# **2.5. «Жизненный цикл разработки ПО. Гибкие методологии разработки ПО»**

## Ссылка на задание: <https://github.com/netology-code/iqa-homeworks/tree/55e04ae48dcda2fbf4bb8e2086a10d1262a740dd/2.5>

## Какую именно методологию и/или модель выбрать и почему?

Я бы выбирала модель в зависимости от того, насколько точно генеральный директор видит конечный продукт.

Исходя из текста задания кажется, что планируется отдельный раздел **только** для пенсионеров/пожилых людей с **ограниченным функционалом**, и продукт планируется только под эту ЦА. Тогда я бы предложила использовать V-образную модель, где нужно в самом начале зафиксировать нужный функционал и протестировать требования, затем тестировать каждый этап разработки.

Другой момент, что пожилые люди — это обычно пользователи с нарушениями зрения, слуха, с проблемами мелкой моторики и пр. Эти особенности нужно учитывать при разработке, НО эти особенности могут относиться не только к пожилым (возрастным) людям. Молодым людям с подобными ограничениями может потребоваться разный функционал интернет-банка.

Например, я предполагаю, что пожилым людям не особо нужен функционал по кредитованию в приложении. В отдельной версии для пожилых людей этот функционал можно убрать.

Но он может потребоваться НЕ возрастным пользователям с нарушениями зрения, слуха и т.д.

Поэтому, если есть планы не просто сделать удобный раздел для пожилых пользователей, но и в целом (для людей с различными физ. ограничениями), я бы советовала смотреть в сторону обеспечения доступности всего приложения/сайта интернет-банка. Тут я бы выбрала итеративную модель. Каждая итерация — проработка одного текущего раздела.

## Как можно контролировать качество данной фичи на всех этапах разработки?

Если выбирать V-образную модель, тогда нужен контроль на каждом этапе. Особенное внимание нужно уделить этапу анализа требований (провести тестирование требований), проектированию (после провести тестирование на потенциальной ЦА) и кодированию (модульное тестирование, интеграционное тестирование, системное тестирование).

После провести тестирование после ввода в эксплуатацию.

## На что обратить внимание?

1. Проанализировать, какой функционал наиболее важен для данной целевой аудитории и выделить в отдельный раздел только его. Если функционала слишком много — пользователь может не найти нужную ему вещь, устанет искать.
2. Весь функционал должен быть доступен для незрячих/ плохо видящих пользователей:
   * 1. если это веб, тогда должна быть масштабируемая вёрстка
     2. проверить достаточно ли контрастен шрифт на фоне
     3. наличие возможности работы с помощью клавиатуры
     4. проверить работу с экранным диктором
3. Нужно подумать, как сообщить пользователям о новом разделе, как сделать его заметным для пожилых людей?
4. Нужно обязательно проверять работу функционала с экранным диктором. программы экранного доступа могут «не видеть» определенные блоки информации на экране и не озвучивать их, неправильно называть элемент (например, «кнопка» вместо «ссылка»), называть те элементы, которые на экране не отображаются и т.п.
5. Также важно провести работу с текстами — нужны понятные формулировки