**Перший рівень**

**Склади порівняльну таблицю найбільш поширених методологій: Надаючи відповіді, обґрунтуй свою думку: чому це саме сильна/слабка сторона/доцільна галузь застосування.**

| *№* | *Назва методології* | *Сильні сторони* | *Слабкі сторони* | *Для якої галузі є доцільною* |
| --- | --- | --- | --- | --- |
| *1* | *Waterfall*  *2. V - модель*  *3. Ітераційна*  *модель*  *4. Спіральна*  *модель*  *5. Scrum*  *6. Kanban* | *+Чітка послідовність дій*  *+Повне документування кожного з етапів*  *+Планування термінів та витрат, необхідних для реалізації проекта*  *+Для замовника більш проста та прозора модель, він може знати всі складові та може їх відслідкувати, на всіх етапах роботи*  *+Контроль поточного процесу на кожному з етапів*  *+Планування тестування та верифікації системи проводяться на ранніх етапах*  *+Можливість проміжного тестування*  *+Відбувається раннє створення готового ПЗ*  *+Ітерації не займають багато часу, це дає можливість доставляти частину функціоналу та продукту, за короткий проміжок часу*  *+Гнучка система, готова до змін на будь- якому етапі розробки*  *+Простіше провести тестування та аналіз ризиків, для невеликої частини життєвого циклу продукту*  *+Можливість оцінки ризиків після кожної ітерації*  *+Приділяється більше уваги початковим етапам розробки - аналізу та проектуванню*  *+Оновлення вимог до проекту, яке уточнюється*  *+Допускає коригування в процесі роботи, що дає можливість покращити якість продукту*  *+Дозволяє користувачу “побачити” систему на ранніх етапах*  *+Підвищує продуктивність, завдяки можливості використання властивостей, які були використані для створення попередньої версії продукту*  *+Залученість всієї команди, у кожного члена команди своя роль*  *+Чітка, спланована робота, протягом спрінта, зрозуміле розподілення задач*  *+Команда має можливість брати більш великі проекти, та грамотно розподіляти роботу в них, контролюючі при цьому не тільки етапи роботи, а й моральний стан та відносини всієї команди*  *У менші розділі простіше вносити корективи*  *+Робота в невеликих спринтах може допомогти команді знайти більше помилок перед випуском продукту*  *+Зручна система постановки задач та організація робочих процесів*  *+Групування завдань*  *+Контроль термінів виконання*  *+Покращення ефективності роботи*  *+Наочність просування роботи* | *-До початку роботи над проектом потрібно підтвердити повні вимоги до системи*  *-Немає можливості повертатися на попередній етап, для внесення змін, доведеться переробляти все повністю*  *-Ризик збільшення витрат часу та коштів на виправлення помилок (тобто, ціна помилки потенційно стає вищою)*  *-Застаріла система*  *-Модель не є гнучкою*  *-Написання коду у середині процесу*  *-Немає динамічного внесення змін*  *-Кожна фаза - окрема одиниця, окремі ітерації один на одного не накладаються*  *-Не завжди вимоги відомі до початку тестування, що ускладнює роботу загальної архітектури системи*  *-Якщо проект має низьку ступінь ризику, або якщо він невеликий, дана модель може бути для нього занадто дорогою*  *-Оцінка ризиків після кожного проходження спіралі зв’язана з великими витратами*  *-Спіраль може продовжуватися до нескінченності, тому що постійна реакція замовника на створену версію може сприяти появі нової спіралі, що значно відтягує завершення роботи над проектом*  *-Можливе складне адаптування, до нового способу ведення справ (в тому сенсі, що не всім людям подобається та підходить одне й те саме)*  *-Всі повинні бути командними гравцями, але не всі можуть відповідати цьому*  *-Не всі полюбляють щоденне спілкування:)*  *-Не підходить для довгострокового планування*  *-Не підходить для великих команд* | *Проекти з чіткими вимогами, які не потребують додаткової роботи (банківські структури, держ. установи)*  *Проекти, які мігрують з однієї платформи на іншу, коли від компанії розробника не вимагають робити тести, тому що тести планує робити замовник або інша сторона.*  *Використовується, коли є часові та фінансові обмеження, задачі, які потребують більш широкого тестового покриття*  *Доречна для крупних проектів, коли відомі тільки ключові моменти, та коли ми точно знаємо, що вимоги будуть змінюватися, за ходом роботи*  *На мою думку, буде більш доречною для невеликих та простих проектів, оскільки, через постійні “покращення” можна застрягнути на початковому етапі створення ПЗ*  *Можу бути неправа - буду рада дізнатися правильну відповідь:)*  *Доречна будь- якій компанії, котра в приорітетах має співробітників та якість продукту, не виставляє жорстких вимог до документації, а цінить реальний результат командної роботи*  *Застосовується в командах підтримки або обслуговування*  *Доречна для невеликих команд (максимум 10 людей)* |

