1. **Level**

| **Назва методології** | **Сильні сторони** | **Слабкі сторони** | **Для якої галузі є доцільною** |
| --- | --- | --- | --- |
| **Waterfall** | **-Чітка послідовність етапів**  **-Легко планувати терміни та бюджет**  **-Зручна звітність**  Це сильні сторони, бо роблять цю методологію постою та зрозумілою, підходящою для проектів зі стабільними вимогами | **- Складно вносити зміни**  **- Повільний зворотний зв’язок з користувачами**  **- Тестування вже готового**  Це слабкі сторони,бо є ризик виявлення помилок на пізніх етапах, це призведе до більших фінансових витрат та часу, є ризик втрати актуальності | **-Розробка банківської системи з фіксованими вимогами -Створення програмного забезпечення для медичного обладнання -Розробка авіаційної навігаційної системи -Впровадження податкової звітності для державного органу -Будівництво**  Підходить для державних проектів та проектів з чітко визначеними вимогами та мінімумом змін, де потрібна повна документація |
| **Scrum** | **-Гнучкість до змін -Регулярний зворотний зв'язок -Прозорість -Командна відповідальність**  **-Постійне вдосконалення -Часті релізи**  Це сильні сторони бо можна легко змінювати напрям розвитку продукту, усі бачать, що робиться і що заважає,швидко виявляються та вирішуються проблеми, продукт швидко потрапляє до користувачів. | **- Висока вимога до зрілості команди**  **- Потрібен досвідчений Scrum Master**  **- Забагато нарад**  Це слабкі сторони бо без правильно спланованого робочого навантаження не працює, а наради відволікають від основної роботи | **-Розробка мобільного застосунку**  **-Digital-маркетинг**  **-Запуск сервісу управління персоналом**  **- Онлайн-платформа для курсів**  Підходить для проектів де клієнт постійно додає нові функції, різні частини функціоналу розробляються поетапно, де присутні постійне доповнення контенту та адаптація до користувачів |
| **Kanban** | **-Висока гнучкість**  **- Візуалізація всього процесу на дошці**  **- Постійний потік задач без жорстких спринтів**  **- Немає обов'язкових нарад**  Це сильні сторони бо підвищують ефективності роботи та взаємодію в команді, витрачається менше часу на обговорення та наради | **- Відсутність фіксованих термінів**  **- Можлива втрата фокусу**  **- Менш структурована взаємодія**  Це слабкі сторони бо можуть призвести до затягування задач та втрати задач з високим пріоритетом | **- Техпідтримка**  **- Підтримка та розвиток існуючого програмного продукту**  **- DevOps**  **- Контент-команди**  Ідеально підходить для роботи з постійним потоком задач, непріоритетних змін, підтримки, обслуговування |

1. **Level**

На мою думку Agile-маніфест з’явився бо виникла потреба зробити важкий процес розробки програмного забезпечення більш легким, побудувати працююче середовище де люди найважливіший актив, а готовність до змін є приоритетом. Він має вирішити проблему великої кількості документації, зробити більш тісною співпрацю з замовником на всіх етапах проекту, щоб клієнт бачив результат на кожному етапі, а не наприкінці проекту. Мотивувати людей працювати над проектом забезпечивши їх необхідним середовищем і підтримкою. Гадаю, що маніфест вирішує ці проблеми, більшість ІТ-команд працюють у коротких спринтах, стало менше бюрократії та більше гнучкості та готовності змін.

Дуже сподобалась ця формуліровка :)

**Agile — це не чекати півроку, а робити щось корисне кожні 1–2 тижні, показувати результат і постійно вдосконалювати.**

1. **Level**

Як засновниця стартапу, яка планує випустити мобільний застосунок для обміну світлинами котиків, я оберу Scrum методологію для розробки, тому що застосунок буде створюватись поетапно з додаванням нових функцій, буде постійно доповнюватись та адаптуватися під користувача.