| **Методологія** | **Сильні сторони** | **Слабкі сторони** | **Галузі доцільності** |
| --- | --- | --- | --- |
| **Waterfall** | Простора у виконанні, та чітке управління проектом | Нееластичність до змін, висока ймовірність непридатності | Проекти з чіткими вимогами, де зміни малоймовірні |
| **V-модель** | Ефективне співвідношення між розробкою та тестуванням | Нееластичність до змін, не враховує ризики паралельних процесів | Проекти з фіксованими вимогами та ясною архітектурою |
| **Ітераційна модель** | Можливість адаптації до змін,  ранні результати | Вимагає постійного покращення, може виникнути конфлікт між ітераціями | Проекти з постійною зміною або неясною вимогою |
| **Інкрементна Модель** | Послідовне додавання функціоналу ранній вивід. | Може бути важко управляти великими обсягами інкременітів | Проекти які можна розділити на невеликі функціональні блоки |
| **Спіральна модель** | Фокус на ризиках,  можливість удосконалення продукту | Складність управління, великий обсяг документації | Проекти де ризики важливі та можливості поетапного вдосконалення |

**1.Waterfall**

**Cильна сторона**:Простота у використанні та чітке управління проектом. Це може бути важливим для проектів з фіксованими вимогами та стандартними процесами, де неочікувані зміни малоймовірні.

**Слабка сторона:** Нееластичність до змін. У випадку, коли вимоги змінюються серед процесу розробки, Waterfall може стати непридатним, оскільки не передбачає легкої модифікації.

**Доцільна галузь застосування:** Проекти з чітко визначеними та стійкими вимогами, де можна попередньо спроектувати всі етапи розробки. Наприклад, розробка програмного забезпечення для вбудованих систем.

**2.V-модель**

**Сильна сторона:** Ефективне співвідношення між розробкою та тестуванням, що дозволяє виявляти помилки на ранніх етапах. Це сприяє зменшенню вартості виправлення помилок пізніше у процесі розробки.

**Слабка сторона:** Нееластичність до змін та не врахування ризиків паралельних процесів. Це може ускладнити випадки, коли потрібно внести зміни після початку розробки.

**Доцільна галузь застосування:** Проекти з фіксованими вимогами та ясною архітектурою. Це може бути ефективним для розробки програмного забезпечення, де передбачені функції та структура продукту заздалегідь.

**3. Ітераційна модель:**

**Сильна сторона:** Можливість адаптації до змін. Це корисно у проектах, де вимоги можуть змінюватися середині процесу розробки.

**Слабка сторона:** Вимагає постійного покращення. Це може призвести до того, що процес буде більш витратним.

**Доцільна галузь застосування:** Проекти, де важлива гнучкість і здатність швидко адаптуватися до змін у вимогах. Наприклад, розробка веб-додатків, де вимоги можуть змінюватися відповідно до ринкових потреб.

**4. Інкрементна модель:**

- **Сильна сторона:** Послідовне додавання функціоналу. Це дозволяє швидко виводити на ринок частину продукту та отримувати відгуки.

- **Слабка сторона:** Може бути важко управляти великими обсягами інкременту, особливо якщо потрібно підтримувати внутрішню сумісність.

**Доцільна галузь застосування:** Проекти, які можна розділити на невеликі функціональні блоки. Наприклад, розробка мобільних додатків, де різні частини можуть бути незалежно розроблені та оновлювані.

**5. Спіральна модель:**

**Сильна сторона:** Фокус на ризиках та можливість удосконалення продукту. Це дозволяє пристосуватися до невідомостей та ризиків проекту.

**Слабка сторона:** Складність управління та великий обсяг документації. Може вимагати більше часу та ресурсів.

**Доцільна галузь застосування:** Проекти, де ризики важливі та можливості поетапного вдосконалення продукту є критичними. Наприклад, розробка систем безпеки або критичних систем.