**Перший рівень**

| **Название** | **Сильная сторона** | **Слабая сторона** |
| --- | --- | --- |
| **Waterfall**- Подходит крупным компаниям, компаниям с ограниченным бюджетом временем, текучкой среди сотрудников. | 1. Закрывает задачи крупных компаний   (задан четкий алгоритм действий, нет дополнительных встреч и изменения сроков)   1. Из-за четких инструкций терпима к изменению кадров ( поскольку на определенном этапе один из сотрудников может уволиться имея четкие инструкции процесс можно передать другому сотруднику) 2. Можно планировать время и бюджет поскольку все заранее запланировано.( ТЗ пишется на самой первой встрече с заказчиком, поэтому команда работает по четко заданному алгоритму и можно запланировать время и бюджет) | 1.Заказчику важно четко понимать, что он хочет, а это бывает достаточно сложно(поскольку продумать все до мелочей не так то просто.  2. Пользователь и заказчик не взаимодействуют друг с другом (это плохо потому что разработчик не может вносить корректировки если столкнется с трудностями)  3. Тестирование происходит в самом конце ( плохо тем что тестирование это крайнее звено разработки и на этом этапе может обнаружится масса недочетов которые придется закрывать костылями в лучшем случае.) |
| **V-модель** подходит компаниям с ограниченным бюджетом и временем при этом с меньшими рисками что что-то пойдет не так поскольку тестирование начинается на ранней стадии | 1. Тестирование проводится на ранних стадиях ( уменьшает риски некачественного продукта имеющего массу костылей)   2 Суровая этапизация (Дает четкое понимание последовательности действий | 1. Отсутствует динамика(нет возможности вносить изменения) 2. Создание программы происходит в середине процесса ( нет возможности параллельной работы с вовлеченными в процесс сотрудниками, все четко описано и переходит из этапа в этап) |
| **Итерационная модель-**подходит для компаний которые не имеют четкого сценария от а до я, позволяет вносить изменения по ходу разработки | 1. Коммуникация с заказчиком( позволяет создать продукт соответствующий требованиям пользователя и заказчика, 2. Непрерывное тестирование (позволяет оценивать успешность проекта) 3. Возможность оценки реального состояния продукта(позволяет заказчику понимать что проект будет успешно завершен) | 1. Нет четкого понимания используемых ресурсов разработки( в процессе могут понадобится дополнительные ресурсы. что повлияет на бюджет и время)   могут затянуться)   1. Нет целостного понимания продукта ( Нету четкого понимания какие будут возможности и ограничения в конечном продукте) 2. Снижается ответственность сотрудников (У недобросовестных сотрудников просыпается осознание что написанное ранее можно будет переделать и они не так усердно делают свою работу) |
| **Инкрементная** модель подходит компаниям которым необходимо быстро поставить на рынок продукт с базовыми функциями | 1. Получение функционального продукта после реализации каждого инкремента (дает возможность так сказать “пощупать” что же получается на каждом этапе 2. Улучшение понимания требований для более поздних инкрементов( за счет того что каждый этап разработки дает вполне видимый результат, практическое использование дает пониманию как его улучшить) | 1. Возможность изменения в технологиях ( может привести к изменению графика работ) 2. Нету четкого определения требований (картина вырисовывается поэтапно,сложно спрогнозировать общую картину) |
| **Спиральная** модель подходит компаниям которым необходимо уделить особое внимание оценке рисков | 1. Особое внимание отводится оценке рисков( Это позволяет сократить их к минимуму либо быть готовыми к некоторым непреодолимым) 2. В приоритет берется реализация с высокой степенью рисков (позволяет остановить на ранних циклах модели работу над проектом и уменьшить расходы) | 1. Могут сдвигаться сроки (Заказчик может пожелать изменение версии и это естественно приведет к изменению сроков) 2. Сложно спрогнозировать критерии на следующие итерации( поскольку каждая итерация не закладывает в себя гарантию что не будут внесены изменения, спрогнозировать следующую итерацию практически невозможно 3. Большое количество промежуточных циклов ( Влечет за собой обработку дополнительной документации) |
| **Скрам** Подходит компаниям которым необходимо внести изменения, так сказать обновиться. | 1 Команда работает этапами( За счет того что команда на каждом этапе определяет цели и пути, значительно ускоряет процесс работы  2.Команда работает на нескольколькими задачами одновременно( также позволяет значительно экономить время.  3 Каждый член команды четко осознает свою задач( повышается уровень ответственности к работе) | 1. не под все сферы деятельности адаптируется(есть компании реализация задач которых требует четкого планового подхода) 2. Необходима регулярная обратная связь от заказчика(процесс может тормозиться из-за того что заказчик не всегда может дать обратную связь) |
| **Канбан-** подходит компании которой необходимо решить не масштабные задачи. | 1. Четкий баланс распределения обязанностей( не получится что некоторые сотрудники перегружены а некоторые гуляют) 2. Команда работает на одной задачей( дает возможность сконцентрироваться на текущем процессе) 3. Наглядность продвижения задач ( из-за того что движение задач видно на доске всем членам команды, можно видеть как движется процесс и на каком этапе возникли сложности) | 1. Нет точных сроков( зачастую конечная дата все же намечается, но нет спринтов, что не позволяет поставить временные рамки по каждой итерации) 2. Не подходит для долгосрочного планирования ( работа выстраивается на решение актуальных задач при этом их приоритетность может меняться в зависимости от обстоятельств) |

**Другий рівень   
  
На твою думку, чому з’явився Agile-маніфест   
Agile** появился дабы добавить процессу разработки гибкости. Если обратить внимание на метод **Waterfal** то становится понятно, что не всегда четкое соблюдение пошагового прохождения реализации задач подходит разным компаниям, метод Agile дает нам возможность ускорить процесс реализации, сделать его более гибким за счет регулярных встреч, коммуникации между командой, обнаружением сложности на самой ранней стадии. Добавить четкой мотивации сотрудникам и осознание что нар решением задачи стоит он и его команда и каждый приносит свою пользу, никто не говорит что делать этап за этап, члены команды сами знают за какой сегмент они отвечают. По сути этот метод ориентирован на сплочение команды для реализации поставленной задачи, то есть вся команда плывет в одной лодке и дополняет действия своих коллег. Немаловажным является способность не боятся чего то неизведанного, стоит попробовать и получить обратную связь после чего отредактировать в соответствии.

**Третій рівень**

Как разработчик стартапа я выбираю методологию Скрам, потому что на первом этапе мыы с командой сделаем интерфейс позволяющий просто делиться со своими контактами картинками котиков и выпустим ее на рынок в пользование, за тем мы начнем добавлять всякие плюшки, такие как возможность формировать группы контактов по интересам, добавим возможность вести блог, объединять породы котиков в разделы, возможность редактирования или подписи фотографии котика. КАждая задача будет закрываться друг за другом давая пользователям возможность уже сейчас пользоваться ее и не ждать год или 2 пока мы выгрузим все 10 возможностей.