**Перший рівень:**

**Scrum**

**Сильні сторони**

- Скрам майстер як рушійна сила проекту, його присутність на всіхєтапах розробки ПЗ як приклад та мотивація усім працювати швидше краще бо він як ніхто вболіває за результат.

- Самоорганізованість та самостійність команди. Близькі відносини між усіма членами команди допомагають як при вирішенні різних питань типу тестувальник запитує у розробника як що є щось незрозуміле з багом.

- Дейлі мітинги - гарний приклад відслідковування етапів виконання завдань, а також можливість вирішення проблем у тих чи інших питаннях на етапі їх появи.

- У працю залучений тестувальник. Це дозволяє швидко реагувати на помили у системі на будь яких етапах розробки, та значно скоротити витрати як час так і гроші.

- Гнучкість системи. В нашому мінливому світі де все змінюється швидко ця система має можливість пристосуватися до будь яких змін.

- Використання дошки завдань. Це дуже крута штука тому що замовник як і всі учасники створення ПЗ можуть бачити на якому етапі вони знаходяться, які завдання у роботі а які тільки треба вирішити.

**Слабкі сторони**

- Стрес пов'язаний зі строками виконання спринтів. На початку психологічно важко пристосуватись до такого темпу роботи, але потім це проходить.

- Замовнику іноді не зрозуміло як працює ця методологія. Треба витрачати час на пояснення принципів цієї методології

- Відсутність документації. Дуже важко у процесі розробки долучити нову людину так як їй практично дуже важко буде пояснити що як і навіщо робиться зараз і буде робитись далі.

- Не можливо запланувати точну дату закінчення. Так ця методологія працює на основі спринтів але вони можуть бути різними за часом і це заважає у визначення дати релізу.

Така модель може використовуватися як для малих проектів так і для середніх і великих, область розробки це ігри, додатки мобільні

**Спіральна модель**

**Сильні сторони**

- Гнучкість проектування. У процесі проходження будь якого витку ми можемо вносити зміни на будь якому етапі.

- Прорахунок ризиків. Після розробки специфікації(документації) вона відправляється відділ розрахунків ризиків що є важливою складовою у коштовних проектах типу банківська сфера.

- Нові функції можливо додавати на пізній стадії розробки

**Слабкі сторони**

- Частіше використовується для великих проектів. Таких проектів не так багато.

- Оцінка ризиків. Вона проводиться на кожній стадії розробки я є дуже затратною у фінансовому плані.

- Відгуки замовника. Можливість постійно залишати відгуки замовником несе з собою часті зміни що кличе за собою постійно змінюються строки завершення ПЗ.

Така модель підходить тільки великим проектам з великим бюджетом, тому що у ній є прорахунок ризиків на усіх витках а це значні витрати у фінансах але вона вигідна наприклад у банківський сфері або у медицинських продуктах.

**Другий рівень**

На твою думку, чому з’явився Agile-маніфест? - Я думаю що була потреба у змінах, існуючі методології та методики не встигали за розвитком світу і не вирішували всю проблематику яка виникала у процесі розробки ПЗ. Також на той час сфера розробки відокремилась (виросла) із того що було на той час, що почали з'являтися нові ідеї більш новітні та прогресивні ніж те що вже використовували багато років. Мабуть що усі тези з маніфесту з'являлися поступово та перевірялись на практиці можливо декількома роками до того як маніфест побачив світі. Вихід цього маніфесту була можна сказати кульмінація наробок багатьох команд та багатьох років праці. Також це була можливість показати що можна працювати краще простіще цікавіше вигідніше відразу для двох сторін і для замовника і для команди розробки.

Які проблеми він мав вирішити і чи це вдалося? - Маніфест мав вирішити питання відносин замовник - виконавець і в першу чергу пояснити що це повинна бути двостороння розмова і вони повинні один одного чути. Що на мою думку вдалося. Також гостро стояло питання великої кількості документації як не доречна коли треба щоб продукт швидко почав працювати а не виконувати тисячу і одну забаганку яку внесли до документації. Я думаю що цей маніфест круто вирішив цей тонкий момент з документацією, хоча і тут є деякі мінуси що в умовах відсутності документації практично неможливо додати нову людину до вже існуючого або розробляємого проекту. Такого було присутнє питання порозуміння у команді. На мою думку воно доречне у маніфесті і є правдивим але як показує практика воно не вирішене повністю, так як й досі існують суперечки наприклад між девелоперами та тестувальниками. Готовність до змін у теперішньому швидкоплинному світ одна й з найважливіших частин маніфесту і я повністю згодна з висловом про те що коли замовник приходить на розробку йому це треба було зробити ще учора, а до моменту випуску продукту його треба змінювати десятки разів щоб наздогнати просто час і знайти й задовольнити кінцевого користувача.

І у завершення хочу додати що Agile манифест на мою думку класна розробка, але зважаючи на людський фактор цей маніфест не може задовольнити всіх. Бо всі різні і бувають як складні замовники так і складні розробники або взагалі бувають складні продукти. Так маніфест це круто але кожна команда/компанія обирає сама з якими критеріями цього маніфесту працювати а з якими ні.

Третій рівень

Оскільки в мене стартап тобто мені потрібно швидко розібратись зі своїм позиціонуванням, а також швидше випустити на ринок продукт я б обрала Ітераційну методологію. Вона швидка і зручна у першу чергу тим що я отримаю робочий продукт вже з декількох інтерацій, відповідно і фідбек від користувачів буде швидким. Легко вносити зміни в ході розробки тому що всі етапи повторюються на кожній ітерації. Також це вигідно з точки зору фінансів.