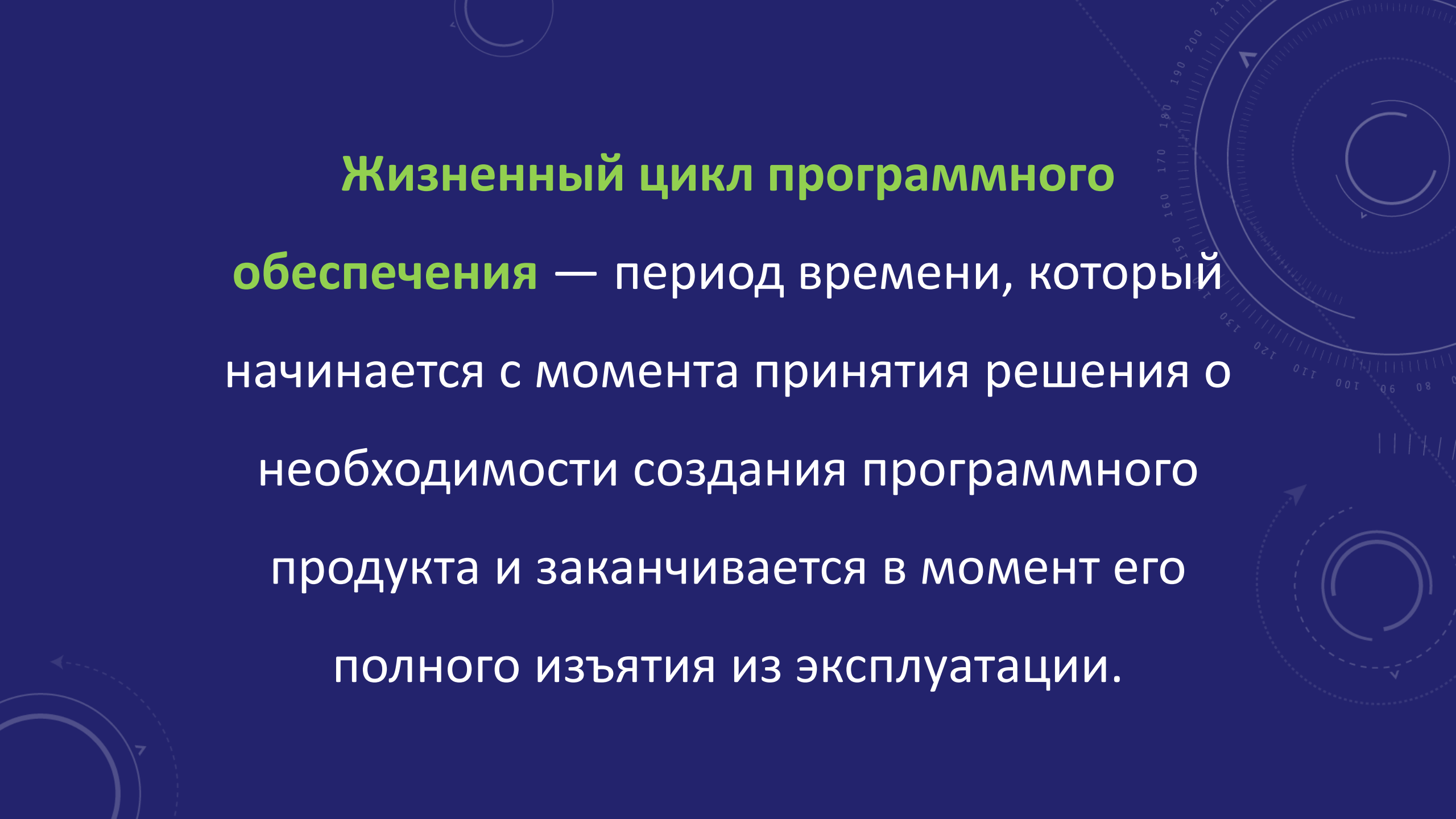
The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several faint, light blue circular elements. On the left side, there is a large circular scale with tick marks and numbers ranging from 140 to 260. Other smaller circles and curved lines are scattered across the background, some with arrows indicating a clockwise direction.

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Выполнил студент гр.20П-3

Хизбуллин А.Н.

