Назва методології Сильні сторони Слабкі сторони Для якої галузі є доцільною  
  
**1. Waterfall**   
  
Сфери - точні сфери промисловості, науки, фінансів. Наприклад розробка програми запуску космічної ракети.   
Сильні сторони  
- Поетапність системи. Тобто ми починаємо від планування і закінчуємо релізом, що дає чітке розуміння витраченого часу на проект та його фіксування.   
- Точність виконання процесу. Даний проект будет дуже точний, тому дана модель використовується для роботи в технологічно складних сферах (медицина, ракетобудування).   
- Зрозумілість етапів. Кожен етап чітко прописаний, що дає чітке розуміння задачі і її вирішення.

| Слабкі сторони |  |
| --- | --- |

- Тестування проводиться після етапу розробки.Із-за цього сама вартість помилки дуже висока, як в часі так і в вартості.   
- Сама її більша слабкість в її суті. Система чітка, але не гнучка із-за чого виникає багато ускладнень і труднощів.   
- Досконалість етапів. Кожен етап повинен бути зроблений ідеально, без цього інший етап немає сенсу. Наприклад, відкрити молочну ферму, але купити коней. Ферма молочна є а молока нема. (Приклад дуже притягнутий за вуха)  
  
**2. V-модель (V-model)**   
Сфери - фактично ті ж самі, як і в каскаді, але які потребують більшого тестування. Наприклад розробка автопілота.   
 Сильні сторони  
- Строга етапність. Працюємо по суворому сценарію, тому процес достань чітки та оптимізований.

- Тестування відбувається на кожному етапі. Це поліпшує дану модель, так як помилки можна усунути на самому початку та виправити дефекти системи.  
- Чіткий тамінг. Кожний етап також прописується по часу, тому ми конверт розуміємо коли закінчимо проект і можемо представить чіткий дедлайн замовнику.   
- Також можна додати всі плюси каскадної системи.

- Тестування займає дуже багато часу. Та як кожний етап підається ретельному тестування, на це йде багато часу та ресурсу. Можливо деякі етапи не потрібно тестувати, але вони тестуються.  
- Метод дорогий. Так, як ми проводимо часте тестування, написання тест- планів чек-листів. Вартість проекту буде більше ніж наприклад при каскадному методу.   
- Малогнучка система. Хоча конжний етап тестується, згідно данної системи ми не можем повертатися назад.   
  
  
  
  
**3. Інкрементальна модель.**    
Сфери - можна використати в будь яких сферах. Створення реклами, сайту, сфери дизайну та проектування.   
  
 Сильні сторони  
- Результат після кожного етапу. Ви бачите “кінцевий” результату після кожного етапу розробки.   
- Зручність в презентації. Замовник, може бачити якийсь готовий блок після першого етапу.   
- Ефект конструктора. Фактично ваш проект буде складатися з готових блоків. І після закінчиння проекту його можно продовжити будувату. Дуже зручно, щось додати.

Слабкі сторони  
- Важко змінити. Так, як ми даємо готові блоки, їх дорого замінити, або внести зміни.  
- Фактичний результат в кінці. Кінцевий результат ви бачите в кінці і тільки тоді розумієте, як взаємодіють всі блоки між собою.   
- Самостійність кожно блоку. Задачі можуть розділитися на декілька команд, та в цілому не спрацювати при інтеграції на один об’єкт.  
  
**4. Ітераційна модель**Сфери - також проекти які зручну бувати циклічно. Як на мене дуже зручно для ігор. Для прикладу, хочу навести Skyrim. Гра після релізу неоднократно вдосконалювалася нарощуючи контент, поліпшилася графічно, інтегрувалася на різні платформи PS4, VR, Steam…  
  
Сильні сторони  
- Поступова розробка. Після першого етапу розробки ми отримуємо готовий функціонал проекту. Спочатку будуємо шасі машини без кузова, ми можемо поїхати. Потім проектуємо дизайн кузова, який стане на шасі. Після складаємо салон, та мультимедію, потім покраска і в Женеву на виставку.   
- Раннє тестування. Ми тестуємо кістяк проекту, що дає змогу мінімізувати витрати по проекту і оптимізувати його на ранніх етапах.   
- Прозорість. Візьмемо приклад з машиную. Замовник бачить початковий етап і розуміє в, що він вклав гроші, що дає змогу оцінити вигідність проекту. Сам процес стає для всіх безпечний.   
  
