**Домашнее задание к занятию «2.5. Жизненный цикл разработки ПО. Гибкие методологии разработки ПО»**

**Задание**

Бабушка генерального директора нашего банка не очень хорошо разбирается в интернет-банкинге и постоянно там путается. Поэтому директор решил для всех пенсионных счетов сделать отдельную вкладку, в которой также можно будет переводить деньги между счетами и оплачивать коммунальные услуги. Так как видит бабушка не очень хорошо, то весь шрифт должен быть увеличен, а картинки — стать более чёткими.

Рассмотрите процесс внедрения новых возможностей в готовый продукт.

Какую именно методологию или модель, по вашему мнению, лучше всего выбрать для этой задачи и почему? Опишите, как можно контролировать качество этой фичи на всех этапах разработки и какие действия необходимо совершить тестировщиками до разработки, во время, и уже после. Обратите внимание, что мы ожидаем не только описание общих действий, но и перечисление того, на что стоит обратить внимание, например, если учесть целевую аудиторию нашего дополнения и его особенности.

Это домашнее задание можно выполнить и в текстовом виде, и в виде таблицы: вы можете выбрать тот формат, который вам ближе и удобнее.

**Ответ**

Для этой задачи лучше всего подойдёт **ИНКРЕМЕНТНАЯ модель.**

Инкрементная модель — это метод, в котором проект проектируется, реализуется и тестируется инкрементно (то есть каждый раз с небольшими добавлениями) до самого окончания разработки.

**Аргументы:**

Так как основной функционал уже готов, и мы добавляем туда дополнительные функции в виде дополнения новой вкладки для пенсионеров, которое облегчит им пользование приложением.

Инкрементная разработка выполняется в несколько этапов: **проектирование, составление ТЗ, разработка, тестирование, внедрение, сопровождение и анализ**. И каждый последующий выпуск системы добавляет функцию к предыдущему выпуску, пока не будут реализованы все разработанные функции.

**Процесс тестирования**

В данной модели процесс тестирования внедряется на каждом этапе создания проекта и продолжается после каждого добавления нового функционала.

* На этапе **проектирования, планирования** тестировщик определяет приоритеты и сценарии. важные для бизнеса и клиента. Определяет, какие из заложенных функций своей некорректной работой могут повысить риски и в какой мере. Например это удобство пользования интерфейсом приложения людям с нарушением зрения и когнитивных способностей из- за пожилого возраста.
* На основании **ТЗ** определяет основной функционал дополнения пишет тест кейсы под проект, составляет чек-листы
* На этапе **разработки** происходит непосредственно тестирование проекта. Тестировщик проходит по всем тест кейсам и чек-листам.
* На этапе внедрения тестировщик отслеживает все проблемы которые возникли у пользователей для немедленного исправления.
* На этапе **сопровождения и анализа**, тестировщик работает над выявлением пропущенных дефектов, причин их пропуска и возможностей улучшения процесса. Например: если дефект был пропущен из-за недостатка требований, то следует внести уточнение в конструктор технических заданий — какие требования необходимо прописать в следующий раз