:росток:Beet Seed — відпрацюй навички на базовому рівні.

Склади порівняльну таблицю найбільш поширених методологій:

| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | Перехід від одного етапу до іншого як тільки він буде закінчений(це сильна сторона тому що процес розробки стає більш простим та зрозумілим). | Неможливість повернутися до попереднього етапу, треба починати все з початку( це слабка сторона тому що, якщо треба щось змінити чи виправити - потрібно все починати знову). | Для простих недовгострокових проектів, наприклад простий сайт-візитівка, де скоріш за все є проста задача, яка не потребує внесення змін. Наприклад, добре для галузі обслуговування чи реклами. |
| 2 | Scrum | Тайм-менеджмент(маємо спринти довжиною декілька тижнів, за які команда зробить визначену частину роботи).Скрам-мастер, який відповідає за організацію процесів. Комунікація в команді та з власником продукту. Все це дозволяє більш продуктивно виконувати розробку продукту. | Щоденні мітинги (витрачається в сумі багато часу на обговорення).  Багатозадачність може бути проблемою, для скрам-мастера може бути складно організувати всі процеси і взаємодію. Для великих проектів потрібно декілька скрам мастерів та команд. | Для проектів де потрібно зробити якісний та перевірений продукт, наприклад в фінансовій сфері чи медицині. |
| 3 | Kanban | Відсутність щоденних мітингів. Немає часових рамок. Задачі пріоритезовані. Всю інформацію, стан та план проекту можна подивитись на дошці. Це дозволяє розробникам бути більш вільними, не бути under pressure, але бути більш відповідальними та самоорганізованими. | Відсутність тайм-менеджменту може затягнути процес. Немає одного відповідального. | Мабуть, підходить для проектів де пріоритетом є якість, а не швидкість. Наприклад, медицина, космічна чи авіа галузі. |
| 4 | Cпіральна | Розробка у вигляді послідовних витків, де на кожному витку додаються нові можливості, перевірка, уточнюється документація. Це добре, тому що дозволяє удосконалювати продукт, вносити неохідні зміни та контролювати іх та процес розробки в цілому. До того ж, власник легко розуміє на якому етапі знаходиться розробка. | Довготривалий процес розробки, що призводить до витрачання великих коштів на розробку. Через це не підходить для малих проектів. | Для великих та довгострокових проектів, коли важливий аналіз ринку та витрат. Це може бути як фінансова галузь, так і технологічна. |

:травы:Beet Sprout — детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

2. Напиши розгорнуті відповіді (0,5 - 1 сторінки тексту) на такі два питання:

* На твою думку, чому з’явився Agile-маніфест?
* Які проблеми він мав вирішити і чи це вдалося?

Agile маніфест з’явився у США 2001 року. Група різних розробників програмного забезпечення вирішили зустрітись для обговорення розробки програмного забезпечення та щоб знайти якесь нове рішення, яке з полегшувало процес розробки на противагу традиційним методам розробки програмного забезпечення, які вважалися вже застарілими, дуже обтяжливими та бюрократичними. Як ми знаємо, розробники довго обговорювали ці проблеми, але так и не дійшли згоди. Але в результаті цієї зустрічі з'явився всім відомий Agile-маніфест. Цей маніфест запропонував більш гнучкі методи розробки та управління проектами.

Загалом, в agile-маніфесті було сформульовано 12 нових принципів, на які слід орієнтуватися в процесі розробки та управління проектом.

На мою думку, Agile-маніфесту вдалося вирішити різні проблеми чи можливо маніфест рекомендує як це вирішити. Головним принципом, як на мене, є необхідність бути більш гнучким на всьому шляху розробки, не боятися впроваджувати зміни навіть на останніх етапах, покращувати процеси та оптимізувати їх за необхідності. Ще одним з принципів є постійна комунікація між усіма членами команди та ж замовником, що сприяє розумінню хто що робить, на якому етапі, чого хоче замовник та показує замовнику прогресс розробки. В результаті впровадження принципів Agile, процес розробки стає більш простим, зрозумілим та відкритим як для команди розробки, так і для замовника. Все це удосконалює процес розробки. До того ж це підвищує мотивацію команди. Ну і звичайно, це дає змогу знизити бюджет.

:лиственное_дерево:Mighty Beet — різнобічно опануй тематику уроку.

| 1. Виконай завдання двох попередніх рівнів.  2. Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків.  Яку методологію ти обереш для процесу розробки і чому? Відповідь текстово обґрунтуй. |
| --- |

На мою думку, застосунок для обміну світлинами котиків - це доволі простий проект, який не потребує довгого планування та багато документації. Користувачу потрібно тільки мати світлини, які можливо було б пошерити завдяки функціоналу застосунку. Я думаю, що waterfall методології було б достатньо. Після планування, формування вимог, аналізу та проектування, ми б могли приступити до написання коду й інтеграції, а потім просто підтримувати цей продукт. Можливо в майбутньому ми могли б розширити функціонал, наприклад способи поширення, але б основна мета застосунку була б та сама.