**Рівень 1**

**Завдання 1**

| **Методологія** | **Сильні сторони** | **Слабкі сторони** |
| --- | --- | --- |
| **Waterfall** | * зручна модель для використання у маленьких проектах з невеликою кількістю виконавців * є можливість чітко побачити терміни та витрати на проект * зрозуміла послідовність дій для замовника | * немає можливості повернутися на попередній етап у разі виявлення помилки * из-за послідовного виконання всіх робіт необхідно більше часу на завершення проєкту * не має можливості швидко отримати фідбек від QA та замовника |
| **V-модель** | * можливість тестування та контролю вимог на ранніх етапах * витрачається менше часу на виконання проєкту порівняно з Waterfall | * у порівнянні з гнучкими моделями, не має можливості перевірити виконані частини однієї фази до повного завершення цієї фази * не має можливості швидко вносити зміни |
| **Спіральна модель** | * є можливість проаналізувати кожний прототип та передбачити наступні ризики/проблемні місця проєкту * зручна для використання модель коли немає чітких вимог до проєкту або вимоги часто змінюються | * потребує великих фінансових витрат на розробку прототипів * емоційне вигорання від виконання одного й тож самого проєкту декілька разів))) |
| **Ітераційна модель** | * ефективний зворотній зв'язок з замовником * можливість виявити протиріччя у вимогах на початкових стадіях проєкту * можливість виконувати різни частини проєкту паралельно | * розмиті термін виконання робіт та витрати, бо кожна версія проєкту демонструється замовнику * необхідність постійної взаємодії з замовником |
| **Канбан** | * прозорість усіх процесів завдяки використовуванню дошки * можливість виконувати частини проєкту паралельно * сильна взаємодія учасників команди, менше менеджменту * більш гнучкий з використанням часу для усіх членів команди * визуально зрозумілий твій внесок у проєкт * можливість визначення пріоритетів | * не є найкращим варіантом для реалізації довгострокового проєкту( через його підхід може розтягнутися термін виконання тасків) * не підходить для работы великих команд, важко відстежувати прогрес проєкту * можливо виконувати тілько декілько задач паралельно |
| **Scrum** | * можливість виконувати частини проєкту паралельно * залучення скрам мастера до менеджменту проєкту * дає можливість уникати дедлайнів, або знизити їх кількість * модель підходить якщо не має чіткого розуміння кінцевого варіанту продукту * постійна робота за product-owner, є швидкий фідбек | * постійний тайм-менеджмент, мітинги( планерки “що будеш робити?” та оперативки “що зробив”) * в поточному спринті не має змоги додати нову задачу * не підходить для работы великих команд, важко відстежувати прогрес проєкту |