|  |  |
| --- | --- |
| Agile | |
| Плюсы | Минусы |
| Проект разделен на короткие и прозрачные итерации. | Невозможно подсчитать точную сумму работы из-за постоянно меняющихся требований. |
| Минимизация рисков благодаря гибкому процессу внесения изменений. | Новые требования могут противоречить существующей архитектуре или функционалу. |
| Работающая часть продукта после первой итерации | Со всеми корректировками и изменениями есть риск, что проект не закончится никогда. |
| Возможность кардинально изменить проект в процессе разработки. |  |
| Вся работа будет проводиться в небольших самоорганизующихся командах, короткими циклами и с фокусом на создание ценности для клиента. |  |
| Тестирование после каждой итерации |  |

Microsoft начинала с небольших agile-команд. Компания продолжала экспериментировать в немного ускоряющемся темпе в период 2008 – 2014. Это был период «раннего agile». Только после того, как Сатья Наделла занял пост CEO Microsoft, agile-подход начал распространяться по всей компании. В этот период Microsoft могла служить примером компании, применяющей идеи бизнес-гибкости.

Amazon сфокусировалась на максимизации ценности для клиента с момента выхода на фондовый рынок в 1997. В этот момент компанию тоже можно было назвать применяющей «ранний agile». Всего пять лет спустя компания включила в свои практики «2pizza-team» (количество участников в команде должно быть таким, чтобы их можно было накормить двумя пиццами) и начала связывать команды в сеть, отказываясь от вертикальной иерархии.

|  |  |
| --- | --- |
| Waterfall | |
| Плюсы | Минусы |
| Понятная и простая структура процессов, удобная для команд с небольшим опытом. | Требования определяются и закрепляются в самом начале, что лишает процесс разработки гибкости. |
| Качественная и детальная документация. | На проект тратится большее количество времени и ресурсов. |
| В проекте можно легко отследить ресурсы, риски и затраченное время. | Отсутствие обратных связей между этапами разработки. |
| Задачи продукта стабильны от начала до конца. | Клиент не участвует в создании продукта после постановки ТЗ. |
| Довольно прост в управлении. | Тестирование в конце |
|  | Изменения могут быть чрезвычайно дорогостоящими или даже невозможными. |

Модель Waterfall использовалась для разработки корпоративных приложений, таких как системы управления взаимоотношениями с клиентами (CRM), системы управления цепочками поставок, системы точек продаж (POS) для розничных сетей и т.п.

Модель Waterfall широко использовалась при разработке программного обеспечения до 2000 года. Даже после того, как в 2001 году был опубликован манифест Agile, модель Waterfall продолжала использоваться многими организациями до последнего десятилетия.

Тем не менее, есть некоторые области, в которых модель Waterfall по-прежнему предпочитается. Для военных разработок модель Waterfall может подойти лучше. Это связано со строгими стандартами и требованиями, которые необходимо соблюдать. В таких отраслях требования известны заранее, а контракты очень четко определяют конечный результат проекта.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3) Для собственного проекта могу попробовать заняться созданием примитивной социальной сети и чата.

Создать стартовую страничку с возможностью перехода на регистрацию или аутентификация.

Страничка пользователя с фотографией и списком друзей.

Так же добавить возможность перехода из странички пользователя на странички друзей, открытие списка переписок или перехода на страничку редактирования профиля.

Страничка редактирования профиля – собственно изменение имени, фамилии, регистрационных данных и фото.

Страничка друга – фотография и информация о человеке и возможность перехода или создания чата – переписку с другом.

Список переписок – простой список с именами тех, с кем вел переписку и возможностью перейти на саму страничку переписки.

Переписка – страничка с текстом переписки и возможностью перейти на страничку собеседника.

Можно использовать tomcat, spring, sqlite.

Сервер запустить на AWS.

Хочется иметь подобие спринтов по ходу которых мне, возможно, захочется добавить новый функционал или убрать ненужный и в итоге результат будет отличаться от изначального плана. На мой взгляд методология Agile подойдет лучше. Главное, что бы было время на это удовольствие.