**Перший рівень**— відпрацюй навички на базовому рівні.

|  |  |  |
| --- | --- | --- |
| ***Назва методології*** | ***Сильні сторони*** | ***Слабкі сторони*** |
| **Waterfall**  ***Підходить*** для складних довготривалих проектів  ***Не підійде*** для розробки комп’ютерних ігор, мобільних додатків | 1. Прозора та зрозуміла для замовників,  2. Можна легко відслідковувати точки контролю та дедлайни,  3. Зручна для документального оформлення відносин за наявності чітко визначених контрольних точок. | 1. Результати по проходженню всіх етапів  2. Модель не передбачає залучення замовника в деталі процесів розробки (важка комунікація).  3. Висока вартість дефектів.  4. Внесення значних змін підчас процесу розробки руйнівні для всього процесу. |
| **Scram**  ***Підходить***  для розробки ПЗ, ігор, але також може використовуватись в інших галузях, таких як реклама, будівництво, планування заходів, різні стартапи  ***Не підійде*** для великих та довго тривалих проектів, які мають чітку специфікацію вимог та процеси, які неможливо протестувати, поки не протестуєш попередній процес | 1. Команда працює короткими етапами, на кожному з яких визначає цілі та шляхи їхнього досягнення, що прискорює процес роботи;  2. Присутній відкритий обмін інформацією, що робить процес максимально прозорим;  3. Скорочується час на пошуки помилок і пояснення проблем;  4. Мінімізація фінансових ризиків завдяки оперативній реакції на зміни та фіксення багів. | 1.Успіх проекту залежить від скрам-майстра, кваліфікації та відданості своїй справі команди;  2. Вимагає регулярної комунікації із замовником, що часом гальмує процес через неможливість отримання фидбеку;  3.Складно адаптувати скрам під великі проекти |
| **Kanban**  ***Підходить для*** групи підтримки програмного забезпечення або служби підтримки***.***  При управлінні стартапами без чіткого плану, але де активно просувається розробка.  ***Не підійде*** для великих та довго тривалих проектів, які мають чітку специфікацію вимог та процеси, які неможливо протестувати, поки не протестуєш попередній процес | 1. Команда концентрується тільки на поточній роботі, пріоритет завдання виставляється менеджером.  2. Використання цього методу заохочує членів команди висловлювати пропозиції щодо покращення;  3. Підтримує вирішення проблем і управління ризиками: завдяки чіткому розумінню того, як все працює та як насправді виконується робота, можна перейти до більш раціонального, об’єктивного обговорення питань, проблем і ризиків. | 1. Не може працювати автономно;  2. Не має чітких часових рамок для кожного етапу, якщо ви не встановите їх самостійно;  3. Не підходить для середовища з короткими виробничими процесами. |

**Другий рівень — детальніше заглибся в практику.**

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

На мою думку причина появи Agile-маніфест пов’язана з агрегацією досвіду, який накопичувався рокам в різних компаніях - від практиків, які самі створювали продукти та зіштовхувались з певними складнощами та проблемами через зміну оточуючого середовища, зміну вимог та очікувань замовників і користувачів протягом розробки продуктів. Працюючи на продуктами знаходили нові методи, нові практики, нові інструменти та інше. Змогли сформувати певні цінності.

* ***Які проблеми він мав вирішити і чи це вдалося****?*

Проблеми, які мав вирішити Agile:

1. Гнучкість та здатність підлаштовуватись під потреби замовника та ринку;

2. Отримання на виході досконалого продукту, який проходить перевірки вкінці кожного спринту;

3. Покращити комунікацію між замовником та всередині команди;

4. Прискорити пошуки та виправлення помилок.

Agile має і певні недоліки, а саме часто відсутність конкретного терміну для релізу всього продукту через регулярні зміни. Також, необхідність наділяти команду повноваженнями та незворотність культурних змін в усій організації. До недоліків можна ще віднести тривалість і вартісність Agile-трансформації.

**Третій рівень — різнобічно опануй тематику уроку.**

*Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків.*

Для свого стартапу я б безумовно обрала б методологію розробки SCRUM. Я вважаю, що ця методологія добре підходить для стартапів, дозволяючи розбивати проєкт на цілі і підзадачі, а також виділяти терміни. Також, мінімізація фінансових ризиків завдяки оперативній реакції на зміни та фіксенню багів. Для себе я ще виділяю - присутність відкритого обміну інформацією, що робить процес максимально прозорим та економить час на пошуки та виправлення помилок 😊