The background is a solid dark blue. It features several faint, light blue circular patterns. Some are solid lines, while others are dashed. Some circles have arrows indicating a clockwise direction. There are also some concentric circles with tick marks, resembling a gauge or a clock face. The text is centered and reads:

Жизненный цикл программного обеспечения — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

ОСНОВНЫЕ ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА ПО

- анализ требований
- проектирование
- кодирование (программирование)
- тестирование и отладка
- эксплуатация и сопровождение

Модель жизненного цикла ПО — структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

Наиболее распространенными моделями являются:

- **каскадная**
- **инкрементная** (поэтапная модель с промежуточным контролем)
- **спиральная**

КАСКАДНАЯ МОДЕЛЬ



КАСКАДНАЯ МОДЕЛЬ

Предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.

Преимущества:

- Последовательное выполнение этапов проекта в строгом фиксированном порядке
- Позволяет оценивать качество продукта на каждом этапе

Недостатки:

- Отсутствие обратных связей между этапами
- Не соответствует реальным условиям разработки программного продукта

ИНКРЕМЕНТНАЯ МОДЕЛЬ



ИНКРЕМЕНТНАЯ МОДЕЛЬ

Продукт проектируется, реализуется и тестируется инкрементно (то есть каждый раз с небольшими добавлениями) до самого окончания разработки.

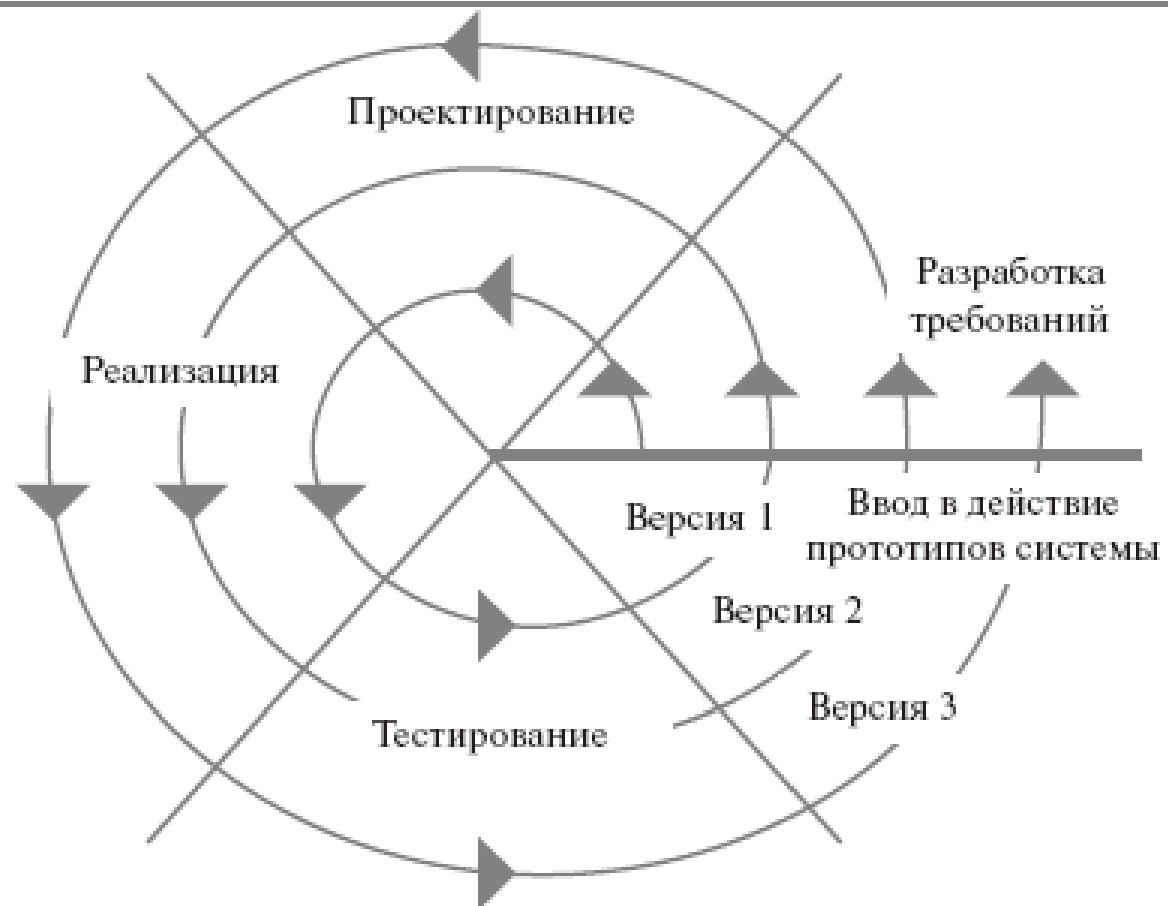
Преимущества:

