1. **Склади порівняльну таблицю найбільш поширених методологій:**

| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Каскадна модель | Контроль витрат та прозорість для замовника.  Тому що має чітке планування.  Гарантує якість в кінці розробки, тому що передбачає додаткові дослідження на кожному етапі | Має великі затрати. Якщо наприклад на етапі тестування знайдено новий дефект то продукт буде проходити всі етапи спочатку без пропусків. Це буде потребувати багато часу та грошей на зарплату співробітникам | Підійде для галузі з великими ризиками, тому що має чітке планування. Наприклад медицини |
| 2 | Kanban | Наочність роботи, що дає змогу всій команді бачити задачі та результати та легше виявити проблеми та виправити їх | відсутність фіксованих ітерацій, встановлених ролей та обов'язків, а також складності у прогнозуванні та управлінні часом. | Для роботи з рутинними завданнями. Наприклад, для роботи служби підтримки із запитами користувачі |
| 3 | Scrum | Складовою скраму є спрінти, я вважаю це великим плюсом. Спрінти визначають фіксовані часові рамки, протягом яких команда фокусується на виконанні конкретного набору завдань. Це надає прогнозованість і структуру в роботі, дозволяючи легше контролювати та оцінювати прогрес проекту. | Можуть виникати такі виклики, як складність управління, неоднаковий рівень досвіду в команді та необхідність адаптації до специфічних умов проекту | На великих проектах, де треба чітко розподілити ролі та спланувати терміни |

**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?**

Технологічні та бізнес-вимоги до програмного забезпечення постійно змінюються. Метод водопадного управління був найпершим та вважається класичним, але його не можливо використовувати для забезпечення швидкої реакції на ці зміни. Зараз бизнес розвивається так швидко, що якщо ти не випустиш продукт то це зробить хтось інший. Scrum, наприклад, дозволяє випустити продукт з мінімально потрібним функціоналом, а потім доповнити його. Якщо користувацькі потреби змінилися то продукт швидко можно під них адаптувати, і робить його конкурентоспроможним.

Також однією з причин, я вважаю, потребу у більшій взаємодії з клієнтом. Класичні методи надавали обмежений контакт з клієнтом на початкових етапах проекту, що часто призводить до розбіжностей між очікуваннями клієнта і реальним результатом.

Agile допоміг вирішити такі проблеми:

Недостатня взаємодія з клієнтом: Agile надає великий акцент на взаємодію з клієнтом на протязі всього процесу розробки. Це допомагає уникнути непорозумінь щодо вимог та очікувань, забезпечуючи продукт, який краще відповідає потребам клієнта.

Agile враховує інтереси різних зацікавлених сторін у процесі розробки, забезпечуючи їх більш активною участю та залученням до проекту. Це може допомогти залучити додаткові інвестиції до проекту, тому що замовник може прийняти участь в розробці проекту, додати своїх ідей та швидко бачить результат.

Agile пропонує ітеративний підхід до розробки, де продукт швидко випускається у короткі сроки, що дозволяє клієнтам та зацікавленим сторонам швидше побачити результати та вносити зміни. Це вирішує проблему повільності та збільшує гнучкість управління проектом.

Неспроможність адаптуватися до змін: Agile надає рамки для ефективного впорядкування змін у вимогах до продукту, технологічних змін та ризиків. Це дозволяє командам швидко реагувати на зміни та впроваджувати їх без значних перешкод.

Низька якість продукту: Agile сприяє постійному контролю якості, який вбудовується у процес розробки. Завдяки ітеративному підходу та постійній зворотній зв'язку від клієнтів, команди можуть швидше виявляти та виправляти дефекти, що призводить до покращення якості продукту.

Недостатня участь стейкхолдерів: Agile підтримує активну участь стейкхолдерів, включаючи клієнтів, користувачів та інших зацікавлених сторін, протягом всього процесу розробки. Це сприяє кращому розумінню потреб та очікувань, що покращує якість продукту та задоволеність стейкхолдерів.

**2. 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.**

В залежності від бюджету проекту та від додаткових даних ( наприклад таких як наявність облікового запису користувача, на яку кількість платформ робимо додаток) команда може бути як з мінімально потрібною кількістю працівників, так і великою командою. Виходячи з того, що скоріш за все буде розроблятись два додатки: на ios та android, та стартапи часто починаються з невеликої команди, я вирішила що проект буде не великим. Також маю практичний досвід спостерігати як починається та розвивається стартап.

Якщо команда буде невеликою, то я б все ж зупинилася на Канбан, тому що в Канбан немає чітко визначених ролей, таких як Scrum Master, Product Owner та інші. А без чіткого розподілення забов'язань та достатньої кількості людей деталізоване планування було б складним. В тому стартапі, який я знаю, навіть project managera не було. Його забов'язання за суміцнитством виконувала інша людина.

Якщо виходити лише з того, що я має ресурс який мені потрібен та я хочу запустити стартап, то я б однозначно обрала Scrum. Тоді я б отримала:

* ітеративний підхід: Scrum передбачає ітеративний підхід до розробки, розділення роботи на короткі, фіксовані часові рамки, які називаються спринтами. Це дозволяє швидко випускати нові функції та швидко реагувати на зміни на ринку.
* ролі та відповідальності: Scrum визначає чіткі ролі та відповідальності у команді, такі як Scrum Master, Product Owner і Development Team. Це сприяє ефективнішому управлінню проектом та забезпечує, що кожен учасник знає свої обов'язки.
* постійний зворотній зв'язок: Scrum надає механізми для постійного збору зворотного зв'язку від користувачів та зацікавлених сторін. Це дозволяє команді вносити зміни та вдосконалювати продукт у відповідності до потреб користувачів.
* гнучкість управління змінами: Scrum забезпечує гнучкість управління змінами, дозволяючи швидко адаптуватися до змін у вимогах та технологіях.
* фокус на результаті: Scrum ставить акцент на постійному випуску функціональних продуктів. Це дозволяє швидше перевіряти продукт на його користування, а також залучати замовників для збору фідбеку.

Технічне завдання виглядає не складно, тому спочатку я подумала про те, що команда, яка буде працювати, буде невеликою. Для розробки одного додатку знадобиться приблизно: один розробник на фронт енд, один на бек енд, один дизайнер, один тестувальник, можливо один на базу даних. Також на старті працівники можуть замінювати один одного, тому, скоріш за все, це буде команда приблизно 10 людей.

Я точно вирішила, що буду працювати за гнучкою методологією. Через маленький стартап я думала зупинитися на Kanban, щоб не витрачати час на планування. Але я вірю в наш стартап і думаю з часом наш колектив стане більше, ми випустимо ще веб версію, можливо випустимо ще аналогічні застосунки на інші теми. Тому з часом я б перейшла на Scrum. За допомогою Scrum ми забезпечимо ефективне керівництво проектом, швидкі релізи нових версій та гнучку реакцію на зміни.