| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | Вимоги фіксовані;  Висока якість та надійність;  Зазвичай мало або відсутні переробки;  Чітка зрозуміла проста модель - крок за кроком;  Такий вид максимально зрозумілий, чіткий і прозорий для Замовника | Кожен крок повинен бути повністю досконалим для наступного - при помилках (які знаходяться АЖ у кінці) несуться великі збитки і треба починати все спочатку;  Вносити зміни у попередні етапи неможливо;  Робити декілька етапів паралельно не можна;  не гнучка метод-я;  Замовник може побачити результат тільки в кінці | підходить при розробці складних і великих проектів і систем із чіткою певною функціональністю;  Пз для військових, для медицини |
| 2 | V-модель | Строга поетапність із проміжним тестуванням (дефекти виявляються з початкової фази);  Планування тестування і верифікація системи виробляються на ранніх етапах;  Швидше моделі №1 (попередньої);  Виявляє відхилення в проекті і ризики на ранніх стадіях;  Універсальне документування полегшує читаність;  Зменшення загальної вартості проекту - ресурси можуть бути заздалегідь прораховані та проконтрольовані | Модель не передбачає роботу з паралельними подіями;  Не передбачено внесення вимоги динамічних змін на різних етапах жц;  У модель не входять дії, спрямовані на аналіз ризиків | У проектах, в яких існують часові та фінансові обмеження |
| 3 | Ітеративна | Раннє створення працюючого ПЗ;  Гнучкість – готовність до зміни вимог на будь-якому етапі розробки;  Кожна ітерація – маленький етап, для якого тестування та аналіз ризиків забезпечити простіше;  Реальна оцінка поточного стану проекту і, як наслідок, велика впевненість замовників і безпосередніх учасників в його успішному завершенні;  можливість організувати зворотний зв'язок з майбутніми кінцевими користувачами з метою створення системи;  більш рівномірне завантаження учасників проекту; | Система часто погано структурована, проект «не прозорий»;  Після уточнення вимог відкидається частина раніше виконаної роботи;  Потребує багато часу, тому можна не вкластися у визначений бюджет і дедлайн;  Кожна фаза – самостійна, окремі ітерації не накладаються одна на одну;  не всі вимоги відомі до початку проектування; | Підходить для малих та середніх  проектів, для стратапів невеликих;  Коли вимоги до проекту можуть мінятися в процесі розробки |
| 4 | Спіральна модель | Додаткові функціональні можливості або зміни можуть бути зроблені пізніше;  Оцінка вартості стає простою, оскільки побудова прототипу виконується дрібними фрагментами;  Розвиток відбувається швидко  Оцінка ризику є правильною;  Контроль на всіх фазах розвитку;  Програмне забезпечення виробляється рано;  Має місце для зворотного зв'язку із клієнтами, і зміни впроваджуються швидше. | Ризик невідповідності графіку чи бюджету;  Документація більше, оскільки вона має проміжні фази;  Аналіз ризиків є важливою фазою, тому потрібні досвідчені люди;  Невигідний для менших проектів - дорого. | У програмній інженерії використовується, коли проект великий;  Коли застосовується створення прототипу;  Для проектів середнього та високого ризику; |
| 5 | Scrum | орієнтований на клієнта, адаптивний;  Отримання швидких результатів, часті постачання змін продукту;  досить простий у вивченні, дозволяє економити час, за рахунок виключення некритичних активностей;  можливо отримати потенційно робочий продукт наприкінці кожного Sprint'а;  Створюється перелік пріоритетів;  Наприкінці спринту робота має бути готова для клієнта | Недостатньо опрацьована документація;  Потрібна активна залученість клієнта до процесу розробки  якщо член команди виходить із середини спринту, може бути важко закінчити вчасно; не всі полюбляють щоденні зустрічі | у маркетингових агентствах та дизайнерських студіях, у телеком-компаніях, у фармакології, у роздрібних мережах -  привабливо для малих компаній та стартапів, тому що позбавляє необхідності від найму або навчання спеціалізованого персоналу керівників |
| 6 | Kanban | Візуалізація процесів;  Гнучкість планування;  Контроль термінів виконання;  Наочність просування роботи;  Угруповання задач;  вага до незавершених завдань;  Постійне вдосконалення | Не підходить для довгострокового планування;  Не підходить для великих команд;  може не підходити в дуже швидко мінливому середовищі;  відсутність чітких термінів може знизити її корисність;  не зосереджена на стратегії чи цілях | У бережливому виробництві - медицина, промислове виробництво, виробництво електроніки.  Канбан — це процес поступового поліпшення будь-якої діяльності, будь то розробка Пз, надання ІТ-послуг, підбір персоналу, маркетинг, продажі, закупівлі. |
| 7 | ХР | Замовник отримує саме той продукт, який йому потрібен;  Команда швидко вносить зміни в код і додає нову функціональність за рахунок простого дизайну коду, частого планування і релізів;  Код завжди працює за рахунок постійного тестування і безперервної інтеграції;  Команда легко підтримує код т. я. він написаний за єдиним стандартом і постійно рефакториться;  Знижуються ризики, пов’язані з розробкою, так як відповідальність за проект розподіляється рівномірно та вихід/прихід члена команди не зруйнує процес (парне програмування);  Витрати на розробку нижче, так як команда орієнтована на код, а не на документацію і збори | Успіх проекту залежить від включеності замовника;  успіх XP сильно залежить від рівня програмістів;  регулярні зустрічі з програмістами дорого обходяться замовникам;  із-за нестачі структури та документації не підходить для великих проектів | застосовується **тільки в області розробки програмного забезпечення.** Воно не може бути використане в іншому бізнесі або повсякденному житті, як scrum чи kanban |

