**Методы проектирования и разработки мобильного приложения «Бетонный агрегатор»**

В качестве метода проектирования и разработки программы, взаимодействующий с базой данных, которая хранит информацию на сервере, была выбрана архитектура “клиент-сервер”.

В ООО “Кодекс” данную архитектуру образует сервер и приложение, подключающееся к нему через API. Приложения формируют запросы к БД, а сервер занимается их обработкой и ответами.

На сервере сети размещается БД и устанавливается серверная СУБД (программный компонент, обеспечивающий хранение информации, ее обработку и представление ее пользователям в сетевом режиме).

На смартфоне через приложение формируется запрос к БД. Серверная СУБД обеспечивает интерпретацию запроса, его выполнение, формирование результата запроса и пересылку его по сети обратно на смартфон (клиент). Клиентское приложение интерпретирует его необходимым образом и представляет пользователю. Клиентское приложение может также посылать запрос на обновление БД и серверная СУБД внесет необходимые изменения в БД.



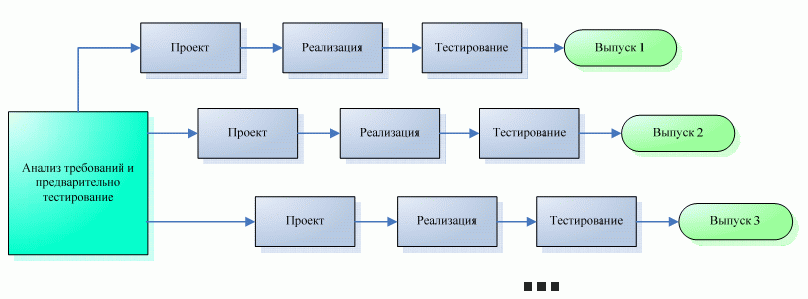
*Рисунок 1. Архитектура “клиент-сервер”*

**Причины выбора архитектуры клиент/сервер:**

* При обработке данных по архитектуре клиент/сервер уменьшается сетевой трафик, так как через сеть передаются только результаты запросов.
* Груз файловых операций ложится на сервер, который мощнее смартфонов (клиентов) и поэтому способен быстрее обслуживать запросы. В связи с этим уменьшается потребность клиентских приложений в оперативной памяти.
* Поскольку данные хранятся на сервере, то на смартфонах (клиентах) освобождается значительный объем дискового пространства.
* Повышается уровень непротиворечивости данных и существенно повышается степень безопасности БД, так как правила целостности данных определяются в серверной СУБД и являются едиными для всех приложений, использующих эту БД.

**Метод проектирования**

В качестве метода разработки приложения была выбрана инкрементная модель разработки, потому что в ходе выполнения проекта и после его завершения может потребоваться расширить возможности ПО, при этом чтобы не затрагивались главные функции.



*Рисунок 2. Инкрементная модель*