***Другий рівень***

***2. Напиши 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?***

*Я трошки шукала інформацію стосовно того, як він взагалі з’явився, і можу зробити з цього певні висновки, що цей метод з’явився на основі того, що, зазвичай, не влаштовує працівників у роботі.*

*Те, чого їм не вистачало під час роботи у різних проектах, вдалося об’єднати та створити майже ідеальну сферу для працівників (майже - тому що не всім людям вона може підійти, бо всі ми різні:).*

*Для мене ця модель нова, я ніколи раніше не чула про неї, але мені це дуже близько. Тому що модель, спрямована на те, щоб цінити та “чути” кожного із співробітників, це вау! Обговорення етапів роботи кожного працівника, спільне вирішення проблем, які виникають у будь- якого члена команди, це неймовірно круто, коли не “кожен сам за себе”, а дійсно командна робота.*

*Як на мене, то це найкраща модель, як для працівника, так і для замовника. Тому що добре мотивовані люди в команді працюють краще, ніж ті, що мають від керівництва тільки вимоги, і не мають можливості бути почутими.*

*В цій системі люди в пріоритеті, тому що взаємодія між людьми важливіша за процеси та інструменти.*

*Та якщо в команді існують якісь принципи, умови, інструменти, які можуть заважати роботі - вони обов’язково будуть усунені.*

*Люди самі можуть обирати спосіб організації роботи, набір процесів, інструментів. Саме ці процеси мають допомагати роботі, а не заважати їй.*

*Ще один важливий принцип роботи в цій методології - це те, що працюючий продукт завжди важливіший за вичерпну документацію, таким чином, вся команда направлена на результат більше, ніж на документування процесу, але зовсім без документів не можна.*

*Ціла команда має завдання №1 - доставити працюючий продукт, функціонал замовнику.*

*Співпраця із замовником важливіша за погодження умов контракту, що також допомагає мати гарні відносини із замовником.*

*Готовність до змін важливіша відповідності початковим планам, тобто, в результаті, замовник може отримати навіть кращій функціонал, ніж планувався на початку, тому що на кожному етапі роботи команда готова допрацьовувати його, за бажанням замовника.*

*б) Хм, як на мене, то це значно спростило роботу в команді, тому що:*

*-всі члени команди мають свою роль у розробці проекту;*

*-в команді існує людина, котра бачить кінцеве призначення та вигляд продукту, що розробляється, та може грамотно організувати роботу в команді, та ніхто більше не має цим перейматися;*

*-в команді також є людина, котра допомагає влитися в команду, пояснює, як саме налагоджені процеси, що теж дуже добре;*

*-команди мають спринти, та обмежений список завдань на період кожного спринта, що значно покращує емоційний стан співробітників, та позитивно впливає на просування роботи;*

*-ретроспектива у кінці робочого процесу дає команді можливість зрозуміти, що вдалося під час роботи над реалізацією цього проекту, а що заважало, або що можна покращити. Цей постійний моніторинг не лишає ніякого шансу на те, що хоч хтось у команді був чимось незадоволений, або не викладався під час роботи на повну, тому що працювати в атмосфері підтримки та цінування кожного з членів команди - дуже мотивуюче. (Але, можливо, я помиляюсь:)*

***Третій рівень***

***2. 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.***

*Розробка мобільного застосунку - це, насправді, результат великої роботи, постійний коллів з власником, для максимального співпадіння очікування та реальності застосунку.*

*Я би обрала систему Agile, тому що це класний варіант, який мені дасть можливість підтримувати постійний зв’язок з виконавцями мого застосунку, я зможу коректувати їх на різних етапах. Також, я зможу отримати демо свого застосунку, та скорегувати, чи все правильно робиться, чи все так, як я собі мріяла, та отримати деякі відгуки від користувача, або від команди розробника, стосовно функціоналу мого застосунка. Гнучка система, у цьому кейсі, спрацює найкраще, як на мене, тому що, знову ж таки, мотивовані, командні “гравці” зможуть зробити мій застосунок якісним.*

*Якщо спиратися тільки на постанову питання, в тому сенсі, що в мене є певна ідея мого застосунку, та додавати би я нічого не хотіла, наприклад, то мені, звичайно, підійшла система Waterfall. Але я знаю себе, в мене може бути 100 ідей за день, тому цей метод мені би точно не підійшов, як людині - замовнику. До того ж, у цього методу є певна послідовність, яку не можна прервати, а я, може, щось же захочу доповнити, у процесі, а не маю фінансових можливостей на пост виправляння/доповнення чого завгодно у моєму застосунку.:)*