1. **Порівняльна таблиця найбільш поширених методологій**

| **Методологія** | **Сильні сторони** | **Слабкі сторони** | **Для якої галузі є доцільною** |
| --- | --- | --- | --- |
| Waterfall | Через зрозумілу структуру розробки проєкту знижується поріг входження для команд.  Дозволяє легко відстежувати ризики, графік виконання, витрачені ресурси завдяки поетапному виконанню і великій кількості документації.  Завдання, які висуваються перед командою, є від початку зрозумілими і не змінюються впродовж виконання проєкту, тому не вимагає постійної взаємодії команди.  Термін виходу нового продукту, його вартість можуть бути розраховані вже на початку процесу його створення.  Не вимагає постійної участі клієнта, що спрощує роботу. | Неможливо вносити зміни в продукт під час розробки. Існує ризик випуску невдалого продукту через те, що фаза тестування проводиться вже на готовому продукті.  А також відсуність залучення клієнта до етапів розробки і тестування може призвести до невідповідності продукту вимогам, що можуть змінитись.  Команди є вузькоспеціалізовані, що ускладнює організацію спільної роботи.  Неможливо змінити проєкт в плані оптимізації витрат, зміни функціоналу чи концепції до випуску готового продукту.  Багато витрат на документування всіх етапів життєвого циклу продукту. | Підходить для сфер, для яких створюваний продукт має зрозумілі чіткі вимоги, є багато ресурсів та часу на реалізацію, вимагається строга документація за результатами всіх етапів:   * авіабудівництво * будівництво * медична галузь * фінансовий сектор * військова галузь |
| Agile | Більше орієнтований на клієнта, тому що забезпечує більшу залученість його до процесу створення продукту.  Забезпечує вищу якість продукту, завдяки постійній взаємодії з замовником.  Команда, яка працює над проектом більш вмотивована та самоорганізована, тому часто дає кращі результати.  Кожна ітерація створює робочу версію продукту, що важливо в умовах важливості часу виведення його на ринок.  Ризики мінімізовані завдяки гнучкій системі внесення змін. | Гнучкість розробки продукту може призвести до того, що він ніколи не вийде на ринок.  Вимагає залучення експерта для прийняття важливих рішень.  Підвищені вимоги до кваліфікації команди, тому що всі, хто включені в процес, окрім своїх прямих обов'язків повинні вміти аналізувати можливі способи покращення ефективності роботи.  Складнощі початкового підрахунку витрат на проект через часті внесення змін для постійного вдосконалення продукту.  Не підходить для крупних проєктів, де важливим є чітко задокументовані процеси. | Підходить для розробки ПЗ ІТ-компаній, інноваційних проектів, де важливо швидко пристосовуватись до змін ринку задля випуску нових конкурентних продуктів і здобутті нових клієнтів. |

1. Розробка програмного забезпечення до появи Agile була передбачуваною і зрозумілою: була визначена чітка послідовність робочих процесів. Однак управління проєктами було негнучким, відхилення від плану загрожувало провалом робочого процесу в цілому. Історія показує, що більшого успіху досягають ті компанії, які комбінують різні інструменти і методології, в залежності від специфіки проєкту і контексту, в якому він реалізується. Представникам компаній-розробників потрібно було шукати альтернативу підходу до роботи, яка керувалась тільки планами і документацією, задля досягнення більших успіхів у своїй праці та створення конкурентоспроможних продуктів. Думаю, розробники ПЗ зіштовхнулись із проблемою відтоку клієнтів, які були незадоволені кінцевим продуктом, оскільки не мали змоги контролювати його в процесі розробки, а на етапі виходу продукту вже пізно було щось перероблювати.

Плюс з часом з'явився великий попит на розробку програм, які користувались масовим попитом для вирішення простих користувацьких завдань, та для автоматизації рутинних офісних операцій. У розробці таких програм найважливішим були зручність у використанні та корисність для користувача. А це складно було зрозуміти на стартовому етапі до появи хоча б першої версії програми.

Так і з'явився підхід Agile, який вирішив проблеми шляхом поетапного створення і покращення продукту через постійну взаємодію з замовником. Застосування цієї методики управління проєктами з розробки ПЗ, в більшості випадків, посприяло пришвидшенню створення готових продуктів та підвищило продуктивність роботи команди (завдяки тісній і продуктивній взаємодії).

1. Для процесу розробки мобільного застосунку для обміну світлинами котиків в умовах стартапу я оберу методологію управління проєктом Scrum. Ми взялися за створення нового додатку і не маємо впевненості в тому, яким він буде в результаті, тому візьмемо за основу розробку у формі невеликих циклів (спринтів) і швидких експериментів (стартап - це ж про експеримент).

Оскільки це стартап, тут має місце обмеженість ресурсів, а, отже нам потрібно розумне їх використання. Збір та аналіз зворотного зв'язку від кінцевого користувача після кожного спринта, швидка адаптація до змін ринку дозволить знизити ризики великих витрат.

Оскільки команда в стартапі невелика, об'єднує людей задля високої мети, це спрощує взаємодію між працівниками, забезпечує виконання принципів Scrum: підтримки один одного і відданості своїм цілям, залучення в процес розробки усіх членів команди. Таким чином в процесі такої співпраці учасники команди матимуть змогу за короткий проміжок часу створити мінімально життєздатний додаток, який будуть покращувати через взаємодію з користувачем додатку. Ми не будемо витрачати багато часу на розробку функціоналу, який не буде відповідати реальним потребам ринку. І, навіть, якщо помилимось - то ціна прорахунку буде невеликою

Тому я вірю в Scrum-методологію.