**Beet Seed**

Склади порівняльну таблицю найбільш поширених методологій:

| **Тип компанії** | **Плюси** | **Мінуси** |
| --- | --- | --- |
| Waterfall | * Гарно підходить під невеликі проекти; * Легкий у підтримці (оскільки фази чіткі та йдуть по черзі); * Гарантує високу якість продукту (оскільки має чіткі та зрозумілі критерії); * Результати добре задокументовані; | * Важко працювати зі змінами у вимогах; * Демонстрація продукту відбувається в кінці проекту(немає проміжних прототипів, відповідно існує ризик, що клієнту не сподобається кінцевий продукт); * Складний менеджмент ризиків (оскільки тестування проводиться майже в кінці проекту). |
| Scrum | * Гнучкість (адаптувати проект до змін набагато легше, ніж умовиний waterfall); * Регулярний зворотній зв’язок (відгук замовника надходить після завершення спринту); * Комунікація команди (щоденні стендапи, на якому проговорють хто, що зробив/ планує зробити, або які виникли перешкоди) | * Необхідність досвідченої команди (необхідно, щоб команда вже мала достатній рівень досвіду, щоб самостійно ухвалювати рішення); * Обмежене передбачуване планування ( оскільки працює на основі коротких спринтів, що може ускладнити більш детальне і довгострокове планування; |
| Kanban | * Орієнтація на безперервну постановку завдань (команда може працювати над завданнями в міру їх надходження, а не чекати початку нового спринту); * Візуалізація роботи (використання дошки Канбан дає змогу команді бачити, які завдання перебувають у процесі виконання, які завдання очікують на виконання, а які завдання завершені); | * Відсутність структури (Команді може бути важко працювати ефективно без певного набору практик і ролей, що може призвести до плутанини та неефективності.) * Відсутність часових рамок (Канбан не встановлює конкретної тривалості часу для роботи, і це може ускладнити ефективне планування та розставлення пріоритетів); * не призначений для довгострокового планування |

\* Пояснення наводив одразу в дужках

**Beet Sprout**

На мою думку, виходячи з опрацьованого матеріалу, Agile-маніфест з’явився тому що індустрії розробки ПЗ потрібні були зміни. Раніше домінували традиційні методології розробки ПЗ, які називають "негручкими". В основному це Waterfall-моделі. Це про послідовність процесів, про лінійність, де кожна фаза спирається на попередню. Плюс Waterfall в тому що він, наскільки я зрозумів, забезпечує структуровану основу, але будь-яке відхилення від початкового плану часто призводило до значних витрат і затримок, тому що виправити помилку спочатку - дешевше і швидше по часу, ніж виправляти і шукати помилки на фінальному етапі проєкту. Через тривалі цикли розробки проходило багато часу від моменту початку розробки та до моменту потрапляння продукту на ринок. Ця затримка могла призвести до того, що продукти ставали застарілими ще до того, як потрапляли, наприклад до App Store чи Play Market, якщо мова про додатки наприклад. І ще - комунікація. Структура традиційних методологій перешкоджала ефективній комунікації між членами команди та зацікавленими сторонами. Це могло призвести до непорозумінь. Я думаю що Waterfall може бути застосований у коротких(по часу) і маленьких за обʼємом проєктах.

Тим не менш, людство і технології розвиваються і виникають потреби у адаптивних підходах для вирішення питань, які не вирішували традиційні. Маніфест мав на меті забезпечити більш адаптивний та гнучкий підхід до розробки програмного забезпечення. Для мене досить логічно що спочатку була створена філософія Agile, яка була описана через цінності, які описуються у 12 керівних принципів.

Agile мав вирішити проблеми із комунікацією, із тим щоб продукти потрапляли на ринок вчасно і можна було отримувати фідбек від користувачів і вдосконалювати його далі відповідно до потреб ринку і не за всі гроші світу. Думаю Agile довів свою ефективність. у швидкому створенні цінності та реагуванні на мінливі вимоги ринку.