- Рабочее приложение выходит на ранней стадии жизненного цикла продукта
- Гибкость. Изменить масштабы и требования проекта относительно менее затратно
- Небольшие итерации упрощают тестирование и внесение правок
- Проще идентифицировать риски, справиться с ними
- Каждая итерация — простая в управлении контрольная точка проекта

Недостатки:

- Каждая фаза итерации неподвижна
- Могут возникнуть проблемы относительно архитектуры системы, так как не все требования собраны заранее для всего жизненного цикла ПО

СПИРАЛЬНАЯ МОДЕЛЬ



СПИРАЛЬНАЯ МОДЕЛЬ

На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка.

Преимущества:

- Мониторинг рисков
- Заказчик может увидеть работающую версию продукта уже на ранних этапах
- Изменения могут быть внесены на поздних стадиях разработки
- Есть возможность гибкого проектирования

Недостатки:

- Мониторинг рисков требует дополнительных ресурсов
- Большое количество промежуточных стадий разработки, большой объем документации
- На самых ранних стадиях дата завершения работы над проектом может быть неизвестна

ДОПОЛНИТЕЛЬНЫЕ МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА

Несмотря на множество исследований, мнение об эффективности методик, принципов и методологий часто основывается на личном опыте, эмоциональном отклике и компетенциях менеджера, который их применял. И не всегда понравившаяся из описания модель будет наилучшей для реализации именно вашего проекта. Поэтому рассмотрим ещё несколько моделей жизненного цикла ПО:

- **RAD model**
- **V-model**
- **Agile model**
- **Iterative model**

RAD MODEL



RAD MODEL

RAD (от англ. rapid application development — быстрая разработка приложений) — концепция, ориентированная на максимально быстрое получение результата в условиях сильных ограничений по срокам и бюджету и нечётко определённых требований к продукту. Эффект ускорения разработки достигается путём использования соответствующих технических средств и непрерывного, параллельного с ходом разработки, уточнения требований и оценки текущих результатов с привлечением заказчика.

