

Жизненный цикл разработки ПО

Ксения Шипина
Системный аналитик



Что изучим?

1

Что такое жизненный цикл разработки ПО?

2

Этапы жизненного цикла

3

Модели разработки ПО



Жизненный цикл разработки ПО

Определение жизненного цикла

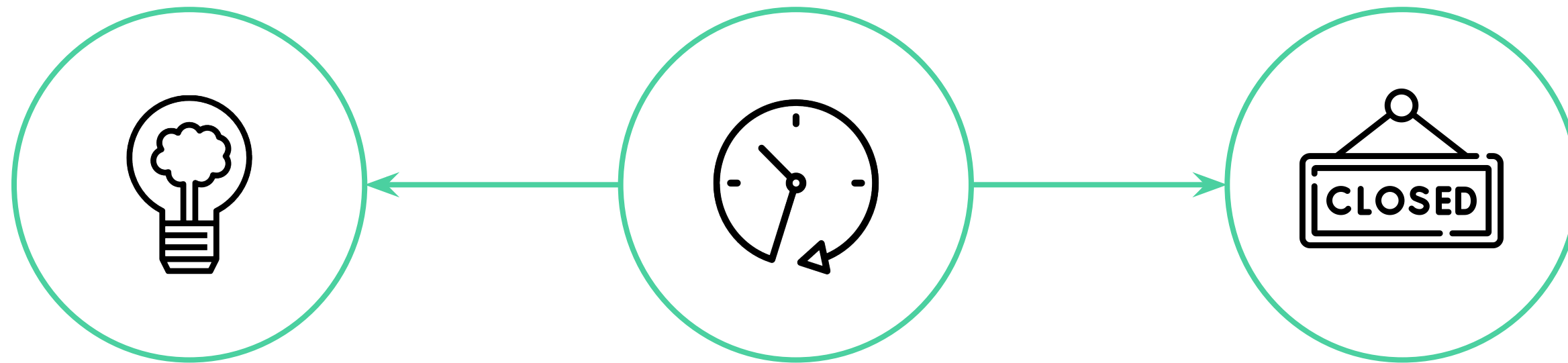
Когда продукт готов?

1



Жизненный цикл разработки ПО

Это **период времени** от момента **принятия решения о необходимости разработки** программного продукта до момента его **полного изъятия из эксплуатации**



Готовый продукт — мёртвый продукт

Из-за высокого уровня конкуренции, быстро меняющихся требований и потребностей заказчика продукты находятся **в постоянном развитии**

В концепции жизненного цикла завершение разработки продукта является завершением его **сопровождения** и **полным изъятием из эксплуатации** пользователями





Этапы жизненного цикла

На каких этапах важна роль системного аналитика?



