**Level 1**

| **Methodology** | **Strong sides** | **Weak sides** | **Best usage** |
| --- | --- | --- | --- |
| Waterfall | * стабільність - незмінність вимог дозволяє чітко поставити таймлайни і оцінити необхідні ресурси. * послідовність проекту - дозволяє залучати роботу відповідальної команди у необхідний проміжок часу, що дозволяє економити ресурси * готовий продукт на виході - оскільки всі вимоги обумовлені наперед, то при закінченні проекту ми отримуємо повністю завершений продукт у необхідні терміни. | * старт наступного кроку залежить від завершення попереднього - може призвести до ланцюгової реакції у вигляді затримок, помилок. * чим більше “кроків” виконано - тим дорожчою буде вартість помилки - знаходження будь якого дефекту на пізніших етапах зумовлює нас повернутися до кроку де цей дефект виник, відповідно, всі кроки, після “дефектного”, повинні виконуватися заново. По-перше, знаходження дефекту на пізніх етапах стає важчим, по-друге, виправлення помилок - це час та ресурси - трудові та фінансові. * готовий продукт на виході - замовник отримує готовий продукт тільки в кінці проекту, що унеможливлює будь яке його використання на протязі всієї розробки. | * Медицинська, космічна галузь - довготривалі, складні і точні за вимогами проекти. * Короткострокові, не складні проекти * Проекти де є чіткі ліміти по часу і фінансуванню |
| V-model | * сильні сторони з Waterfall моделі * контроль та залучення практик тестування під час кожного етапу розробки - ми можемо проводити аналіз і давати оцінку того чи іншого сегменту з боку майбутнього тестування і вказувати на можливі недоліки. | * слабкі сторони з Waterfall моделі * можлива некваліфікована оцінка або не компетентність, що призведе до зайвих кроків або навпаки - не знаходження критичних недоліків | * Проекти де є чіткі ліміти по часу і фінансуванню * Проекти де важливість контролю та тестування на всіх етапах є важливим (державні структури) |

| **Methodology** | **Strong sides** | **Weak sides** | **Best usage** |
| --- | --- | --- | --- |
| Spiral model | * мінімізація ризиків - дана модель має на меті детально спланувати розробку, зробити аналіз всіх ризиків щоб задовольнити споживача * ітерація проектів - дозволяє бути більш гнучким до можливих змін замовника та реалізовувати їх на подальших “витках”. | * висока вартість моделі - оцінка ризиків та надання відповідних результатів може нести великі часові та фінансові втрати. * неможливість визначення часових рамок випуску продукту | * великі та high risk проекти - банківська та фінансові структури, сфера дослідження наук. |
| Scrum | * flexibility - швидке реагування на будь яку зміну вимог замовника за мінімальних втрат. * тайм менеджмент (Sprint) - чіткі межі коли і як проект має бути завершений * контроль та підтримка - наявність ролі scrum master * оцінка та improvement - наявність planning (планування тасків на Sprint) and retrospective meetings (всі негативні моменти оцінюються і покращуються). | * плинність вимог - вимоги можуть коригуватися, доповнюватися і змінюватись на любому етапі розробки, що може негативно впливати на часові рамки, продуктивність. * важкість ведення документації - через можливі постійні зміни ведення документації є вкрай важким. | * Стартап проекти * Застосунки, де постійно відбувається зміна потреб користувачів |
| Kanban | * Канбан дошка - перелік, пріоритизація завдань та їх статус * самоорганізація команди - завдання розподіляється командою у найбільш прийнятний для них спосіб. * flexibility | * неможливий контроль за роботою команди (неможливість оцінки трудовитрат) * відсутність часових рамок | * Стартап проекти * Як Improvement tool для стабільних проектів |

**Level 2**

**На твою думку, чому з’явився Agile-маніфест?**

Щоб дати відповідь на дане питання треба поглянути на 4 основні тези: **Люди та співпраця, Працюючий продукт, Співпраця із замовником, Готовність до змін.** На мою думку, це результат технологічного ривка людства, де основні цінності перейшли на сторону розвитку, співпраці, націленість на результат. Можна сказати це наступний виток еволюції, який можна порівняти з винаходом вогню або інструментів. Маніфест склали люди, які розуміють важливість пристосування до активного середовища, люди що розуміють що таке вихід з комфортної зони, а не закладених застарілих правил, які вже мають парадокс пестициду :) В маніфесті двічі вказана “співпраця” - так, дійсно, в наш час тільки ефективна співпраця приносить вагомий результат. Співпраця є всюди навколо нас, в кожній галузі, без неї суспільство, як таке, не існувало б. Я думаю метою з'явлення Agile маніфесту було тверезе бачення реальних речей які максимально приносять користь, економлять час і підвищують продуктивність і висвітлення їх суспільству, а не буржуйських принципів, які зазвичай заважають процесу і витрачають дорогоцінний час.

**Які проблеми він мав вирішити і чи це вдалося?**

Якщо брати до уваги чисто мою суб'єктивну думку, то маніфест має вирішувати такі проблеми:

* Співпраця має бути прозорою, ефективною та націлена на результат, сторони мають бути зацікавлені та мотивовані у позитивному результаті, що призведе до задоволення обох.
* Якість понад усе - все що створює людина має бути працюючим і відповідним до очікувань, жодна теорія не покаже такий результат як реальна практика, жодна документація не підтвердить якість і працездатність продукту як його реальна презентація
* Зміни - це не завжди погано, це новий виклик що робить нас кращими і досвідченими.

Ми живемо у світі комерції і бізнесу, де найцінніший ресурс це час. На мою думку, маніфест не може вирішити більшість проблем, адже все дуже індивідуально. Але, якщо дивитися через призму методологій у ньому, то далеко не всі, на жаль, компанії дотримуються цінностей маніфесту. А ті, що дотримуються, не можуть 100% гарантувати їх виконання. Адже це, як у тестуванні, де ми не можемо гарантувати що все правильно або по правилах, і постійно пристосовуємось до нових викликів.

**Level 3**

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

Я би розглядав 2 можливих методології:

1. **Waterfall** - проект і вимоги є стабільними - нам потрібні котики якими обмінюються. Ресурси у цьому випадку можуть витрачатися покроково на кожному етапі розробки, можливо визначити необхідний час і фінанси. Проект не є складним і не носить різнобічний функціонал, тобто критичного впливу при виявленні проблем на певній стадії не буде відбуватися. Головна ціль - отримати працюючий додаток у короткі терміни.
2. **Iteration (Kanban)** - позаяк у нас ще немає конкретного замовника і організації для даного проекту, то Scrum не буде розглядатися. Натомість для стартапу я б використав Канбан методологію, де команда, кожен її учасник, сфокусований на певний відрізок роботи, без зайвого контролю і обмеженням у часі (звичайно, дані пункти можуть певним чином з'явитися, якщо до проекту приходить інвестор). Цю методологію я б використовував якщо є ймовірність зміни вимог або їх доповнення під час розробки (люди ще ж хочуть світлини собачок). Потрібно проаналізувати ринок і розуміти які шляхи можуть бути обрані задля підвищення шансу виходу продукту на ринок і чи всі умови були взяті до уваги.