| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterall | **Стійка до поновлення кадрів.** Завдяки дуже докладному документуванню кожного етапу, учасники можуть приходити та йти, але на термін роботи це ніяк не вплине.  **Дисциплінує**, завдяки плану та чіткій послідовності етапів та суворому менеджменту.  **Гнучка на ранніх етапах**. До етапу розробки можна легко вносити зміни до попередніх етапів.  **Прозора**. Заздалегідь зрозуміло, на якому етапі відбуватиметься, тому стає простіше прогнозувати бюджети та набирати команду. | **Дуже багато документів**, які потрібно постійно актуалізувати. Через це робота над проектом часто перетворюється на справжню бюрократію — поки що все з усіма не узгодиш, у документах все не пропишеш, з місця ніщо не зрушить.  Найдокладніший план може створити не лише **ілюзію безпеки**, а й **хибні враження** про роботу над проектом. За фразами типу «60% проекту виконано» може бути ніякого корисного результату. Це лише маніпуляція. Все-таки "робити" - не означає "зробити".  **Користувача та замовника повністю ізолюють від процесу розробки**. У результаті перший не може поступово звикати до продукту, а другий вносити коригування, якщо щось йде не так. Тому продукти, виготовлені за каскадною методологією, виявляються не орієнтовані на масового користувача.  Усі вимоги мають бути одразу відомі. Зробити це дуже складно, тому що замовник часто сам не знає, чого він хоче. У таких ситуаціях гнучкі методології зручніші.  Через те, що у Waterfall **тестування відбувається тільки наприкінці,** проектом можуть займатися некомпетентні фахівці, і цього ніхто не помітить, поки не стане пізно. А коли на етапі тестування знаходять вагон і маленький візок проблем, їх починають просто закривати латками, адже іншого вибору немає. На перших етапах модель може бути більш-менш гнучкою, але маса проблем на етапі тестування тягне за собою плачевні наслідки. | Більша частина або вся робота над проектом проводиться на аутсорсі  У вас є чітка концепція продукту, який хочете отримати  Ви не обмежені в часі і ресурсах створення продукту  Створення продукту або бізнесу побудовано на дотриманні суворої послідовності виконання задач.(Напр.Cisco Ericsson AB, Toyota) |
| 2 | Інтеративна модель | **раннє створення** працюючого ПЗ;  **гнучкість** – готовність до зміни вимог на будь-якому етапі розробки;  **кожна ітерація – маленький етап**, для якого тестування та аналіз ризиків забезпечити простіше, ніж для всього життєвого циклу продукту. | **кожна фаза** – самостійна, окремі ітерації не накладаються одна на одну;  можуть виникнути **проблеми з** **реалізацією загальної архітектури** системи, оскільки не всі вимоги відомі до початку проектування. | Для великих проектів;  Коли відомі, принаймні, ключові вимоги;  Коли вимоги до проекту можуть мінятися в процесі розробки. |
| 3 | Skrum | **Прозорість обміну інформації в команді.**Кожен учасник – командний гравець, задіяний в процесі на повну. А замовник завжди знає, на якому етапі перебуває проект, може вносити свої корективи та чітко контролює дедлайни.  **Самостійність команди.** Команда сама планує свою діяльність за демократичним принципом одноголосного голосування. Якщо хтось один в командні не згоден – в ідеалі команда шукає альтернативне рішення, аби всі учасники проекту отримували задоволення і були вповні залучені до процесу створення проекту.  **Результат очевидний.**Scrum дозволяє кожному члену команди бачити свій особистий внесок в розробку проекту і фінальний результат в будь-який зручний час на будь-кому етапі. Це, до речі, дуже мотивує!  **Ринкові ризики зведені до мінімуму.** Команда швидко адаптується до змін, внесених замовником у проект. Зрештою, всі уникають зайвої роботи, а замовник отримує бажаний результат.  **Відсутність фінансових ризиків.** Чим менше *bugs* – тим більше грошей вкладається в бюджет проекту, а не витрачається на виправлення дефектів. Це пов'язано з тим, що члени команди дивляться на проект в одному напрямку і розуміють один одного «з півслова». | **Замовник сам не знає, чого бажає.** Scrum тоді може бути надто ризикованим для команди, яка виконує завдання від такого замовника.  **70% успіху – в руках scrum-майстра.** Від його вміння організовувати команду і залежить ефективність процесу виконання завдання.  **Надто великий проект.** Якщо проект глобальний, до нього доводиться залучати декілька scrum-команд, кожна з яких є самостійною та незалежною. Але тут всі будуть змушені працювати як одне ціле. Це дуже важко, особливо якщо scrum-майстри – люди з дуже різним світоглядом і темпераментом. | Над проектом працює досвідчена, висококваліфікована команда  Ви працюєте над стартапом  Потрібно швидко отримати робочу версію продукту  Замовник виступає в якості партнера, а не інвестора  Продукт розробляється в сфері, схильній до постійних змін.  (Напр. Альфа Банк, Home Credit, Райффайзен Банк) |
| 4 | Kanban | **Гнучкість планування.** Команда концентрується тільки на поточній роботі, пріоритет задачі виставляється менеджером.  **Висока включеність команди в процес розробки.** Завдяки постійним зборам, прозорості процесів і можливостям самоорганізації працівники гуртуються і проявляють щирий інтерес.  **Менша тривалість циклу.** Якщо потрібний навичок має кілька людей, — тривалість скорочується, якщо ж тільки одна людина — з’являється вузьке місце. Тому співробітники повинні ділитися знаннями і тим самим оптимізувати тривалість циклу. Тоді вся команда зможе взятися за роботу, яку забуксувала, і відновити плавний потік.  **Менше вузьких місць**. Ліміти РВП дозволяють швидко знаходити вузькі і проблемні місця, які з’явилися через дефіцит уваги, людей або навичок.  **Наочність.** Коли всі виконавці мають доступ до даних, то вузькі місця легше помітити. Kanban-команди, крім самих карток, зазвичай використовують два загальних звіти: графіки управління і сукупного потоку. | **Система погано працює з командами** чисельністю понад 5 осіб  Він **не** призначений **для довгострокового планування**. | Використовуються в розробці, маркетингу, будівництві, логістиці – в будь-яких рішеннях, де присутній постійний потік завдань.  (Напр. Microsoft, Corbis |