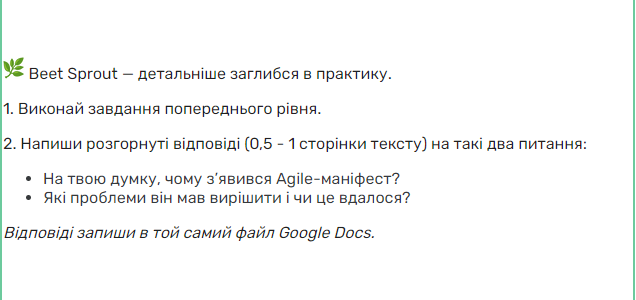


| **№** | **Назва методології** | **Сильні сторони** | **Слабкі сторони** | **Для якої галузі є доцільним** |
| --- | --- | --- | --- | --- |
| 1 | ***Каскадна модель*** | Повне документування = зрозумілість адже виконуються заплановані дії та фази моделі чітко визначені. | Відсутність гнучкості через лінійну структуру - якщо єпотреба повернутися виправити помилку чи внести зміни необхідно пройти ще раз усі фази, а це збільшить витрати та внесе відчутні зміни в планування | У галузях де чітко визначені вимоги з самого початку.  1) Фінансові системи  2) Логічні системи  3) Медичні системи  4) Створення ботів. |
| Планування терміні закінчення роботи, бо робота виконується в логічній послідовності | Надлишкове проектування, щоб уникнути помилок треба багато факторів враховувати. |
| Зручність для учасників проекту - з моменту закінчення своїх завдання на такому проекту можна брати участь в іншому. | У користувачів немає можливості впевнитися у якості розробленого продукту до завершення всього процесу розробки, бо вони не можуть оцінить якість не бачивши продукт. |
| Завдяки зрозумілості структорою моделі може керувати недостатньо досвідчений персонал. | Програмний продукт розробляється за один раз , через послідовне виконання кожного етапу.Немає можливості розбити систему на частини |
| Вимоги до такої моделі не змінні, тому можна використовувати її, як шаблон в який є можливість помістити методи для виконання. | Для кожної фази створюються результативні дані, які по його завершенню вважаються замороженими. Це означає, що вони не повинні змінюватися на наступних етапах життєвого циклу продукту. |
| При уважності виконанні роботи дефекти виявити можливо на ранніх стадіях, без великих витрат. | Працівників поділяють в залежності від їхніх навичок на «perfect» та «code monkeys». |
| 2 | ***V-модель*** | Легке управління - кожна фаза має чіткі цілі. | Пізнє тестування - неможливість вносити зміни без впливу на графік. | Вимоги та критерії чітко визначені проект є коротким, використовуються не динамічні технології та інструменти.  1) Медичні системи.  2) Банківські системи.  3) Оборонні проекти. |
| Тестування з самого початку - дефекти можна виявити на ранніх стадіях. | Недостатньо гнучка для динамічних змін - цього не передбачено. |
| Суворі етапи виконання - виконуються паралельно ліва - процес розробки, а права - тестування. | Аналіз ризиків не передбачений. |
| Переважно для малих і середніх проектів. | Створення програми в середині процесу розробки - відбувається на етапі написання коду. |
| 3 | ***Спіральна модель*** | Прозорість для користувачів - на ранніх етапах мають змогу побачити реалізацію та приймати участь в плануванні, аналізі, виконання оцінювальних дій. | невизначе -  ність з вартістю проекта - якщо проект низький за рівнем або невеликий за розміром. Оцінка ризиків після проходження кожного витка спіралі пов'язана з витратами. | 1) Для систем "Розумний дім"  2)Для державних систем.  3) Для аерокосмічних системи |
| Гнучкість можливість вносити зміни і додавання нової функціональності навіть на відносно пізніх етапах. | Складність застосування для розробників менеджерів та розробників, через складну структуру. |
| Чіткість в документації. | Високопрофесійні працівники. |
| Висока якість - завдяки зворотному зв'язку. | Нескінченність - оскільки кожна реакція у відповідь замовника на створену версію може породжувати новий цикл, що віддаляє закінчення роботи над проектом (прийняття загального рішення про припинення процесу розробки) |
| Висока продуктивність - завдяки повторним використанням властивостей | Обробка внутрішньої та зовнішньої документації - існує велика кількість проміжних стадій. |
| Передбачуваність дій - за допомогою поставлених цілей. | Надмірність -використання моделі може виявитися дорогим і навіть неприпустимим за коштами, оскільки час, витрачений на планування, повторне визначення цілей, виконання аналізу ризиків та прототипування. |
| Фінансування - не потрібно розподіляти кошти заздалегідь. | Перепризначення розробників - при виконанні дій на етапі поза процесом розробки виникає. |
| 4 | ***Agile*** | Якість продукту - забезпечується за допомогою вбудованого тестування та регулярних перевірок робочого продукту на протязі розробки. | Менш передбачуваність - для деяких програмних продуктів розробники не можуть повною мірою кількісно оцінити необхідні зусилля, особливо на початку життєвого циклу розробки великих продуктів. | Широко використовується в маркетингу, освіті та навіть у виробництві автомобілів. |
| Збільшення прибутку - додаючи деякі переваги у наступних релізах, ви продовжуєте розвивати продукт. | Постійна взаємодія з замовником - зміни в проекті,що вносяться можуть бути частково незрозумілими для розробників, які повинні вносити змінити своєчасно. |
| Прозорість - оскільки гнучка методологія розробки заохочує залучення користувачів до спільних злагоджених зусиль. | Сфокусованість на деталях - ризик втратити глобальний орієнтир всього проекту. |
| Зниження ризику - оскільки команда виявляє та виправляє будь-які проблеми на ранній стадії. | Відсутність документації - новим членам команди складно зрозуміти, як їм діяти. |
| Фактично дедлайн виключено - так як відсутній чіткий план, дедлайн адаптується під зміни. | - |
| Відсутність рутини - документація займає мало часу. | - |