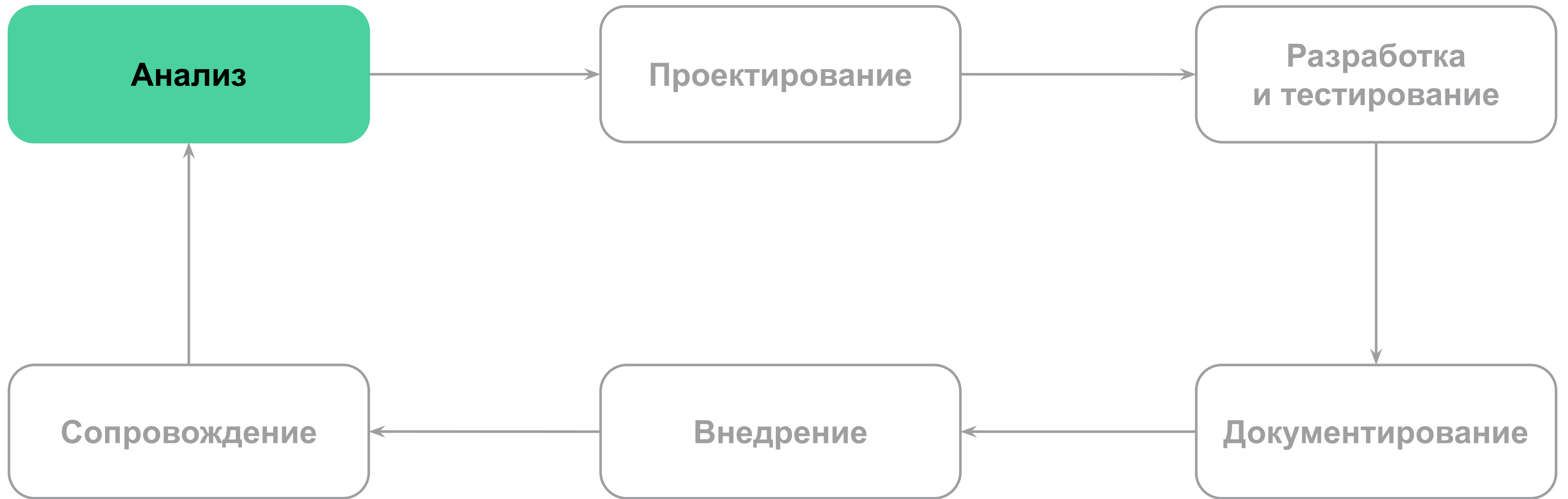
2



Этапы жизненного цикла разработки ПО



Этапы жизненного цикла разработки ПО



Анализ

Включает несколько шагов:

1. Формирование видения продукта,
его функциональности и границ ответственности
2. Моделирование бизнес-процессов
3. Анализ требований
4. Планирование



Анализ. Планирование

MVP (minimum viable product) — версия продукта, обладающая минимальным набором функций, несущих ценность для удовлетворения основных потребностей пользователей.

Основная задача создания **MVP** — быстрое получение обратной связи для формирования гипотез дальнейшего развития продукта.

Основные артефакты планирования:

- Цель проекта, задачи
- Фазы проекта и результат
- Трудозатраты для решения задач
- Проектный план

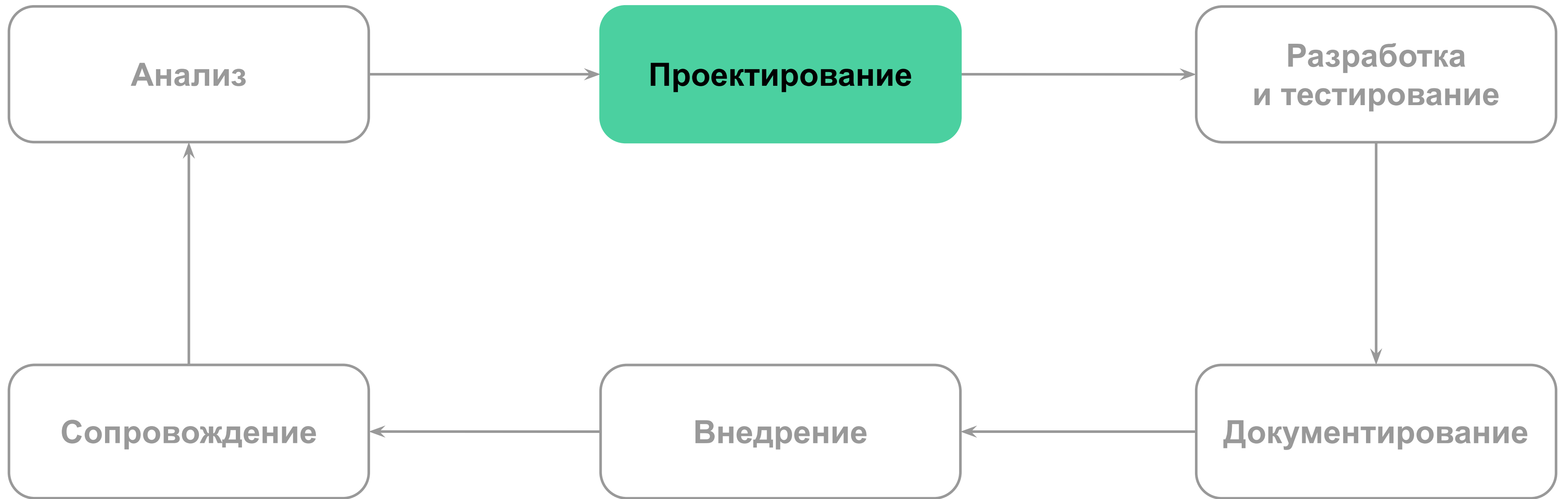


Анализ. Ответственные за этап

1. Формирование видения продукта — **заказчик, product owner**
2. Моделирование бизнес-процессов — **системный/бизнес-аналитик**
3. Анализ требований — **системный/бизнес-аналитик**
4. Планирование — **project manager**



Этапы жизненного цикла разработки ПО



Проектирование архитектуры

Архитектура — набор решений по организации структуры

кода и процесса разработки, призванных:

- ускорить/удешевить достижение бизнес-цели
- достичь приемлемой производительности/увеличить производительность
- сделать приложение легко расширяемым на случай, если это понадобится
- уменьшить количество ошибок, связанных с безопасностью хранения и доступа к данным, работой кода на разных наборах данных и т. д.