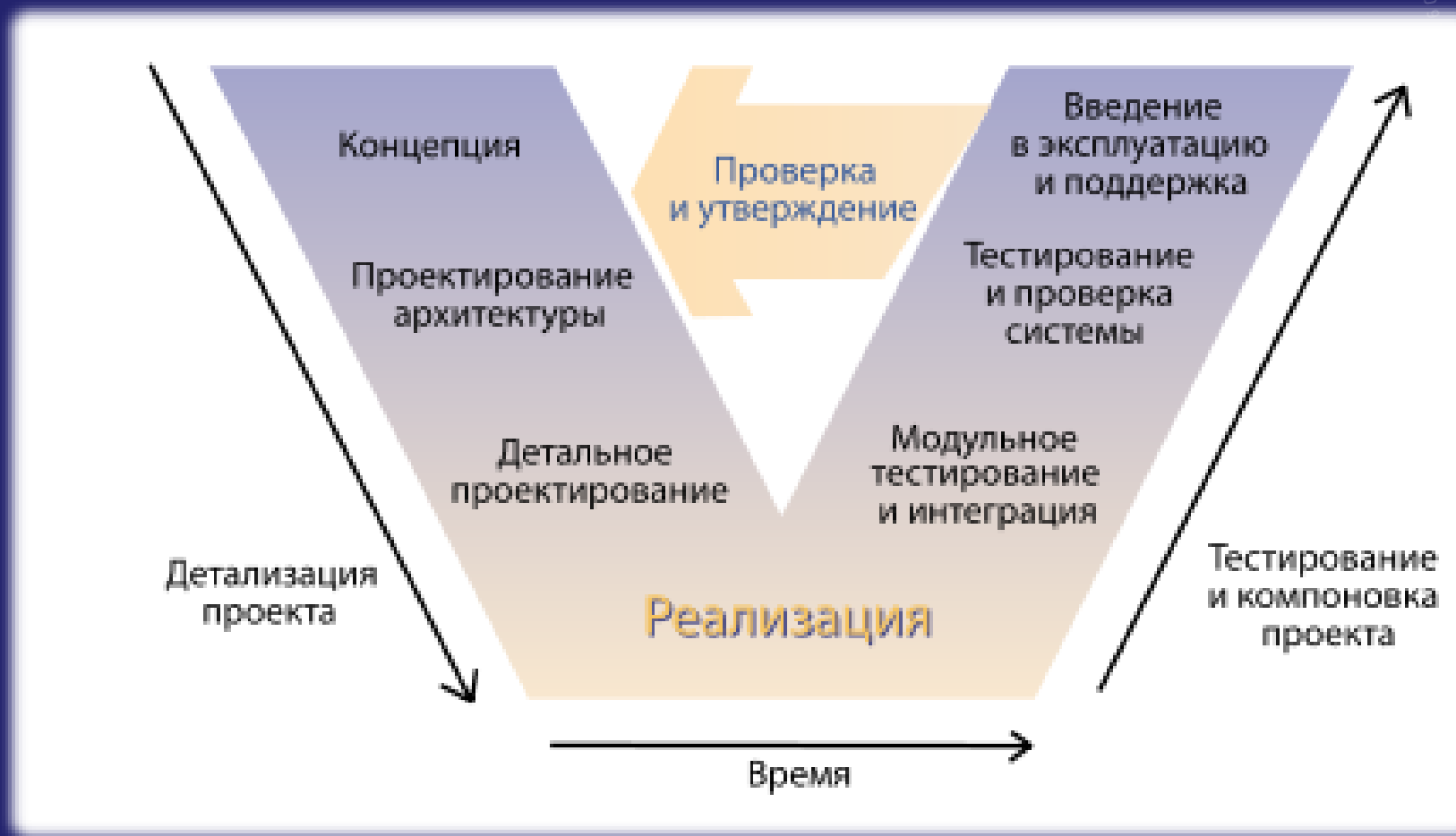
Преимущества:

- быстрота продвижения программного продукта на рынок
- интерфейс, устраивающий пользователя
- лёгкая адаптируемость проекта к изменяющимся требованиям
- простота развития функциональности системы

Недостатки:

- Требуется много усилий для сбора всех требований на начальном этапе
- Навыки моделирования имеют много зависимостей
- Не подходит для малобюджетного проекта

V-MODEL



V-MODEL

V-Model – детализация проекта возрастает при движении слева направо, одновременно с течением времени, и ни то, ни другое не может повернуть вспять. Итерации в проекте производятся по горизонтали, между левой и правой сторонами буквы.