2.1 На мою думку Agile - маніфест з'явився для пришвидшення роботи працівників, дешевша вартість продукту для замовника, що задовольняє його потреби. Замовника в свою чергу може на етапі розробки змінювати вимоги до ПЗ та підлаштовувати до сучасного ринку потреб на будь- якому етапі. Завдяки такій розробці готовий продукт є висококонкурованим. Agile є гарним рішенням для розробни.

2.2 Методологія Agile була розроблена з метою вирішення ряду проблем, пов'язаних з традиційними (водоспадними) методами розробки програмного забезпечення. Мала вирішити проблеми:

Низької комунікації та співпраці;

Недостатньої гнучкості в змінених вимогах замовлення;

Швидкості та ефективності розробки;

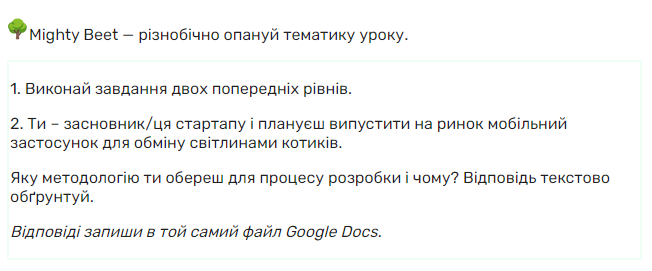
Зменшити документацію;

Мінімізувати витрати;

На виході продукт має бути конкурованим;

Ризики не повинні викладатися на останній момент.

Методологія Agile успішно вирішує ці проблеми, однак вона також має свої недоліки та вимагає від учасників проекту достатнього рівня комунікації, гнучкості та дисципліни, щоб працювати ефективно.



Для свого стартапу мобільного застосунку обміну світлинами котиків я обрала б гнучку методологію Agile. Зараз мобільні додатки активно розвиваються, створюються нові функції, які роблять додатки більш цікавими та широко використовуваним. Пришвидшений вихід розробки продукту є однією з переваг, бо перевірка ідей та реакція на них допоможе зробити додаток зручним та конкурентоспроможним, а головне затребуваним.