| Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- |
| Waterfall | * легка для розуміння і структурована * повне документування всіх етапів * замовник чітко розуміє кожен процес | * помилки видно тільки по завершенню всіх етапів * внесення правок замовником під час розробки може мати руйнівні наслідки * члени команди націлені кожний на свій процес і не комунікують між собою | * державні установи * створення складних інженерних конструкцій * інженерія будівництва * автомобільна промисловість * фармацевтика |
| Спіральна | * завжди можна зробити крок назад * можна створити додаткові функції пізніше * акцент на аналізі ризиків | * розробка забирає багато часу * не підходить для невеликих проєктів * залучає висококваліфікованих спеціалістів | * банківська сфера * державні установи * дослідницькі проєкти * критичні системи |
| Scrum | * постійний зв’язок з замовником * щоденні мітінги * робота розбита на спринти * невелика команда * визначає чіткі ролі в команді | * замовник іноди сам не знає що хоче бачити в результаті * результат здебільшого залежить від навичок скрам-майстра * потрібен високий рівень самоорганізації | * маркетинг * дослідницькі центри * консалтинг * фін. установи * IT |
| Kanban | * одразу після завершення задачі людина бере іншу * висока гнучкість | * немає часових обмежень * не підходить для довготривалого планування | * IT * кіноіндустрія * HR * маркетинг |
| Ітераційна | * адаптація під потреби користувача * раннє створення працюючого продукту | * можуть бути відомі не всі вимоги до кінцевого результату * потрібна гнучка команда | * електронна комерція * цифровий маркетинг * хмарні технології * ігрова розробка |
| Інкрементна | * поетапна розробка ПЗ * адаптація до умов ринку | * потрібна участь замовника на етапах планування наступного кроку * стабільні вимоги | * розробка ПЗ * інженерія ПЗ * інтеграція систем |

**Waterfall** має послідовну і просту для розуміння структуру, легко керується оскільки має чітко сформульовані вимоги, процес і детально написану документацію. Але вона не гнучка, що не дає можливості запровадити зміни легко і незатратно. Через свою структурованість підходить для галузей, в яких повинні бути чіткі технологічні процеси: машинобудування, медицина, військове виробництво.

**Спіральна модель** адаптується під зміни до вимог та ризиків, в ній постійно аналізуються ризики що допомагає усувати проблеми на ранніх стадіях розробки. Через постійний аналіз і оцінку ризиків розробка може затягнутися, що призведе до великих витрат. Також залучення досвідченої команди теж впливає на витрати і не дає просуватися молодим працівникам. Ця модель буде доцільною в галузях, де важливо враховувати ризики, забезпечувати високу якість продукту: банківська сфера, критичні системи.

**Srum** використовує спринти, які допомагають впроваджувати нові функції, швидко отримувати фідбек, робити прогрес прозробки стабільним, присутність стейкхолдера сприяє кращому розумінню і задоволенню потреб замовника. Щоденні обговорення допомагають швидко усувати проблеми. Відсутність високої самоорганізованості впливає на ефективність і результат процесу, скрам-майстер повинен розумітися на процесах людської взаємодії щоб не виникало непорозумінь і затримок. Гнучкість методу підходить для галузей для яких важлива швидка адаптація до умов і запросів ринку: менеджмент, IT.

**Kanban** підтримує неперервний потік роботи, коли людина завершує завдання і відразу бере інше, допомагає концентруватись на пріоритетності і покращує продуктивність. Фокус спрямований на поточні завдання, що може негативно вплинути на довгострокове планування. Підходить для галузей де потрібно відстежувати процеси і керувати етапами проєктів: IT, управління продуктом.

**Ітераційна** модель завдяки своїй гнучкості дозволяє вносити зміни під час кожного етапу розробки, впроваджувати нові зміни для покращення продукту, швидка розробка і реліз частин функціоналу дає змогу швидко отримати клієнту результат. Через постійну зміну пріоритетів і впровадження змін потрібна гнучка команда, яка зможе швидко реагувати і бути готовою до раптовостей, щоб не було проблем з виконанням ітерацій. Така модель підійде для роботи з компаніями, які постійно розвивають свій продукт: ігрові компанії, цифровий маркетинг, електронні компанії.

В **інкрементній** моделі поетапна розробка і тестування допомагає контролювати процес та вирішувати проблеми щойно вони з’являються, вона дозволяє легко вносити зміни, враховуючи нові вимоги. Ця модель вимагає тісного зв’язку між командою розробки і замовником, якщо комунікація неефективна це може вплинути на прогрес та якість проєкту; якщо вимоги постійно змінюються або визначені нечітко це може ускладнити роботу. Інкрементна модель може застосовуватись в галузях, де потрібно поступово розробляти та тестувати продукт: інтеграція систем, інженерія ПЗ.