**Методы проектирования и разработки приложения “** **Passing”**

**Метод проектирования**

В качестве метода проектирования приложения была выбрана архитектура «сверху вниз».

При использовании подхода сверху-вниз изначальная задача делится на части, которые каждая в свою очередь может состоять из подзадач. Таким образом, делятся полученные задачи на ещё меньшие подзадачи до тех пор, пока основная задача не будет состоять из простых и однозначно понятных подзадач. Недостатком такого подхода является то, что в случае больших программных решений его нецелесообразно использовать, поскольку основная задача в таком случае делится на огромное количество подзадач, в случае чего возникают ситуации, когда одна и та же проблема решается несколько раз (потому что некоторые подзадачи требуют решения похожих проблем). В дополнение к этому инженер ПО должен довольно рано начинать думать над конкретными алгоритмами, при помощи которых поставленные задачи могут быть решены.

Плюсы и минусы использования метода проектирования «сверху вниз»

Плюсы:

* Организация целенаправленно внедряет решение для ресурсов, связанных с отдельным управляемым приложением.
* Первая стадия реализации сразу демонстрирует весь спектр возможностей решения по управлению идентификационной информацией.
* По завершении всех этапов реализации для управляемого приложения вы получите более глубокое и зрелое решение по управлению идентификационной информацией.
* На начальных этапах этот метод не так сильно затрагивает рабочие операции и технический персонал, как в случае реализации "снизу-вверх".

Минусы:

* На первых этапах решение охватывает только ограниченную область.
* На первых этапах управление охватывает небольшой процент учетных записей пользователей.
* Возможно, вам придется разработать пользовательские адаптеры на ранних этапах.
* Служба поддержки и предприятие в целом не смогут быстро ощутить преимущества решения.
* Вероятно, стоимость реализации окажется выше.

На основании всех перечисленных плюсов и незначительных для нашего проекта минусов, мы решили выбрать этот метод.

**Метод разработки**

В качестве метода разработки приложения была выбрана модель жизненного цикла «итеративная» или «инкрементальная».

Итеративная или инкрементальная модель предполагает разбиение создаваемой системы на набор кусков, которые разрабатываются с помощью нескольких последовательных проходов всех работ или их части.

На первой итерации разрабатывается фрагмент системы, не зависящий от других. При этом большая часть или даже полный цикл работ осуществляются на нем, затем оцениваются результаты, и на следующей итерации либо первый фрагмент переделывается, либо разрабатывается следующий, который может зависеть от первого, либо как-то совмещается доработка первого фрагмента с добавлением новых функций. В результате на каждой итерации можно анализировать промежуточные результаты работ и реакцию на них всех заинтересованных лиц, включая пользователей, и вносить корректирующие изменения на следующих итерациях. Каждая итерация может содержать полный набор видов деятельности от анализа требований до ввода в эксплуатацию очередной части ПО.

Итеративный процесс предполагает, что разные виды деятельности не привязаны намертво к определенным этапам разработки, а выполняются по мере необходимости, иногда повторяются до тех пор, пока не будет получен нужный результат.

Плюсы и минусы использования итеративного метода разработки

Плюсы:

* Снижение рисков — раннее обнаружение конфликтов между требованиями, моделями и реализацией проекта; большая фокусировка на основных задачах; динамическое формирование требований и управление ими.
* Организация эффективной обратной связи проектной команды с потребителем, создание продукта, реально отвечающего его потребностям.
* Быстрый выпуск минимально ценного продукта (MVP) и возможность вывести продукт на рынок и начать эксплуатацию гораздо раньше.

Минусы:

* Проблемы с архитектурой и накладные расходы — при работе с хаотичными требованиями и без проработанного глобального плана архитектура приложения может пострадать, а на её приведение к адекватному виду могут потребоваться дополнительные ресурсы. По сути, за возможность менять требования в ходе создания продукта, приходится так или иначе расплачиваться.
* Нет фиксированного бюджета и сроков, а также нужна сильная вовлеченность заказчика в процесс — для некоторых заказчиков это неприемлемые условия сотрудничества с разработчиком, им лучше подойдёт водопадная модель.

При выборе метода разработки, наиболее подходящего для нашего проекта, мы опирались на следующие пункты:

* Эффективная обратная связь с игроками (потребителями).
* Динамическое формирование требований и управление ими.
* Создание продукта, отвечающего ожиданиям игрока.
* Быстрый выпуск демоверсии.
* Возможность вывести продукт на рынок как можно раньше.