**1. Beet Seed** — відпрацюй навички на базовому рівні.

Склади порівняльну таблицю найбільш поширених методологій:

| **№** | **Назва методології** | **Сильні сторони** | **Слабкі сторони** | **Для якої галузі**  **є доцільною** |
| --- | --- | --- | --- | --- |
| 1 | водоспадна | кожен етап має чіткі цілі, допоки вони не будуть досягнуті, поки не будуть протестовані усі вимоги - перехід до наступного етапу неможливий. така чітка структура підвищує якість кінцевого продукту та забезпечує кращий контроль над процесом його створення  - достатньо передбачувана та послідовна модель | - не гнучка система. якщо на етапі тестування буде знайдено суттєву помилку - неможливо повернутися на сходинку назад та виправити, треба тільки починати весь цикл спочатку, що також може підвищувати вартість проєкту  - вимоги затверджуються тільки на початку, що не дає змоги внести корективи на подальших етапах  - результат видно лише наприкінці проєкту | -авіа промисловість  -банківська справа  -держсектор  -машинобудування  -інженерія  -медичне обладнання  це усі сфери, які вимагають високої точності бо є високоризиковими  -або подібні проєкти, які мігрують з однієї платформи до іншої. в такому випадку методологію можна лишити, бо змінюється лиш місце розміщення проєкту |
| 2 | v-модель | - тестування відбувається на кожному етапі, що підвищує шанси знайти суттєві помилки  - більше можливостей впливати на фінальну якість продукту, знову ж таки завдяки багатоетапному тестуванню  -чітка структура, бо є послідовні етапи з визначеними цілями | - не підійде проєктам, де висока вірогідність потреб до змін  - не зовсім гнучка модель, потребує чітко визначених вимог відпочатку  - на етапі модульного тестування якщо знайдеться суттєва помилка, то так само буде важко вносити зміни, бо вимоги були затверджені (хоч і протестовані) на початкових стадіях  - метод орієнтується на перевірку, а не на адаптацію | там, де важлива точність, передбачуваність та надійність, тому ті самі сфери, що і у водоспадній моделі. використовуються там, де ціна помилки надто висока |
| 3 | інкрементна | - достатньо гнучка, можна швидко вносити зміни  - відносно швидко можна отримати готовий продукт  - швидше отримується фідбек від кінцевого користувача  - наявність готового продукту після кожної версії | - регулярні зміни можуть впливати на структуру та процес створення продукту  - важко сформувати фундаментальні вимоги від початку | crm, e-commerce, ios, android, мобільні додатки  там, де легко почати з невеликого базового функціоналу, та легко розширювати його з кожним релізом, поступово |
| 4 | спіральна | -дозволяє додати додатковий функціонал на будь-якому етапі розробки, бо це передбачає концепція спіралі  - постійна оцінка ризиків, дуже корисно у складних проєктах  -поступова розробка, можна вчасно виявити проблему | - дуже вартісна, бо треба час та кошти на планування, аналіз та реалізацію кожного витка спіралі  - внесення змін може створювати нові ітерації, що збільшує час на створення продукту | -авіапромисловість  - банківська справа  - медичне обладнання  в цілому використовується в тих самих галузях, що і водоспадна/v-модель, але в проєктах, де потрібна більша гнучкість та оцінка ризиків після кожного етапу |
| 5 | scrum | - активна взаємодія з командою та зацікавленими сторонами/представником замовника, що дає можливість швидко реагувати на будь-які зміни  - з пункту вище я б додала, що витікає адаптивність методу, бо є можливість оперативно реагувати на зміни на ринку/в процесах | -зниження якості продукту, якщо не всі задачі з беклогу будуть виконані або якщо на які-небудь задачі стоїть мінімальний пріоритет чи горять дедлайни | розробка сайтів, додатків, e-comerce, ігор - усе, що потребує динамічної розробки/впровадження нових даних/підтримки після, де треба експериментувати та швидко реагувати на фідбек кінцевих користувачів. |
| 6 | kanban | -гнучкість з таймінгом та пріоритетністю. усі задачі розташовані на дошці. порівняно зі скрамом, тут немає спринтів/часових обмежень та можна змінювати пріоритетність задач протягом періоду розробки. | -відсутність чітких ролей, що вимагає підбір команди з високою самодисциплиною, що може викликати складнощі  - переваги методу також можуть бути і слабкими сторонами: якщо постійно змінювати пріоритетність задач на ходу, в потоці виконання дійсно важлива задача може бути витеснена менш критичною. | доцільна для техпідтримки, коли є постійний потік задач |

p.s. якщо я правильно зрозуміла, agile - це система, гнучкий підхід. або щось по типу майндсету. а скрам та канбан - це вже методи, які базуються на цьому agile підході.

