**Перший рівень**

Порівняльна таблиця найбільш поширених методологій

|  | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1. | Waterfall | 1. Повне документування кожного етапу.   Сильна сторона, адже це підвищує продуктивність роботи та сприяє ефективному пошуку дефектів.   1. Чітке планування термінів та витрат.   Сильна сторона, тому що бізнес знає строки релізу продукту, його виходу на ринок, а також від початку знає суму капіталовкладень.  Команда розробки орієнтується в строках виконання роботи, це теж покращує ефективність розробки ПЗ.  3. Модель є легкою для розуміння та використання.  Її застосування є зрозумілим для працівників, не потребується спеціальне навчання для організації роботи за цією моделлю.  4. Якість має першочерговий пріоритет.  Сильна сторона, адже документування кожного етапу, чіткість, зрозумілість вимог на початку розробки сприяють створенню якісного ПЗ. | 1.Необхідність затвердження всіх вимог ще на першому етапі Reguirements.  Негнучкість моделі. Ринок постійно вимагає нового. Тому модель розробки ПЗ має бути готовою до змін на будь-якому етапі.  2.Якщо виникає потреба у внесенні змін до вимог, то процес розробки повертається до першої стадії Reguirements та переробки всього зробленого.  Слабка сторона, адже це затягує вихід готового продукту на ринок (розробка ПЗ стає довгою) та збільшує витрати (продукт стає значно дорожчим).  3.Найчастіше проблеми виявляються на етапі тестування.  Одним із принципів тестування є раннє тестування, завдяки якому виправлення дефекту є більш дешевим для компанії.  4. Багато документації. Часто із сильної сторони може стати слабкою.Адже велика кількість технічної документації може бути обтяжливою для замовника та користувача.  5.Клієнт не бере участь у створення продукту після постановки ТЗ.  Відповідальність за продукт покладається на команду розробників. | Воєнна, авіабудування, космічна галузь, медицина та фінансовий сектор.  Цим сферам потрібні чітко вибудувані процеси та терміни, які пропонує Waterfall. |
| 2. | Scrum | 1. Постійний зв'язок із замовником.  Команда розуміє вимоги, в разі потреби корегує їх.  2. Адаптивність до потреб ринку.  Модель є гнучкою в реагуванні на зміни вимог. Вимоги можуть змінюватися навіть наприкінці проєкту.  3. Короткі ітерації.  Кожна ітерація (2-4 тижні) закінчується готовим продуктом ( або його елементом), який може швидко доставлятися замовнику на перевірку. Також короткі ітерації дають змогу швидко доставляти нові версії продукту на ринок.  4.Висока ефективність команди.  Завдяки методу підвищується відповідальність кожного працівника за результат. Під час щоденних нарад (Daily Scrum Meeting або Stand-Up Meeting) учасники розглядають та усувають проблеми. | 1.Обмеження на розмір команди (5-9 членів команди).  З одного боку, велика команда - це швидкий процес розробки. З іншого, у великій команді неможлива ефективна комунікація. А комунікація в середині команди - це один із принципів  Scrum. | Цю модель використовують світові технологічні гіганти Xerox, Honda, Canon, Amazon, Toyota, Google, а також незлічена кількість ІТ компаній.  Командно-орієнтований підхід, відсутність бюрократії, швидкий вихід продукту на ринок - зробили цю модель однією із найбільш затребуваних. |
| 3. | Kanban | 1.Обмеження за завданнями.  Сильна сторона, адже таким чином відбувається регулювання навантаження. Наприклад, один член команди може одночасно бути залученим до виконання не більше 2-4 задач. Ще це сприяє зосередженості команди на найголовнішому.  2.Візуалізація роботи.  Kanban дошка показує, над чим працює розробник, що тестить тестувальник, чим зайнятий дизайнер, які задачі ще не в роботі. Це також покращує взаємодію в команді.  3.У будь-який момент учасники можуть вносити зміни до розробки.  Сильна сторона, адже це сприяє покращенню продукта.  4. Відсутні постійні наради.  Безперервний потік завдань зменшує час на планування. | 1.Немає часових обмежень.  Це може стати мінусом, тому що в учасників може не виникнути відчуття терміновості в роботі. Хоча це може “нейтралізуватися” швидкістю проходження завдання по Kanban дошці (саме так оцінюється ефективність роботи)  2. Оскільки немає конкретного керівника, жоден учасник не несе відповідальності за дотримання командою цілей.  Я загалом вірю в людей, але, на мою думку, краще, коли є хтось, хто керує і бере на себе відповідальність за роботу команди.  3.Можна пріоритетні завдання “пропустити”, оскільки на дошці ця інформація не відображається.  Це суттєвий недолік, так як може виникнути ситуація, коли важлива функціональність продукту буде розроблятися, тестуватися значно пізніше. Це призведе до затримки виходу продукту на ринок та фінансових втрат. | Виробництво, маркетинг та ІТ.  Метод нагадує безперервний виробничий процес |

**Другий рівень**

**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?**

**Пояснення до виконання завдання:** я перечитала велику кількість матеріалів про методології та методи розробки ПЗ. І на даний момент повна каша у використанні цих термінів. Звернула увагу на те, що в ДЗ першого рівня потрібно охарактеризувати основні методології, як приклад наводиться

В 2001 році на гірськолижному курорті зібралися 14 розробників ПЗ, щоб спільно виробити нову методологію управління проектами. Досягти згоди їм не вдалося. Але вони створили Agile-маніфест, який став проривом у управлінні проектами. Хоча метод Scrum з'явився ще в 1995 році.

