**Методы проектирования и разработки приложения.**

**Метод проектирования**

Сутью структурного программирования является возможность разбиения программы на составляющие элементы.

Распространены две методики (стратегии) разработки программ, относящиеся к структурному программированию: программирование "сверху вниз" и программирование "снизу-вверх".

Программирование "сверху вниз", или нисходящее программирование **–**это методика разработки программ, при которой разработка начинается с определения целей решения проблемы, после чего идет последовательная детализация, заканчивающаяся детальной программой. Является противоположной методике программирования «снизу-вверх».

При нисходящем проектировании задача анализируется с целью определения возможности разбиения ее на ряд подзадач. Затем каждая из полученных подзадач также анализируется для возможного разбиения на подзадачи. Процесс заканчивается, когда подзадачу невозможно или нецелесообразно далее разбивать на подзадачи.

В данном случае программа конструируется иерархически - сверху вниз: от главной программы к подпрограммам самого нижнего уровня, причем на каждом уровне используются только простые последовательности инструкций, циклы и условные разветвления.

Достоинства структурного программирования:

1) повышается надежность программ (благодаря хорошему структурированию при проектировании, программа легко поддается тестированию и не создает проблем при отладке);

2) повышается эффективность программ (структурирование программы позволяет легко находить и корректировать ошибки, а отдельные подпрограммы можно переделывать (модифицировать) независимо от других);

3) уменьшается время и стоимость программной разработки;

4) улучшается читабельность программ.

**Метод разработки**

В качестве метода разработки приложения была выбрана итеративная модель.

Итеративная разработка — это процесс создания программного обеспечения, который осуществляется небольшими этапами, в ходе которых ведется анализ полученных промежуточных результатов, выдвигаются новые требования и корректируются предыдущие этапы работы.

Жизненный цикл проекта при итерационной разработке разбит на последовательность итераций, каждая из которых, по сути, является проектом в миниатюре, то есть включает в себя все процессы разработки ПО (сбор и анализ требований, составление спецификаций, непосредственную реализацию, тестирование и запуск), но в рамках одной итерации разрабатывается не весь проект, а только его версия или отдельная часть.

Как правило, цель каждой итерации — это получение версии ПО, включающей в себя как новые или проработанные возможности, реализованные в ходе текущей итерации, так и функциональность всех предыдущих итераций. Результат же финальной итерации содержит всю требуемую функциональность продукта.

Основные преимущества итеративной модели разработки:

* Риски снижаются ранее, поскольку интеграция элементов выполняется постепенно.
* Поддерживается изменение требований и тактики.
* Облегчается улучшение и отладка продукта, что позволяет получить более надежный продукт.
* Возможно обучение организаций в этом подходе и улучшение их процессов.
* Увеличивается возможность повторного использования.

Основные недостатки итеративной модели разработки:

* 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.
* Нет фиксированного бюджета и сроков, а также нужна сильная вовлеченность заказчика в процесс.