**Beet Seed**

1. Порівняльна таблиця методологій:

| № | Методологія | Плюси | Мінуси | Доцільність галузі |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | Чіткі вимоги і деталізація, змога ранньго прорахунку необхідних грошей і витраченого часу,простота моделі | Якщо необхідні зміни, повертаємося на перший етап і втрачаємо більше часу і грошей, складність у виправленні помилки, не зрозуміло хто виправить | Медицина, космічні і військові розробки |
| 2 | V-модель | Ранішне і проміжне тестування, наявність результату на кожній стадії | Немає гнучкості і контролю ризиків, код пишеться тільки в середині проекту | Підходить для проєктів з часовими обмеженнями і для проектів з широким тестовим покриттям |
| 3 | Scrum | Фокус на потреби клієнта, можливість швидко адаптуватись до змін, швидкий фідбєк від користувачів | Проєктна документація швидко старіє, важко планувати бюджет | Для проєктів де багато ідей, але немає впевненості напочатку які з них будуть реалізовані,також можуть з’являтись нові ідеї протягом розробки |
| 4 | Kanban | Дозволяє оптимізувати процес, наочність просування процесу, простота | Не підходить для довготривалого планування і великих команд, немає чітких сроків, відсутність ретроспективи | Для повторюваних проєктів, у рамках яких потрібно ретельно стежити за завантаженням окремих виконавців |

1. *Waterfall*- ця методологія підходить для галузей, де прописані чіткі вимоги і передбачають всеосяжний контроль ризиків. Фокус орієнтований на якість продукту, навіть якщо процес займе багато часу, тому модель Waterfall застосовують в сферах де, помилок не повинно бути, як от медицина, космічні і військові розробки. Але основний мінус моделі- якщо щось пішло не за планом, ми повернемося на початковий етап, а це значить, що витратимо більше грошей і часу на розробку і не зрозуміло,хто має виправляти помилки.

*V- модель*- це покращена версія моделі Waterfall. Підходить для проєктів, які вимагають високого рівня тестування,адже в ній тестування починається не в середині проекту, а з перших етапів. Але модель також немає гнучкості, тому можно також понести збитки і труднощі з виправленням, якщо треба щось поміняти.

*Scrum-* гнучка модель, яка дозволяє швидко випускати продукт, отримувати фідбек і змінювати продукт. Замовник може бачити навіть мінімальний резутат робіт і робити правки. Для проєктів, які націлені на тривалий SDLC і які можуть постійно адаптуватися до умов ринку. Мінусом є те, на початкових етапах висока імовірність невдачі, що потребує гнучкості бюджету.

*Kanban*- також гнучка модель, яка націлена на вирішення поставленої задачі. Вона проста в використанні і її легко впровадити, завдяки візуалізації наочно бачно як просувається процес. Але мінусом є відсутність тайм-менеджменту і аналізу в кінці проекту. Не підходить для довготривалих проєктів, але є доцільною для повторюваних або схожих проєктів, де треба слідкувати за процесом і навантаженням кожного члена команди.

**Mighty Beet**

2) Agile маніфест з’явився на світ в 2001 році, після зустрічі представників різних концепцій розробки програмного забезпечення. Я вважаю, що поштовхом для його створення стала необхідність в спрощенні процесу розробки програмного забезпечення. На той час з’явилася гостра необхідність боротьби з занадто застарівшими підходами і принципами. Треба було знайти альтернативу важкому плануванню і розробки з великою кількістю документації. Була необхідність в створенні чогось нового, що буде спрощувати процес розробки ПЗ, щось, що буде допомагати і надихати в створенні нових продуктів і буде основою для можливості адаптуватись до швидких змін.

Agile це система цінностей, яка допомагає тримати фокус тільки на головному, без зайвих формальностей, і створювати продукти набагато швидше й ефективніше. В центрі методології, на мою думку, знаходиться комунікація і готовність до змін. Але також, за допомогою маніфесту були вирішені такі проблеми як:

-прозорість розробки (замовник баче весь процес і має змогу оцінити і видозмінити його)

-продукт стає конкурентоспрможним на ринку (через ранній запуск і гнучкість)

-розробники стають більш заохочені, коли мають простір для втілення своїх ідей, замість жорстких рамок

-”одна голова добре,а дві краще”, при комунікації всіх членів команди ідеї будуть обговорюватись і на виході можна отримати продукт всеобічний, високої якості або створити щось інноваційне.

Я вважаю, що Аgile маніфест чудово впорався з вищеперерахованими проблемами в розробці програмного забезпечення.

**Mighty Beet**

2) Як засновниця стартапу мобільного застосунку для обміну світлинами котиків я б обрала Scrum методолгію. Тому що:

* вога дозволяє випустити продукт на ринок якнайшвидше
* я зможу отримувати реальний фідбек від користувачів і вдосконалювати свій застосунок
* я маю ідею, але не можу сказати, що не захочу щось міняти посеред процесу. Можливо я захочу додати функцію обміну не тільки фото, а й відео
* я б хотіла бачити результати розробки і безпосередньо приймати участь в проекті
* якщо в процесі розробки відбудуться суттеві зміни на ринку (наприклад потенційний конкурент випустить новий функціонал), можна буде з легкістю адаптувати до них мій застосунок