Чому виникла така потреба? На мою думку, головних причин є дві.

По-перше, кінець ХХ - початок ХХІ сторіч - це час швидких, бурхливих змін в технологіях, виробництві, фінансовій та банківській сферах. Світ стрімко змінював своє обличчя. Тому використовувати старі бюрократичні технології не мало сенсу. Прибічники таких систем ризикували залишитися на узбіччі стрімкого злету технологій. Тому сам час диктував необхідність змін. **“Готовність до змін важливіша за дотримання плану!”**

Що пропонували звичні технології? Вичерпну документацію, планування, наради, прискіпливу увагу до контрактів, досконалість процесу та інструментів. Це важливі складові успішного процесу. Але їх домінуюча роль на межі століть перетворилася із переваг на недоліки. Час вимагав змін.

По-друге, кінець ХХ століття остаточно закріпив новий погляд на людину. Нова філософія сприйняття людини, її місця в житті, думаю, що спонукали творців маніфесту закріпити окремим пунктом (першим пунктом!) роль та місце ЛЮДИНИ в новому світі та новій методології. Людина не “винтик та шпунтик”, не “раб” виробництва, не “сліпе кошеня” в руках бізнесу. ЛЮДИНА - основа всього і важливіша за все. За процеси та інструменти. Кожен крок у розробці ПЗ за гнучкою системою наповнений увагою та турботою про людину.

(Далі переходжу до відповіді на 2 запитання). Проходячи відбір на навчання по програмі Фонду народонаселення України при ООН, я писала у мотиваційному листі, що турбота та піклування про людину у великих ІТ компаніях вражає, окрилює та надихає. В цей страшний час війни компанії зуміли потурбуватися про кожного свого працівника, його родину, його безпеку. Всі (від джуна до найвищих щаблів) ніхто не був покинутим чи забутим. Тому що **“Люди та співпраця важливіші за процеси та інструменти”**. А якщо згадати мирне життя, то перша цінність маніфесту втілилася у регулюванні навантаження на працівників, наданні всебічної допомоги новачкам, створенні умов для підвищення кваліфікації, гідній заробітній платні - все це мотивує людину працювати ефективно та швидко.

Вичерпна документація чи працюючий продукт? Чому надати перевагу? Можна безкінечно довго вдосконалювати вимоги, документацію, а час спливає. Ніша на ринку заповниться іншими продуктами. Втрати бізнесу неминучі. Тому **“Працюючий продукт важливіший за вичерпну документацію”.**

Agile-маніфест мав задекларувати перехід до гнучкої методології розробки ПЗ. І, на мою думку, це вдалося.

**Третій рівень**

**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.**

**Пояснення до виконання завдання:** я перечитала велику кількість матеріалів про методології та методи розробки ПЗ. В результаті повна каша стосовно використання цих термінів. Звернула увагу, що ДЗ першого рівня пропонує охарактеризувати основні **методології** і наводиться приклад Waterfall ( хоча ми вчили це як **модель)**. Тому даю відповідь про ітеративну модель. Ще й тому, що методологію SCRUM молодий стартаповець навряд чи втілить в життя зі своєю невеличкою командою.

Якби я, ось така як зараз (лише почала навчання, опановую теорію, але не маю практичного досвіду) стала засновницею стартапу, то, напевне, обрала б ітеративну модель розробки ПЗ. Чому?

1. У мене була б ідея створити мобільний застосунок для обміну світлинами котиків. І на початок розробки більше ніяких задач. Думаю, що бізнес мені сказав би, що цього мало, потрібно окрім обміну світлинами додати ще щось, щоб збільшити кількість користувачів, а,значить, і дохід. Ітеративна модель дозволяє додавати нові функціональності в процесі розробки.
2. Я, як власниця стартапу, хотіла б швидше вийти на ринок ( тому що в боргах, як у шовках, члени команди не мають гідних умов праці, чудова ідея про світлини котиків може спасти на думку комусь іще. Ітеративна модель дозволяє виводити на ринок продукт після чергової ітерації, а потім продовжувати його вдосконалювати, додаючи нові функціональності.
3. Оскільки Ітеративну модель рекомендують застосовувати до великих проєктів, я б розвинула свій проект у великий (наприклад, додала б функцію обміну/продажу/віддам у хороші руки; або функцію “Обери своєму котику зачіску” або “Стрункішаємо разом із котиком”( котики часто бувають загодованими).

Ітеративна модель - це ескіз, із яким художник поступово вирішує що робити далі.

АЛЕ не виключено, що коли я наберуся досвіду, то оберу іншу методологію чи метод.