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

Склади порівняльну таблицю найбільш поширених методологій:

| Назва методології | Сильні сторони | Слабкі сторони | Галузь застосування |
| --- | --- | --- | --- |
| **Waterfall** | суворе послідовне виконання всіх етапів, кожен з яких повинен завершитися перед початком наступного. |  | Будь яка галузь, де існує виробництво деталей, з яких потім збирається виріб |
| **V-model** | на кожному етапі відбувається контроль поточного процесу, для того, щоб переконатися в можливості переходу на наступний рівень |  | людино-машинного інтерфейсу охоплює інженерні рішення, котрі забезпечують взаємодію оператора з керованими ним машинами. |
| **Spiral model** | Даний інваріант висвітлює важливість усієї системи та довгострокові проблеми, що охоплюють весь її життєвий цикл |  | [Інформатика](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)  [Комп'ютерна інженерія](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D1%96%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D1%96%D1%8F)  [Виробнича інженерія](https://uk.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D0%BE%D0%B1%D0%BD%D0%B8%D1%87%D0%B0_%D1%96%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D1%96%D1%8F)  [Історія програмної інженерії](https://uk.wikipedia.org/w/index.php?title=%D0%86%D1%81%D1%82%D0%BE%D1%80%D1%96%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D1%97_%D1%96%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D1%96%D1%97&action=edit&redlink=1)  [Менеджмент](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%BC%D0%B5%D0%BD%D1%82)  [Керування конфігурацією](https://uk.wikipedia.org/wiki/%D0%9A%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BA%D0%BE%D0%BD%D1%84%D1%96%D0%B3%D1%83%D1%80%D0%B0%D1%86%D1%96%D1%94%D1%8E)  [Управління проектами](https://uk.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D1%94%D0%BA%D1%82%D0%B0%D0%BC%D0%B8)  [Управління якістю](https://uk.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D1%8F%D0%BA%D1%96%D1%81%D1%82%D1%8E)  [Загальне управління якістю](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D1%8F%D0%BA%D1%96%D1%81%D1%82%D1%8E) |
| **Scrum** | передбачає роботу короткими відрізками – спринтами. Тривалість мінімального робочого циклу – від 1 до 4 тижнів. Як правило, результат спринту – це робочий продукт з очікуваним приростом функціоналу. | відсутність довгострокового плану дій; часто недостатньо опрацьована документація;необхідність постійного залучення замовника; певні вимоги до чисельності та рівня підготовки команди. |  |
| **Kanban** | це метод поліпшення робочого процесу і частина філософії agile. В його основі — «Маніфест гнучкої розробки програмного забезпечення | Обмеження на розмір команди. Спосіб підходить для команд чисельністю до 5-10 осіб. З більшою кількістю співробітників стає важко відстежувати прогрес у роботі Короткострокове планування. Канбан-методологія не призначена для довгострокового планування. В цьому і полягає його суть - в відставання відправляються тільки актуальні завдання, а їх пріоритет змінюється відповідно до ситуації. | В основному виділяють три канбан-напрямки - виробниче, програмне і персональне. |
| **Extreme programming (XP)** | Замовник отримує саме той продукт, який йому потрібен, навіть якщо на початку розробки сам точно не уявляє його кінцевий вигляд.  Команда швидко вносить зміни в код і додає нову функціональність за рахунок простого дизайну коду, частого планування і релізів.  Код завжди працює за рахунок постійного тестування і безперервної інтеграції.  Швидкий темп розробки за рахунок парного програмування.  Висока якість коду.  Знижуються ризики, пов'язані з розробкою, тому що відповідальність за проект розподіляється рівномірно.  Витрати на розробку нижче, тому що команда орієнтована на код, а не на документацію. | Успіх проекту залежить від залучення замовника, витрати часу на проект, тому що на початку ніхто не знає повного списку вимог.  Успіх XP сильно залежить від рівня програмістів, методологія працює тільки з senior фахівцями.  Менеджмент негативно відноситься до парного програмування  Регулярні зустрічі з програмістами дорого обходяться замовникам.  Вимагає занадто сильних культурних змін.  Через нестачу структури і документації не підходить для великих проектів. |  |

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

**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?**

**Agile - маніфест**  з'явився в процесі вдосконалення методів розробки програмного забезпечення займаючись безпосередньо розробкою та допомагаючи у цьому іншим.Найвищим пріоритетом для нас 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 частіше, з періодичністю від кількох тижнів до кількох 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 спосіб як донести інформацію до команди, так і поширити її в ній.  
 Працюючий продукт – головний показник прогресу.Інвестори, розробники і користувачі повинні мати можливість підтримувати постійний ритм стільки, скільки потрібно. 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 самоорганізованих командах.Команда регулярно намагається знайти способи підвищення ефективності та відповідно корегує свою роботу.  
  
**Третій рівень** — різнобічно опануй тематику уроку.

1. Виконай завдання двох попередніх рівнів.

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.   
  
Як на мене система принципів, сукупність ідей, понять, методів, способів і засобів, що визначають стиль розробки програмного забезпечення для створення додатку з обміном світлинами чудово підійде методологія Канбан. Легко візуалізувати, розробка додатку не потребує великої команди просто вкластись в часові рамки цієї методології.Також Канбан є частиною Agile-філософії в цьому закладена комунікація з командою , гнучкий метод еволюційного управління змінами в проектах. Це означає, що існуючий процес вдосконалюється невеликими кроками (еволюційно). У підсумку, роблячи багато невеликих змін (а не одне велике), ми зменшуємо ризики для всього проекту в цілому.

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

**1. Виконай завдання двох попередніх рівнів.**

**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.**

***Відповіді запиши в той самий файл Google Docs.***