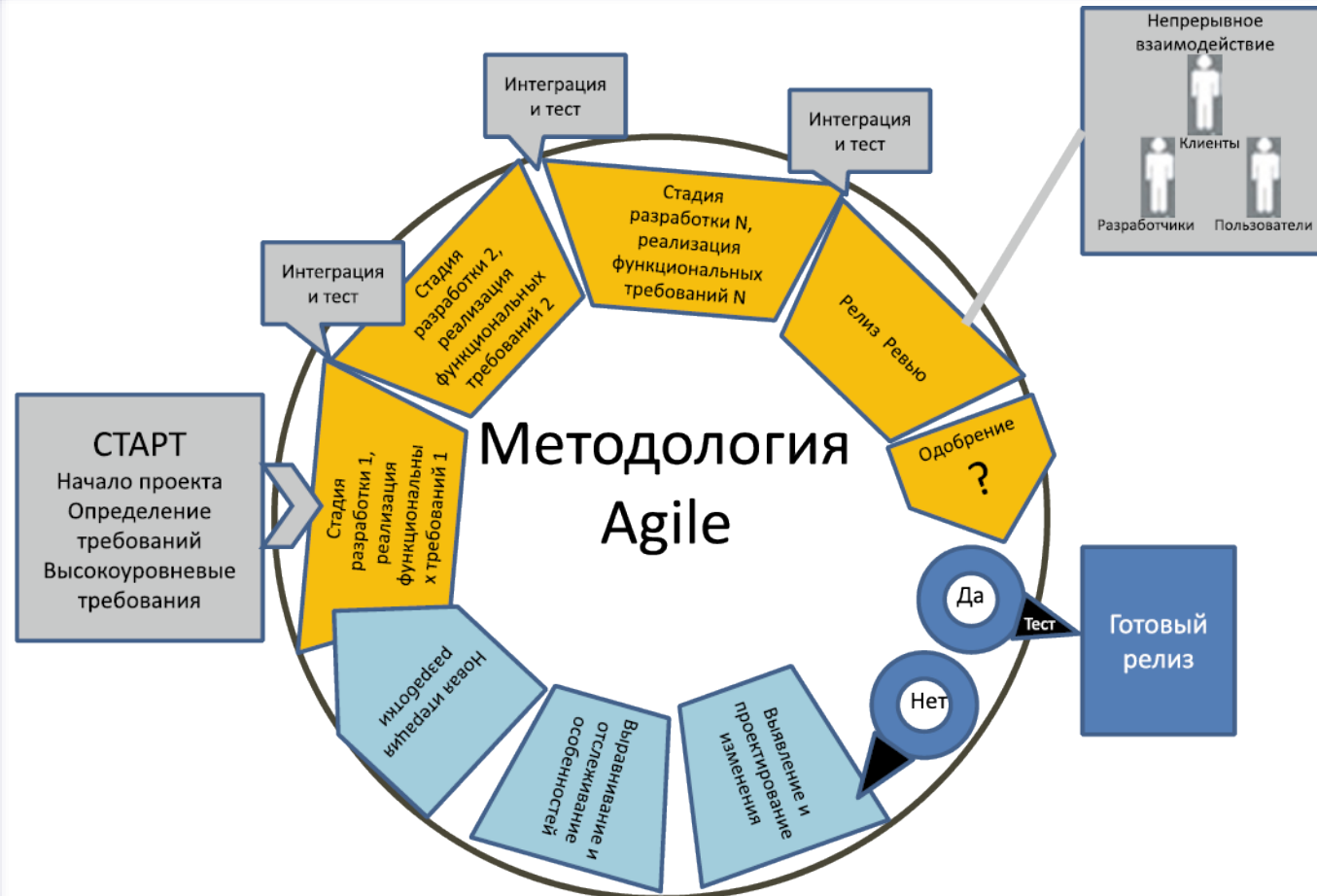
Преимущества:

- Количество ошибок в архитектуре ПО сводится к минимуму

Недостатки:

- Если при разработке архитектуры была допущена ошибка, то вернуться и исправить её будет стоить дорого, как и в «каскаде»

AGILE MODEL



AGILE MODEL

Agile (от англ. «гибкий») – набор принципов гибкой разработки (всего их 12) и идей. Основные идеи Agile:

- люди и взаимодействие важнее процессов и инструментов
- работающий продукт важнее исчерпывающей документации
- сотрудничество с заказчиком важнее согласования условий контракта
- готовность к изменениям важнее следования первоначальному плану

AGILE MODEL

Преимущества:

- тестирование на ранних стадиях
- возможность оценки добавленного функционала “в действии”
- исследование пользовательского опыта на всех этапах
- возможность быстрой презентации на рынке “сырой”, но работающей версии

Недостатки:

- отсутствие четкого плана развития проекта
- постоянная угроза переделывания большой части работы
- снижение качества продукта в угоду скорости и упрощения

ITERATIVE MODEL



ITERATIVE MODEL

Итеративная модель— это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой последующих этапов работы.

Преимущества:

- Быстрый выпуск минимального продукта даёт возможность оперативно получать обратную связь от заказчика и пользователей. А значит, фокусироваться на наиболее важных функциях ПО и улучшать их в соответствии с требованиями рынка и пожеланиями клиента
- Постоянное тестирование пользователями позволяет быстро обнаруживать и устранять ошибки

Недостатки:

- Использование на начальном этапе баз данных или серверов — первые сложно масштабировать, а вторые не выдерживают нагрузку. Возможно, придётся переписывать большую часть приложения
- Отсутствие фиксированного бюджета и сроков. Заказчик не знает, как выглядит конечная цель и когда закончится разработка