Слабкі сторони  
Мінусів немає)) Жарт!  
- Часті зміни. На кожному етапі можливі внесення зміни, що збільшує навантаження.   
  
**5.Extreme Programming (XP)**  
Сфери - маленькі проекти які потребують швидкої реалізації. Типу рекламна сторінка, оналйн зустрічі, інформаційна стаття… і таке інше. Сайти які не мають багатьох функцій.   
  
Сильні сторони  
- Швидкість реалізації. Суть цього метода в швидкий та ефективний розробці проекту.  
- Невеликий штат працівників. Це дозволяє економити фінанси.   
- Чітке розуміння кінця проекту. Так , як в основном даний етап використовується для реалізації маленьких проектів. Умови і задачі зрозуміли з самого початку.   
- Простота тестування. Об’єкт тестування не великий тому, як проекти самі по собі не об’ємні тому це теж економія і швидке вирішення поставлених задач.   
  
Слабкі сторони  
Якщо розглядати в розрізі невеликих проектів, сблокостей доволі не багато. Так, як всі помилки доволі просто виправити.   
- Мінливість умов та вимог. Так як проекти швидкі самі по собі, тому можуть часто змінюватися умови.  
  
**6. Scrum**Мій фаворит.   
Сфери - можна застосувати до всього. Так як вона дуже гнучка. Не вигідна тільки для невеликих проектів. Я б застував її навіть в робочому процесі який не пов'язаний з розробкою.   
  
Сильні сторони  
-Контроль. Кожний етап сильно контролюється, так як для реалізації скраму, потрібно чітко виставити задачі і розуміти що конкретно робиш ти.   
- Повне розуміння вимог. Скрам починається зі створення документації. Який пишется для всього проекту. Потіс розділяється на блоки\етапи. Перед початком кожного етапу пишеться backlog, в якому прописані конкретні задачі на спрінт. Щоденні мітинги на тему задач.  
- Видимість результату. Після закінчення кожного спринта, ви бачите готовий результат, який працює так, як і планувалося.   
- Гнучкість. Завжди можна внести та редагувати backlog, тільки не забудь повідомити.   
- Дедлайн. Кожний спринт має чітко визначений період реалізації. Тому завжди зрозуміли сроки.  
Слабкі сторони  
- Не підходить для малих проектів. Тут і додати нема чого) Скрам затратний процес по часу та фінансах, тому ліпше його використати для чогось великого.   
- Мотивація. Скарам потребує великій циклічності іноді працівники можуть виорати, так як довго працюють над одним проектом. Монотонність задач, щоденні мітинги. З часом це виснажує інтровертів.   
- Ефективний для малої команди. Як ми говорили вище для скраму потрібно проводити щоденні мітинги. Тому об’єктивно барати в команду невелику кількість команд, щоб всі стигли висловитися. Тому можна розділити задачі на декілька окремих команд.   
  
**7. Канбан**  
Сфери - розробка автомобілів Toyota. Я вважаю, що кандан можна застосувати фактично в будь яких сферах.   
  
Сильні сторони  
- Дошка. Головний елемент канбану це дошка з задачами. Всі цілі та задачі, знаходяться на електронній платформі по типу дошки (Trello, Jira).Тому учасники бачать всі задачі, які потрібно виконати для закінчення проекту, що створює ефект прозорості проекту, ви завжди в курсі своїх пройдених, даних та майбутніх задач.   
- Пріоритетність. Всі задачі робляться по пріоритетності, дає змогу ефективно працювати над вирішення ключових питань.  
- Самореалізація. Суть в тому, що кожна нова задача потрапляє на дошку після виконання попередньої. Тому процес рухається швидше ніж Scrum.  
- Оптимізація процесу. Філософія канбану, полягає в швидкому вирішенні поставлених задач та економії витрат.   
  
Слабкі сторони  
- Вигорання працівників. Навантаження поступово збільшується, тому можливе вигорання самураїв.   
- Кожна система не ідеальна. Тому існує різні типи Agill методів. Щось підходить для данного проекту, а щось ні. Сам канбан також розрахований на невелику групу людей, яку легше контролювати. Але я вважаю, що це може збільшити час на реалізацію проекту із-за недостатньої кількості працівників.