**Перший рівень.**

1. Склади порівняльну таблицю найбільш поширених методологій.

| № | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | Дана модель краще всьго підходить для розуміння замовникам, котрі дуже далекі від IT- галузі.  Легко відстежувати контрольні точки в проекті. | Через дуже чітку послідовність дефект в продукті може коштувати дуже дорого.  Замовник отримує результат тільки після завершення всіх єтапів розробки. | Підходить для довготривалих проектів у котрі залучають велику кількість співробітників. |
| 2 | V- модель | Покращена версія Waterfall, оскільки контролювати можна кожний єтап процесу розробки продукту.  Покращений тайм менеджмент співробітників. | Ця модель коштує дорожче для замовника ніж Waterfall.  Складніше для сприйняття замовникам котрі дуже далекі від IT- галузі. | Підходить для маленьких проектів.  У проектах, в яких існують часові та фінансові обмеження. |
| 3 | Ітераційна модель | Раннє створення працюючого ПЗ.  Готовність до зміни вимог на будь-якому етапі розробки.  Кожна ітерація – маленький етап, для якого тестування та аналіз ризиків забезпечити простіше, ніж для всього життєвого циклу продукту. | Можуть виникнути проблеми з реалізацією загальної архітектури системи, оскільки не всі вимоги відомі до початку проектування. | Для великих проектів.  Коли вимоги до проекту можуть мінятися в процесі розробки. |
| 4 | Інкрементна модель | Адаптація до умов ринку.  Швидший відгук замовника з приводу кожної розробленої версії. | Вимоги мають бути сформовані на початку проекту.  Має лінійну структуру схожу на каскадну модель. | Підходить у тому випадку, якщо є чіткий план дій, але продукт потрібно запустити досить швидко, а зміни можна буде вносити потім. |
| 5 | Cпіральна модель | Покращений аналіз ризиків.  Хороша документація процесу розробки.  Можливість внесення змін і додавання нової функціональності навіть на відносно пізніх етапах.  Раннє створення робочих прототипів. | Може бути досить дорога у використанні.  Управління ризиками вимагає залучення висококласних фахівців.  Успіх процесу у великій мірі залежить від стадії аналізу ризиків.  Не підходить для невеликих проектів. | Коли важливий аналіз ризиків і витрат.  Великі довгострокові проекти з відсутністю чітких вимог або ймовірністю їх динамічної зміни.  При розробці нової лінійки продуктів. |

**Другий рівень.**

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

1. На твою думку, чому з’явився Agile-маніфест?

Маніфест гнучкої розробки— це документ, що описує основні принципи, на яких базується [гнучка розробка](https://uk.wikipedia.org/wiki/%D0%93%D0%BD%D1%83%D1%87%D0%BA%D0%B0_%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0). Agile-маніфест створювався для того щоб робити кращий продукт оминаючи деякі бюрократичні аспекти, тому що бігатня з паперами займає майже половину всього терміну розробки, тому Agile-маніфест направлений більше на якість продукту, ніж на чітке дотримання контракту. Навіть якщо за для якості потрібно відійти від плану і зробити кращий продукт.

1. Які проблеми він мав вирішити і чи це вдалося?

Отже, принципи Agile існували до створення Маніфесту. Люди застосовували для розробки програмного забезпечення. Ці цінності було зафіксовано у Маніфесті Agile. Потім ці принципи взяли та почали застосовувати у роботі.

Найвищим пріоритетом для Agile- маніфесту є задоволення потреб замовника,

шляхом завчасного та регулярного постачання програмного забезпечення.

Схвальне ставлення до змін, навіть на заключних стадіях розробки.

Agile-процеси надають можливість використовувати зміни задля забезпечення конкурентоспроможності замовника.

Працюючий продукт слід випускати якомога ча тіше, з періодичністю від пари тижнів до пари місяців.

Впродовж усього проекту розробники і представники бізнесу повинні працювати разом щодня.

Над проектом повинні працювати вмотивовані професіонали.

Щоб робота була виконана, створіть їм умови, надайте підтримку і повністю на них покладіться.

Особиста комунікація – найефективніший та найпрактичніший метод як донести інформацію до команди, так і поширити її всередині.

Працюючий продукт – головний показник прогресу.

Інвестори, розробники і користувачі повинні мати можливість підтримувати

постійний ритм як завгодно довго. Agile допомагає налагодити такий сталий процес розробки.

Постійна увага до технічної досконалості і якості проектування підвищує гнучкість проекту.

Простота – мистецтво мінімізації зайвої роботи – вкрай необхідна.

Найкращі вимоги, архітектурні та технічні рішення виникають у командах, що здатні самоорганізовуватись.

Команда регулярно намагається знайти способи підвищення ефективності

та відповідно корегує свою роботу.

Я можу з упевненістю відповісти: так. Маніфест, як і раніше, актуальний. Думаю, зараз він актуальний як ніколи раніше.