Проектирование пользовательского интерфейса

UI (user interface) — дизайн пользовательского интерфейса, отвечающий всем требованиям функциональности, выявленным на этапе анализа.

UX (user experience) — повышенное удобство и интуитивно понятный процесс взаимодействия пользователя с продуктом.

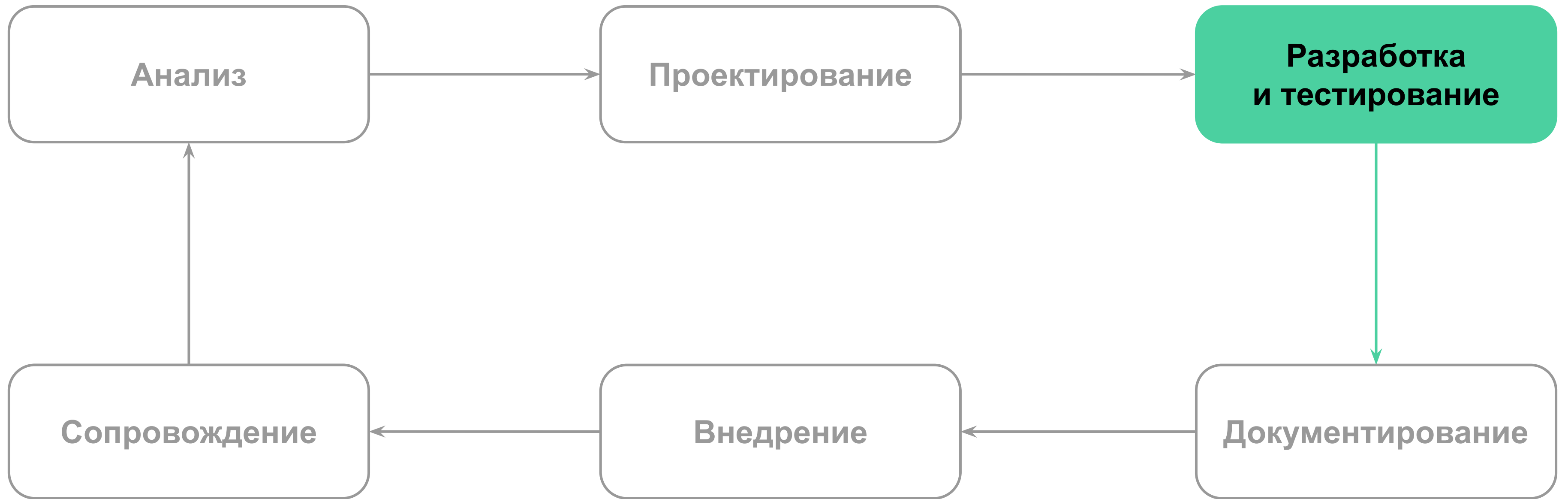


Проектирование. Ответственные за этап

1. Разработка архитектуры программного обеспечения — **архитектор**
2. Проектирование UI/UX — **дизайнер/UX-проектировщик**



Этапы жизненного цикла разработки ПО



Разработка и тестирование

Разработка — написание исходного кода программного продукта.

Тестирование — выявление ситуаций, в которых поведение системы является неправильным, нежелательным или не соответствующим спецификации.

