**1.**

| **Назва** | **Плюси** | **Мінуси** | **Для якої галузі є доцільною** |
| --- | --- | --- | --- |
| ***Waterfall*** | Більше уваги приділяється раннім етапам розробки, тому ймовірність помилок зменшується в порівнянні з іншими моделями.  Є більш безпечними, тому що вони орієнтовані на план.  Легко управляти, оскільки кожна фаза має свої конкретні результати. | Тестування починається тільки після того, як реалізація завершена. Тому, якщо на якомусь етапі станеться помилка, то визначити це можна буде тільки коли прєкт готов і доведеться переробляти все з 0 і через це зростає вартість кінцевого продукту.  Не підходить для тривалих та безстрокових проєктів. | Там, де є можливість протестувати тільки вже готовий продукт.  Також, коли вимоги добре відомі й фіксовані, технології зрозумілі та команда досвідчена, яка “в темі”. |
| ***V-model*** | Легка в управлінні.  Дозволяє легко тестувати і аналізувати.  Тестування йде паралельно з розробкою, це економить час і гроші. | Не підходить для складних і великих проєктів.  Не підходить там, де вимоги постійно змінюються.  В разі будь-яких змін в процесі, необхідно оновити документацію. | Там, де потрібне проведення ретельного тестування продукту.  Для невеликих і середніх проектів з чітко визначеними вимогами. |
| ***Scrum*** | Гнучка система дозволяє вносити зміни в проєкт.  До кінця кожної ітерації готовий гнучкий програмний продукт, який можна показати замовнику і він сам “бачить” як йде процес.  Через безпосереднє спілкування фахівців, відносно невеликий обсяг письмової документації.  Зазвичай згуртована команда, в якій комфортно працювати.  Відомий наперед обсяг робіт.  Фокус на замовнику та його потребах - створення продукту, якого потребує клієнт. | Нестабільна вартість. Потрібна висококваліфікована команда, в якій немає місця новачкам.  Важко реалізувати в великій команді. | Підходить для програмних проєктів, які постійно змінюються або мають надзвичайно високі вимоги до масштабування. |
| ***Kanban*** | Простота метода.  Можна швидко вносити зміни в проєкт.  Немає суворих часових кордонів (відносно).  Легко визначити, в якому місці є ступор в процесі.  Зазвичай згуртована команда, в якій комфортно працювати.  Відомий наперед обсяг робіт.  Дошку канбан можна адаптувати під майже будь-яку сферу бізнесу.  Фокус на замовнику та його потребах - створення продукту, якого потребує клієнт. | Нестабільна вартість. Команда має працювати в +- одному темпі, щоб не гальмувати весь процес.  Важко реалізувати в великій команді. | Там, де можливі зміни в пріоритетах в проєкті і де важливі саме пріоритетність й строки виконання задач. |
| ***Iterative*** | Для початку реалізації не потрібна повна специфікація вимог.  Кожна створена версія хоч і не ідеальна ще, але працездатна.  Продукт будується і покращується крок за кроком, отже є можливість відслідковувати дефекти на ранніх стадіях.  Менше часу витрачається на документування, більше часу приділяється проектуванню. | Можуть виникнути дороговартісні проблеми, оскільки не всі вимоги зібрані заздалегідь для всього життєвого циклу. | Для дуже масштабних проєктів, де основне завдання заздалегідь визначене, але допустимо його деталі вдосконалювати, змінювати у процесі роботи. |

**2.2**

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

В умовах швидких змін у сучасному світі та прискорення розвитку технологій, старі жорсткі методи керування проєктами стали в деяких випадках менш ефективними. Тому автори вирішили зібрати всі актуальні проблеми, з якими не можуть дати ради жортскі методології, і, враховуючи досвід роботи з ними, на підставі цих проблем розробити щось нове, що б відповідало сучасним потребам в розробці, і стандартувати це, як систему цінностей.

Це дало змогу виставити пріоритети в роботі і звести бюрократичні папірці до мінімума, тим самим зосередившись, насамперед, над якістю продукта.

Робота в команді і ідентифікація себе, як члена команди, групи, а не просто чергового робітника, дає кращі результати в роботі. Тому що команда, як родина - підтримує і допомагає, коли ти потребуєш того. Бо у всіх один проєкт, і всі зацікавлені в його життєздатності і розвитку.

Тісна співпраця з замовником дозволяє краще почути й “відчути” замовника і його потреби, що в результаті дасть більш якісний продукт і можливість мінімізувати витрати на його розробку.

Через те, що я не маю практичних знань в цій сфері і можу спиратись тільки на теоретичні дані, мені важко сказати, чи всі проблеми маніфест вирішив, чи ні. Але, враховуючи, що гнучкі методології, в основі яких закладений Agile-manifest досить популярні, то, думаю, він був розроблен не дарма і справляється зі своєю задачею.

**3.2**

*Ти – засновниця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків.*

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

Якщо я планую тільки обмін світлинами з функцій застосунку, то можливо я обрала б V-model. Вона проста, підходить для невеликих проєктів. А через паралельне тестування можна зекономити кошти і час, якщо ретельно прописати специфікації застосунка. Через те, що функція в застосунку тільки одна - обмін світлинами, то логічно, що я не потребую проміжних етапів застосунку, мене влаштовує, що я отримаю вже готовий продукт.

Якщо я ще не впевнена, що в застосунку буде тільки обмін світлинами, а можливо я згодом вирішу, що застосунок може перерости, наприклад, в соцмережу власників котиків, то думаю, я б обрала Scrum. В мене ще немає кінцевого бачення мого продукту, то ж мені потрібна методологія відкрита до змін і до масштабування проєкту. В кінці кожного циклу я отримую працездатний продукт і можу вирішувати, чого мені не вистачає або що покращити чи змінити. Молода, дружня команда відповідає духу стартапу з творчим проєктом соцмережи.

А якщо мені вдасться зібрати в команді професіоналів любителів котиків, то успіх взагалі гарантований😆