*Надаючи відповіді, обґрунтуй свою думку: чому це саме сильна/слабка сторона/доцільна галузь застосування.*

**2. Beet Sprout** — детальніше заглибся в практику.

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?

маніфест з'явився тому, що старі методи розробки програм були надто складними, повільними та з великою кількістю помилок. перші електронні прилади переключались лише тумблерами. потім, винайшли першу мову програмування, але цього також було замало. якісної комунікації між замовниками та програмістами не було, що призводило до створення достатньо сирих продуктів, які працювали не так, як було заплановано. поява водоспадної моделі зробила тогочасний прорив.

класичні, не гнучкі моделі розробки використовувались багато років.вимоги замовників змінювались на протязі розробки, але внесення коректив було непростим процесом, тому кінцевий продукт не завжди відповідав очікуванням. всередині команди розробників продукту комунікація страждала, кількість документації ускладнювала роботу. стало зрозуміло, що треба змінювати підхід.

у 2001 році зібралася група досвідчених спеціалістів, які поділяли єдині цінності щодо підходів в розробці продукту. вони намагались сформулювати та зафіксувати основні принципи та ідеї роботи. зрештою, вони домовились щодо багатьох аспектів роботи, результат їх зусиль ми маємо у вигляді agile маніфесту.

порівняно з класичними методологіями, де створення кінцевого продукту було повільним та складним процесом, підхід agile допоміг значно пришвидшити роботу, розбити її на етапи, створюючи кінцевий продукт готовими (або майже готовими) частинами (ПРАЦЮЮЧИЙ ПРОДУКТ, а не багато бюрократії - документації). це, в свою чергу, додало можливість отримувати зворотній звʼязок від замовника (СПІВПРАЦЯ ІЗ ЗАМОВНИКОМ, а не сліпе слідування контракту) та змогу команди шляхом спільного обговорення та дій (ЛЮДИ ТА СПІВПРАЦЯ, а не сталі процеси) швидко реагувати на нього (ГОТОВНІСТЬ ДО ЗМІН, а не дотримання “водоспаду”). це покращило як якість продукту, так і пряму комунікацію з замовником.

на мою думку, маніфест суттєво спростив роботу, бо чітко сформулював та сформував те, до чого, можливо, багато людей інтуїтивно прийшли на той час самостійно, але не могли узагальнити та вивести в єдину, об'єднану концепцію.

**3. Mighty Beet** — різнобічно опануй тематику уроку.

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.

ДУЖЕ дуже класне завдання!! 😍😍😍

я обирала між двома методологіями: інкрементною та скрамом.

обидві методології базуються на розробці продукту частинами, обидві можуть доволі швидко випустити щось робоче, навіть якщо це не повна версія, що власне класно для стартапу, але в обох є свої нюанси.

особисто я обираю скрам, тому що в нього є перевага - вимоги, які постійно оновлюються та змінюються під час проєкту. це дає нам високу адаптивність під ринок, на відміну від інкрементної моделі, де вимоги фіксуються на початку проєкту, а їх зміна може потягнути за собою додаткову переробку якихось готових елементів (що в свою чергу забере додатковий час та кошти).

в нашому випадку ми маємо стартап, де навіть якщо є сформована концепція/план/бачення продукту, ми не можемо гарантувати, що кінцевого користувача одразу задовольнить продукт. у скраму залученість користувачів висока, регулярно збирається фідбек і у команди є можливість оперативно відреагувати на нього.

більш того, після кожного спринта у нас буде частина готового продукту, який можна буде швидко відправити в реліз. умовно, на першому спринті виходить головна сторінка, на другому - особиста стрічка, де можна публікувати фото, на третьому - можливість додавання в друзі і т.д. паралельно збирається фідбек користувачів: не вистачає функції категоризації за породами котиків чи можливості залишити реакцію під фото? пріоритизуємо задачу та додаємо у наступний спринт. це дуже гнучко.

тому, для нового проєкта, який тільки “стає на ноги” з динамічними/змінними вимогами, де ще немає стабільності як у великих компаніях, де треба знижувати ризики (завдяки частому релізу та реагуванню на фідбек) я обираю методологію скрам.