

1. Склади порівняльну таблицю найбільш поширених методологій:

Назва методології	Сильні сторони	Слабкі сторони	Для якої галузі є доцільною
1 Waterfall	<ul style="list-style-type: none"> Зрозуміла схема робочого процесу; Наступний етап розпочинається лише після успішного завершення попереднього; є чітке розуміння який продукт повинен бути на виході; бюджет витрат і ресурсна база завчасно визначені; можливість відхилення від запроєктованих часових термінів виконання; 	<ul style="list-style-type: none"> Не можливість оперативно внести зміни/покращення у продукт; Тестується уже готовий цілісний продукт; якщо розробка вимагає збільшення ресурсів, стадію тестування можуть знехтувати; низький рівень комунікації з замовником; 	<ul style="list-style-type: none"> Більше підходить до проектної роботи. Складні, суспільно важливі проекти, довготривалі проекти (військова промисловість; космічна галузь; державні проекти); Також, може бути застосована для короткотермінових проектів, у яких є чітке розуміння мінімального набору функціональності кінцевого продукту; <p>// оформлення вимог</p> <ul style="list-style-type: none"> вимоги оформлені часто у вигляді деталізованого техн завдання, як додаток до договору між замовником і виконавцем <p>In Waterfall, it is referred to as 'Requirement/Specification Document'</p>
2.SCRUM	<ul style="list-style-type: none"> Гнучко приймаються рішення про зміну/покращення продукту; Постійна комунікація з замовником для врахування його потреб; команди, мотивовані на якість і швидкість; висока взаємодія між командою; короткі і зрозумілі терміни розробки; продукт повинен бути створений до конкретного терміну мінімізація ризиків завдяки гнучкій системі внесення змін 	<ul style="list-style-type: none"> Первинний проект і фінальний проект можуть відрізнятися (ризик нескінченних змін продукту); практично неможливо точно підрахувати підсумкову вартість проекту; підвищені вимоги до кваліфікації і досвіду команди 	<ul style="list-style-type: none"> сфера, схильна до постійних змін (ІТ-компанії, стартапи, проекти в інноваційних сферах) Більше підходить до процесної роботи (розробка ПЗ, впровадження, супровід) <p>// оформлення вимог</p> <ul style="list-style-type: none"> вимоги надаються для невеликих функціональних можливостей або особливостей, оскільки продукт готується крок за кроком. <p>in Agile or SCRUM it is referred to as 'Epic', 'User Story'.</p>

2. Напиши розгорнуті відповіді (0,5 - 1 сторінки тексту) на такі два питання:

- На твою думку, чому з'явився Agile-маніфест?
- Які проблеми він мав вирішити і чи це вдалося?

Agile-маніфест розробила група програмістів у 2001 році. На цей момент уже було сформоване розуміння, що традиційні підходи до розробки не повністю відповідають викликам часу (стрімкий розвиток економік). Традиційні методи розробки передбачали жорсткі та деталізовані вимоги на початку проекту; відгуки від клієнтів отримувались лише на пізніших етапах проекту, що призводило до непорозумінь між розробниками та замовниками; неможливість враховувати зміни, які можуть виникати у процесі розробки. Це призводило до затримок та великих інтервалів між початком роботи та отриманням реальних результатів. Під тиском виконання графіків та бюджетів, розробники іноді могли уникати докладного тестування та вдосконалення програмного продукту, що впливало на його якість.

Проте, у багатьох сферах бізнесу вимоги змінюються швидко внаслідок зміни ринкових умов, конкуренції та вимог клієнтів. Тому, Маніфестом сформульований альтернативний підхід до розробки, який орієнтується на більшу адаптивність, гнучкість та спрямований на задоволення потреб клієнта. Маніфестом вдалося вирішити частину проблем традиційних моделей, а саме:

підвищення ефективності роботи через короткотривалі стринти, оперативну комунікацію щодо вимог з замовниками, ранню взаємодію з тестувальниками. Це дозволяє зменшити ризики якості продукту через постійну адаптацію та вдосконалення.

Разом з тим, впровадження Маніфесту має свої обмеження, передусім:

складно застосовувати у великих організаціях зі складною структурою через необхідність управління багатьма командами, координації та інтеграції різних частин;

при великій кількості стейкхолдерів важко узгоджувати та імплементувати зміни до вимог.

3. 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.

А). Продукт розробляється для регіональної спільноти любителів котиків. Команда з багаторічним досвідом розробки мобільних застосунків. Є чітке розуміння функціональності готового продукту як зі сторони замовників, так і усередині команди.

Швидше за все Waterfall, бо: точно знаємо, що хочемо отримати на виході; розробляємо для обмеженої чисельності користувачів; команда досвідчених розробників саме такого типу застосунків. Відповідно, розраховуємо, що все піде за планом і буде виконано у короткі терміни.

Б). Продукт розробляється для регіональної спільноти любителів котиків. Команда молодих спеціалістів-ентузіастів. Є розуміння функціональності готового продукту, але потік ідей не припиняється.

Швидше за все Agile, бо: не впевнені, що саме хочемо отримати на виході, тісно комунікуємо з замовниками (регіональна кото-спільнота) і усередині команди; команда потребує досвіду і вищої кваліфікації. Відповідно, розраховуємо, що на розробку і тестування буде потрібний додатковий час, оскільки вимоги до продукту часто оновлюються. Разом з тим, завдяки гнучкій системі внесення змін, мінімізуємо ризики провалу.

В). Продукт розробляється для світового ринку. Команда з багаторічним досвідом розробки мобільних застосунків. Є чітке розуміння функціональності готового продукту. Разом з тим, відслідковуються ринкові тренди у подібному сегменті застосунків.

Швидше за все Agile, бо: хочемо випустити продукт актуальний для ринку, тому працюємо мотивовано і якісно з чітко поставленими цілями для їх досягнення у короткі терміни; гнучко вносимо зміни, орієнтуючись на ринок/конкурентів, у тому числі.