємо, та як цього досягти. А це в часи швидких змін не завжди можливо | 1. корпоративні програми 2. Будівництво 3. освітні програми |
| 2 | V-модель | 1. тут більше контролю над проєктом. 2. нижча вартість бага. 3. Має чіткий порядок дій. | 1. Не підходить для проєктів, які можуть змінюватися із часом. 2. Не підходить для творчих проєктів. 3. Може бути складною для розуміння особливо для команд, що раніше із нею не працювали. | 1. Використовується для розробки медичного програмного забезпечення. 2. авіація 3. космос |
| 3 | Incremental Model | 1. клієнт має змогу оцінювати частини роботи вже під час створення продукту. 2. більш гнучка, можна вносити корективи під час роботи. 3. дає більш якісний результат. | 1. має бути згуртована команда, щоб всі чітко розуміли, що і коли робити. 2. важча в управлінні, бо можуть вноситися зміни на різних етапах. 3. знову ж таки із-за змін кінцевий продукт може бути менш стабільним. | 1. веб саіти 2. мобільни застосунки |
| 4 | Iterative Model | 1. висока якість кінцевого продукту 2. гнучкість при внесенні змін та відносна дешевизна внесення змін 3. швидкий відгук від клієнта | 1. якщо замовник не знає, чого бажає, то може бути важко досягати узгодження 2. затратна, бо багато часу йде на постійні обговорення 3. може бути нестабільний результат, якщо було занадто багато змін у процессі, особливо на його завершенні | 1. веб-портали  2. інтернет-магазини |
| 5 | Spiral Model | 1. допускає коригування у процесі роботи, тому більш гнучка 2. детальне обговорення кожного шматка роботи 3. високий контроль етапів | 1. високі витрати 2. складна у використанні | 1. доцільна при стартапах 2. медичні системи 3. авіація |
| 6 | Agile Model | 1. гнучка 2. швидкий фідбек від замовника 3. низька вартість багів | 1. немає точного розуміння що отримуємо у кінці 2. необхідно мати високі комунікативні навички 3. погано працює у не професійній команді. | 1 мобільні додатки  2. ігри  3. веб-застосунки |
| 7 | Scrum Model | 1. гнучка 2. швидко реагує на зміни 3. швидкий фідбек від замовника | 1. має бути професійна та згуртована команда | 1 мобільні додатки  2. ігри  3. веб-застосунки |

:травы:Mighty Beet — детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

2. Напиши розгорнуті відповіді (0,5 - 1 сторінки тексту) на такі два питання:

* На твою думку, чому з’явився Agile-маніфест?
* Які проблеми він мав вирішити і чи це вдалося?

Agile маніфест був створений у 2001 році групою досвідчених розробників програмного забезпечення, які працювали на реальних проектах та зіштовхнулися із проблемами традиційних методологій розробки ПЗ, таких як Waterfall. Вони дійшли висновку, що традиційні методології не сприяють гнучкості та швидкому реагуванню на зміни у вимогах та навколишньому середовищі.

Таким чином, Agile маніфест був створений з метою створення методології розробки, яка була б гнучкою, адаптивною та орієнтованою на швидке досягнення кінцевої мети. Він був розроблений на основі наступних принципів:

1. Люди важливіші за процеси та інструменти.
2. Те, аби продукт добре працював більш важливо, ніж те як чітко складено документацію.
3. Розробники (команди) мають бути готовими швидко реагувати на зміни.
4. Розробники (команди) мають постійно поліпшувати якість продукту та його цінність.

Саме щоб досягти цих завдань і було розроблено маніфест (деякі із них навіть стали його частинами).

Основні проблеми, які Agile маніфест має вирішувати, включають:

* Необхідність швидкого розгортання нових функцій задля досягнення конкурентних переваг;
* Вимога зниження витрат на розробку та покращення прогнозованості проектів;
* Необхідність швидкого реагування на зміни у вимогах та умовах ринку;
* Необхідність підвищення задоволеності клієнтів продуктом.

Можна сказати, що ріст конкуренції та пришвидшення темпу життя зумовили умови для створення такого маніфесту.

Виходячи із того, що нинішній темп життя не просто не зменшився, але й став ще швидшим, принципи маніфесту стають ще більш актуальніми.

Мені здається, тут можна провести паралель з принципом пестициду - якщо не змінювати підхід, то він перестане працювати.

:лиственное\_дерево: Mighty Beet — різнобічно опануй тематику уроку.

1. Виконай завдання двох попередніх рівнів.

2. Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків.

Яку методологію ти обереш для процесу розробки і чому? Відповідь текстово обґрунтуй.

В мене є асоціація, що я у цьому випадку буду замовником, що буде часто вносити корективи у ТЗ. Тому маємо обирати модель, яка це дозволить.

До того ж я буду бажати бачити шматки додатку на різних етапах, приймати участь у його розробці. Наша модель має дозволяти і це.

Тут доцільно буде використовувати Agile або Scrum.