1.

| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Scrum | Команда складається з універсальних фахівців, тому можуть допомагати один одному (етапи проєкту можуть виконуватися паралельно).  Всі члени команди обговорюють задачі, які вони беруть в спринт та дотримуються термінів спринта.  Ретроспектива та **щоденні мітинги**.  Головний показник - продуктивність, тобто кількість виконаних завдань за певний час. Можна встановити приблизний термін закінчення проєкту. | Потрібна чітка специфікація і вимоги для планування спринтів.  Потрібно більше ресурсів (додаткові ролі), наприклад, scrum-master та product owner, що удорожчує проєкт.  Пріоритети завдань в беклозі визначає product owner (суб’єктивний підхід).  Категорично заборонено додавати завдання в поточний спринт, навіть якщо виникають поточні потреби.  Недороблені завдання йдуть назад в беклог (на наступний спринт).  **Щоденні мітинги.** | Для великих проектів.  Для продуктових компаній.  Розробка СRM для підприємства або організації.  Розробка ПЗ для фінансових установ, медичне обладнання |
| 2 | Kanban | Не потрібна чітка специфікація для планування процесу розробки.  Процес розробки лінійний, тому не потрібні додаткові члени команди, керуючі процесом.  Пріоритети завдань в беклозі визначає команда (можливість оптимізації процесу).  Термінові завдання можуть вноситися в будь-який час.  Не обов’язкові щоденні мітинги. | Стадії проєкту виконуються вузкопрофільними командами (фахівцями) не одночасно, а по черзі (спочатку аналітики, потім дизайнери і тільки потім розробники).  Задача може бути в роботі скільки завгодно часу, поки команда її не виконає, тому термін будь-якого етапу невизначений, що ускладнює прогнозування термінів завершення модуля.  Вимірюється середній час проходження завдання по дошці, тому головний показник не виконані завдання, а середній час виконання. | Для невеликих або ж великих довгострокових проєктів.  Для поліпшення вже існуючих продуктів.  Розробка мобільних додатків.  ПЗ для комунальних підприємств (випуск нових версій при зміні методик розрахунків) |

2. Чому з’явився Agile-маніфест?

Методика каскадної розробки проєкту (Waterfall) - одна з перших методик, яка була доречною в свій час. ЇЇ привабливість полягала в чіткій та зрозумілій послідовній структурі процесу розробки (кожний з етапів не починається, поки не закінчиться попередній), наявності вичерпної документації, зрозумілі команді задачі, що не змінюються протягом всього процесу розробки і (що головне для замовника) терміни та вартість проекту, які можна прорахувати до початку розробки. Але ось ця “жорстка” структура, мабуть, і стала передумовою виникнення ідеї та принципів Agile.

Так, замовник “на виході” отримував цілком дієздатний продукт! Але… Існували, на мій погляд, два важливих фактори, які в підсумку, не задовольняли клієнта на 100%. По-перше - закріплена в контракті умова надання тільки кінцевого продукту виключала можливість вносити зміни на будь-якому етапі розробки. Тобто, якщо у клієнта виникала об’єктивна необхідність в зміні якихось умов, навіть якщо це б привело до вдосконалення продукту, йому потрібно було чекати випуску готового продукту і потім вже його поліпшувати. А це автоматично новий проєкт, витрати и т.д.

По-друге - тестування. Тобто, відбувається тестування вже готового продукту. Тут варто пригадати один з принципів тестування - “Раннє тестування економить час та гроші”. І ось цей принцип в такій моделі, на мій погляд, проігнорований. Тобто, ресурси, необхідні для виправлення та повторного тестування можуть бути вивільнені тільки за рахунок їх зменшення на інших стадіях розробки проєкту.

Тому й виникла ідея Agile-підходу до створення гнучких методик, які б визначали замовника (клієнта) як повноправного учасника процесу розробки і дозволяли враховувати зміни, які в перспективі поліпшать якість кінцевого продукту. Цю ідею група IT-фахівців виклала у вигляді маніфесту. А гнучкий підхід був реалізований в найбільш поширених методологіях Скрам та Канбан.

Які проблеми він мав вирішити і чи це вдалося?

Час від початку розробки до отримання робочої версії продукту. Цей фактор реалізований в Scrum у вигляді ітерацій, тобто циклів розробки, які становлять в середньому 2-4 тижні і на виході клієнт вже бачить робочу версію частини продукту, а при їх інтеграції скласти уявлення про кінцевий продукт.

Можливість внесення змін та врахування побажань замовника. Agile-підхід дозволив практично на будь-якому етапі вносити зміни та повертатися в разі необхідності на попередні етапи розробки продукту.

Варіант відсутності вичерпної інформації на стадії збору вимог. В методології Канбан при розробці невеликих проєктів така можливість існує. Тобто вимоги до кінцевого продукту можуть озвучуватися замовником в процесі розробки. Гнучкість методу полягає у врахуванні цих вимог (в межах розумного).

Висока ступінь залучення стейкхолдерів у процес розробки продукту. Команда фахівців самоорганізована і максимально залучена до процесу. Наприклад, тестувальники можуть допомагати розробникам. Іноді до процесу розробки підключається і замовник (клієнт).

Ризик отримати неякісний продукт. При такому підході ці ризики мінімізовані, тому що тестування відбувається на всіх етапах розробки продукту.

3. Я оберу методологію Канбан. Тому що цей продукт, на мій погляд, не містить в собі багато функцій, які б потрібно було розробляти як частини цілого. Якщо б у мене виникла така ідея, я б запустила її “в роботу” якнайшвидше, можливо, навіть, не маючи остаточної картинки як я це бачу. А вже в процесі розробки доповнювала якимись нюансами. Ну і звичайно ж терміни виконання та отримання якісного (хвала QA!) продукту, який я б могла протестувати якнайшвидше)