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

Agile-маніфест - це документ, що описує основні принципи, на яких базується гнучка розробка.

Agile - це як спосіб мислення, гнучкий спосіб мислення.

Коли в компанію приходять різні люди із своїми правилами, цілями та поглядами на те як треба працювати, а також досвідом із попередніх компаній, то компанія створює Свої спільні правила. Ці правила про те як поводити себе в компанії та як комунікувати один з одним для того, щоб покращити та полегшити співпрацю і мати найкращій результат (продукт).

Із часом міняються не тільки основні правила на роботі, а і цінності у суспільстві. Раніше головним було - зробити Продукт і не важливо якою ціною людського ресурсу. Коли людина емоційно виснажується - його робота стає такою ж виснаженою, тобто зробленою не на відмінно. Щоб досягти максимально класної якості продукту - потрібно мати не тільки професіоналів своєї справи, а ще й мотивованих працівників.

Цей маніфест написаний виходячи із надання цінності людським ресурсам, а також щоб робітники швидше адаптувалися до змін у світі технологій. Замовник 5 років тому і Замовник сьогодні - це різні моделі поведінки та запити до команди. Час іде, побажання змінюються, методи роботи - прискорюються.

Зараз найкращий робітник - це професіонал своєї справи, який слідкує за останніми тенденціями у технологічному напрямку, який комунікує із командою та розуміє запит замовника, який може самоорганізувати свій робочий час та випускає якомога частіше робочий продукт. Саме ці принципи і прописані у маніфесті, завдяки яким можна *спростити робочий процес та підвищити ефективність роботи*.

Якщо не “перегинати палку” із окремими пунктами маніфесту, а дотримуватись всіх разом - відбудеться оптимальне спонукання робітників до відтворення гарної, відповідної до вимог замовника якості Продукту.

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

Яку методологію ти обереш для процесу розробки і чому? Відповідь текстово обґрунтуй.

Я буду використовувати на початку свого стартапу **Ітеративну модель**, тому що:

1. Перший робочий варіант Пз вийде швидко;
2. Його можна оновлювати ( додавати нові функції) швидко і фінансово економічно;
3. Ця модель працює над невеликими стартапами й поки не знає, яким має бути кінцевий продукт;
4. Треба створити продукт, що задовольнятиме конкретні потреби - обмін світлинами котиків;
5. Постійний розвиток свого проекту