Рівень: Beet Seed

**1. Waterfall (Водоспад, каскадна модель):**

Сильні сторони:

- Простота управління проектом: кожен етап має чітко визначені завдання.

- Легка документація: через прогресивне виконання етапів.

Слабкі сторони:

- Нереалістичне припущення про стабільність вимог.

- Важко впроваджувати зміни під час виконання проекту.

**2. V-модель:**

Сильні сторони:

- Забезпечує ретельне тестування на кожному етапі.

- Легше керування через чітку структуру.

Слабкі сторони:

- Неспроможність реагувати на зміни вимог у середині процесу.

- Збільшена тривалість проекту через послідовний характер робіт.

**3. Ітераційна модель:**

Сильні сторони:

- Гнучкість та здатність до адаптації до змін.

- Ранні результати та зворотний зв'язок від замовника.

Слабкі сторони:

- Потрібно визначити базовий функціонал на ранніх етапах.

- Збільшення управлінського навантаження.

**4. Інкрементна модель:**

Сильні сторони:

- Швидка поставка функціоналу.

- Можливість виправлення помилок на ранніх етапах.

Слабкі сторони:

- Потрібна чітка стратегія визначення пріоритетів.

- Ризик неправильної інтеграції інкрементів.

**5. Спіральна модель:**

Сильні сторони:

- Здатність до адаптації до змін та ризиків.

- Відображення усіх етапів розробки та їхніх ризиків.

Слабкі сторони:

- Складність управління та контролю за кожним обертом.

- Збільшення тривалості проекту через багатократні ітерації.

Рівень: Beet Sprout

1. На мою думку, Agile-маніфест виник у відповідь на виклики традиційних методологій розробки програмного забезпечення та зростаючу потребу в більш гнучких та спрямованих на співпрацю підходах. Основні причини виникнення Agile-маніфесту можуть бути наступними. По-перше, це динамічний характер ринкового середовища. Споживачі та ринки постійно змінюються, і традиційні методології, які передбачають докладне заздалегідь визначення вимог, не завжди можуть ефективно враховувати цю динаміку. Agile виник як відповідь на потребу у швидкій адаптації до змін у вимогах та середовищі. По-друге, це потреба в ефективній комунікації та співпраці. Традиційні методології, орієнтовані на документацію та процеси, можуть створювати бар'єри для ефективної комунікації між розробниками та замовниками. Agile-підходи покладають акцент на співпрацю, забезпечуючи більшу взаємодію та залучення всіх учасників процесу розробки. По-третє, підвищення вимог до якості та швидкості поставки. У світі, де конкуренція висока, клієнти все більше вимагають високоякісних продуктів та швидких термінів поставки. Agile сприяє ітераційній розробці, що дозволяє постійно покращувати продукт та надавати його частинами. По-четверте, необхідність залучення користувачів та замовників. Agile визнає важливість взаємодії з користувачами на кожному етапі розробки, щоб забезпечити відповідність продукту вимогам та очікуванням замовників.Останнє, це адаптивність до змін у технологічному ландшафті. Швидкий розвиток технологій вимагає гнучких методів розробки, які можуть швидко відгукатися на нові технологічні виклики та можливості. Agile дозволяє ефективно впроваджувати інновації та експериментувати.

2. Agile-маніфест, який був запропонований у 2001 році, мав на меті вирішити низку ключових проблем, з якими стикалася індустрія розробки програмного забезпечення. Основні проблеми, на розв'язання яких був спрямований Agile, та оцінка того, наскільки вони були успішно вирішені, можуть бути викладені наступним чином:

**1) Негнучкість традиційних методологій:** Традиційні методології, такі як Waterfall, часто вважалися занадто жорсткими, що ускладнювало внесення змін після початку проекту. Agile вніс гнучкість у процес розробки, дозволяючи командам легче адаптуватися до змін вимог або ринкових умов.

**2) Потреба у швидкій доставці продуктів:** Agile запровадив ітераційний підхід, який дозволяє командам регулярно та часто доставляти робочий продукт, значно зменшуючи час між плануванням та доставкою.

**3) Недостатня залученість клієнтів:** Agile ставить акцент на тісній співпраці з клієнтами протягом усього процесу розробки, що сприяє кращому розумінню їх потреб та очікувань і, як наслідок, підвищенню задоволеності клієнтів.

**4) Проблеми комунікації в командах:** Agile методології вводять регулярні зустрічі, такі як щоденні стендапи, що поліпшує комунікацію всередині команд та допомагає уникнути непорозумінь.

**5) Недоліки управління змінами:** Agile сприяє постійному оцінюванню та управлінню змінами, що дозволяє проектам залишатися релевантними та відповідати актуальним потребам, навіть якщо вони змінюються протягом процесу.

Чи вдалося Agile вирішити ці проблеми? На мою думку, Agile зробив значний вплив на процес розробки програмного забезпечення, вирішуючи багато з вищезгаданих проблем.

Рівень: Mighty Beet

Як засновник стартапу з мобільним застосунком для обміну світлин котиків, я б обрав Agile з наступних причин:

**1) Гнучкість та адаптивність:** Agile забезпечує швидку адаптацію до змін у динамічному середовищі мобільних застосунків.

**2) Ітеративний розвиток:** Продукт розвивається через короткі спринти, що дозволяє вчасно виправляти помилки та враховувати зворотній зв'язок користувачів.

**3) Зосередження на користувачах:** Agile акцентує на потребах користувачів і їх залученні у розробку.

**4) Спрощення планування та оцінювання:** Ефективне планування та оцінка роботи через керовані спринти.

**5) Підвищення продуктивності та моралі команди:** Сприяння співпраці, самоорганізації та навчання в команді.

**6) Ранній реліз та безперервне вдосконалення продукту:** Швидкий вихід на ринок з MVP та отримання зворотного зв'язку.

**7) Прозорість та зв'язок з зацікавленими сторонами:** Підвищення прозорості та постійний зв'язок з командою, інвесторами та користувачами.