**Домашня робота №2**

**Завдння №1**

| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | Простота | Помилка в одному з етапів відкидує проект на його початок | Для галузей в яких створення продукту не відбувається ітеровано, наприклад будівництво космічних кораблів |
| 2.. | Agile | Гнучкість  Зворотній зв’язок  Підходить великим проектам | Обмеженість в часі для певних завдань | Може бути використано в освіті де йде постійно процес змін та є необхідність в гнучкій системі |

а) Простота в Waterfall аргументується тим, що все чітко розписано, кожна стадія розробки починається з закінченням попередньої

Недолік полягає в тому, якщо на будь якій стадії скажем тестування знаходять помилку то необхідно повернутись на перший етап, а це вже тягне за соблю втрачені кошти та час.

Доцільно використовувати в будівництві космічних кораблів, адже Waterfall є жорсткою системою де кожен пункт перевіряється максимально сильно, а це є необхідністю в системі життєбезпеки персоналу.

Гнучкість Agile підходить для великого кола галузей де потрібна адаптація та реакція на зміни.

Зворотній зв’язок тому що замовник в постійному контакті з командою.

Підходить великим проектам адже під час затяжного періоду розробки світ не стоїть на місці і є можливість змінювати процес та фінальний продукт.

Негативна сторона це те, що певні завдання мають часові рамки, а це в свою чергу створює навантаження на команду.

Ця методолігія корисна для освіти адже зміни йдуть весь час і розробка будь якого ПЗ потребує оновлення згідно нових вимог, це програми для вивчення різних предметів куди постійно вносять корективи.

**Завдання №2**

На мою думку Agile маніфест з’явився тому, що необхідні були зміни, саме вони і були описані в 4 принципах. Недоліки вотерфол сприяли винекненню цієї моделі через свою жорсткість, не здатність пристосовуватись до змін. Самі відповіді на це питання на мою думку криються в написаних принципах. Люди та співпраця важливіші за процеси та інструменти - напевно тут зіграло роль, що в інших моделях була мала комункація між замовником та командою, що негативно впливало на кінцевий продукт. Замовник не міг втручатись в процес, а вже отримував кінцевий продукт, якщо проект займав рік то ми можемо уявити скільки змін відбулось за цей час, а отже продукт міг бути неактуальним і втратити свою цінність. Працюючий продукт важливіший за вичерпну документацію - цей принцип напевно був сформований для зменшення обігу документації та концентрації на роботі, адже якщо пройде тиждень і хтось захоче перевірити як прцює та чи інша функція, то може отримати відмову, це буде щось на зразок “ми багато працювали, у нас не вийшло все запустити, але гляньте на ось цю кімнату забиту папками, там ми розписали весь процес”

Співпраця із замовником важливіша за обговорення умов контракту - все звісно в цьому світі вирішують гроші, але за них ти не купиш любов, здоров’я та той продукт який дійно хочеш, тому що сидів і придумав, що хочеш понюхати кольори. Цей принцип виник напевно тому що при розробленні товару потрібна комунікація для кращого розуміння того, що саме хоче замовник. Найкращий контракт це супер, але біда коли взяли гроші і на половині проекту ви не можете його завершити, адже більше з замовником ви не спілкувались і не можете реалізувати його вимоги.

Готовність до змін важливіша за дотримання плану - У сучасному світі вимоги до програмного забезпечення часто змінюються дуже швидко. Традиційні моделі розробки, які передбачають жорстке планування на початку проекту, не завжди здатні ефективно реагувати на ці зміни. Agile ж, навпаки гнучкий та адаптивний, що дозволяє швидко реагувати команді на будь які виклики.

Agile вирішив багато проблем які були притаманні старим моделям, звісно він не панацея, але зазнав великого успіху.



**Завдання №3**

Будучи засновником старапу для застосунку з обміну світлинами котиків я б вибрав методологію відштовхуючись від багатьох факторів. Провівши аналіз визначив б скільки буде потенційних користувачів, свій бюджет, наявність конкурентів, так же дизайн застосунку, чи буде це суто для обміну картинками, чи будуть додаткові функції на зразок чатів і тд. В загальному оцінити весь об’єм роботи і визначити розмір проекту. Якщо проект буде не великим і з мінімальними функціями я б використав Канбан, адже цю методологію можна використовувти майже відразу з мінімізацією втрат в часі та не буде навантажувати команду. Якщо проект буде більший то можна використати Скрам, адже буде більша команда, та не потрібно буде під час створення продукту вдаватись в значні зміни, що дозволить розробити його якісно таким як задумано.