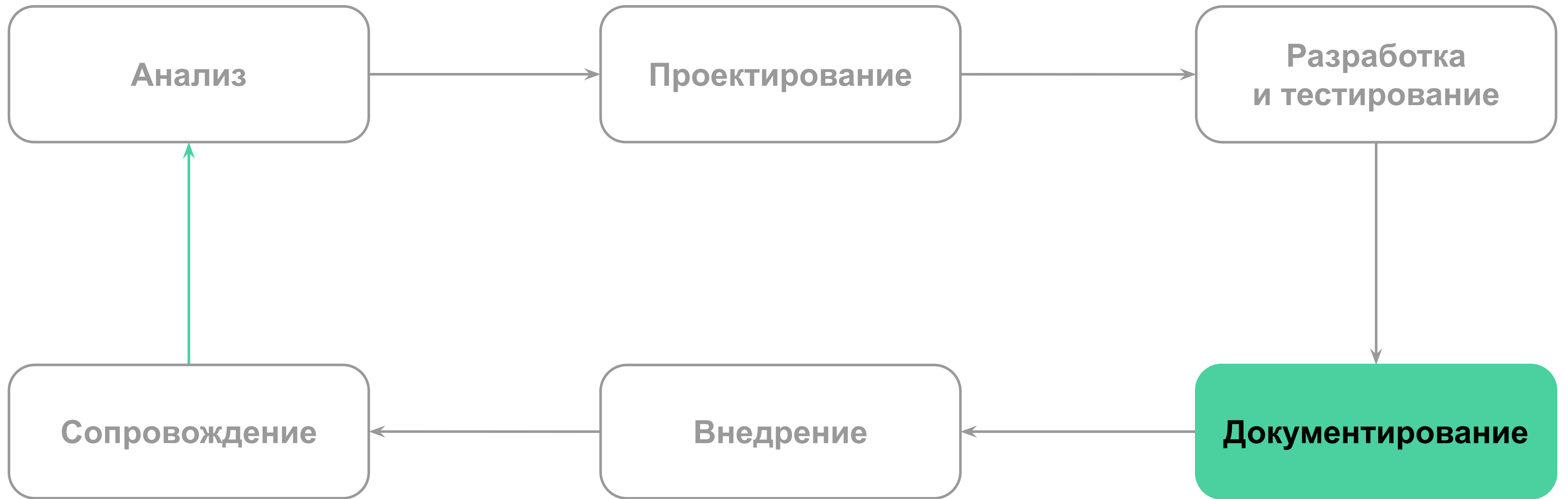


Разработка и тестирование. Ответственные за этап

1. Разработка — **team lead** команды разработки
2. Тестирование — **QA lead/команда** тестировщиков



Этапы жизненного цикла разработки ПО



Документирование

Процесс создания документации, которая необходима для релиза продукта:

- диалоговой (оперативной) документации
- руководства пользователя
- справочного текста, описывающего,
как пользоваться программным продуктом.



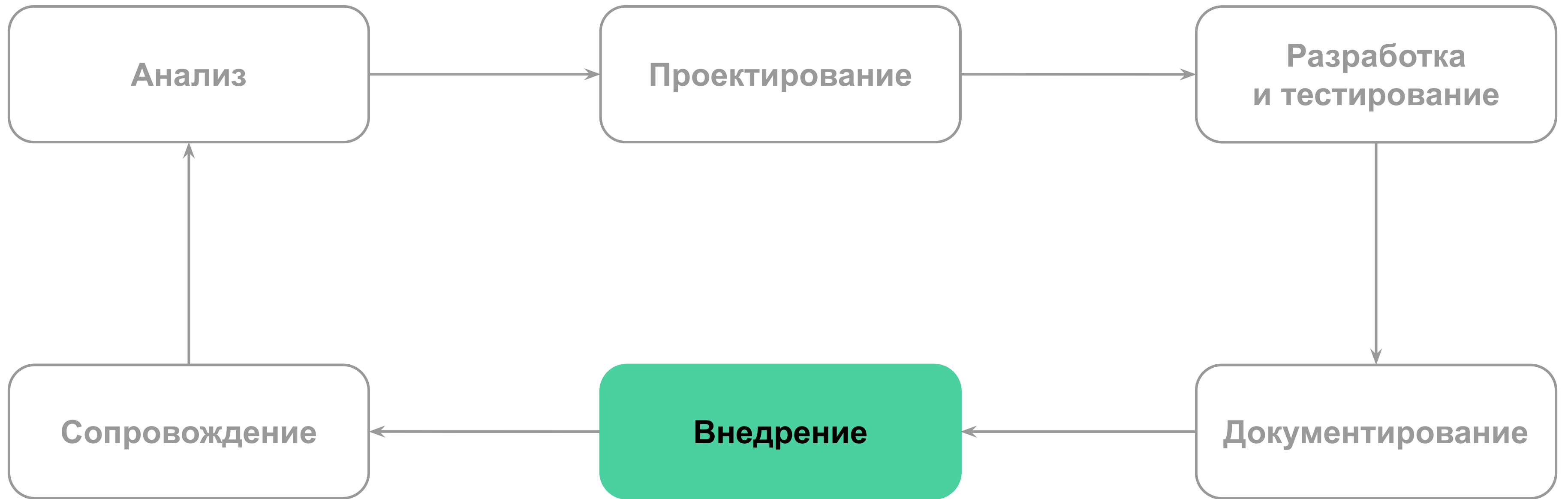
Документирование. Ответственные за этап

Зависит от культуры документирования в компании:

- аналитик
- технический писатель
- тестировщик



Этапы жизненного цикла разработки ПО



Внедрение

Внедрение — процесс настройки программного обеспечения под определённые условия использования, а также обучения пользователей работе с программным продуктом



Внедрение. Уровни

Технический — внедрение как процесс настройки оборудования: самый малозатратный этап, нужно только «железо» и квалифицированные люди.

Технологический — внедрение в технологические процессы: настройка под бизнес-процессы или выстраивание бизнес-процессов.

Организационный — «внедрение» в головы людей и корпоративную культуру.



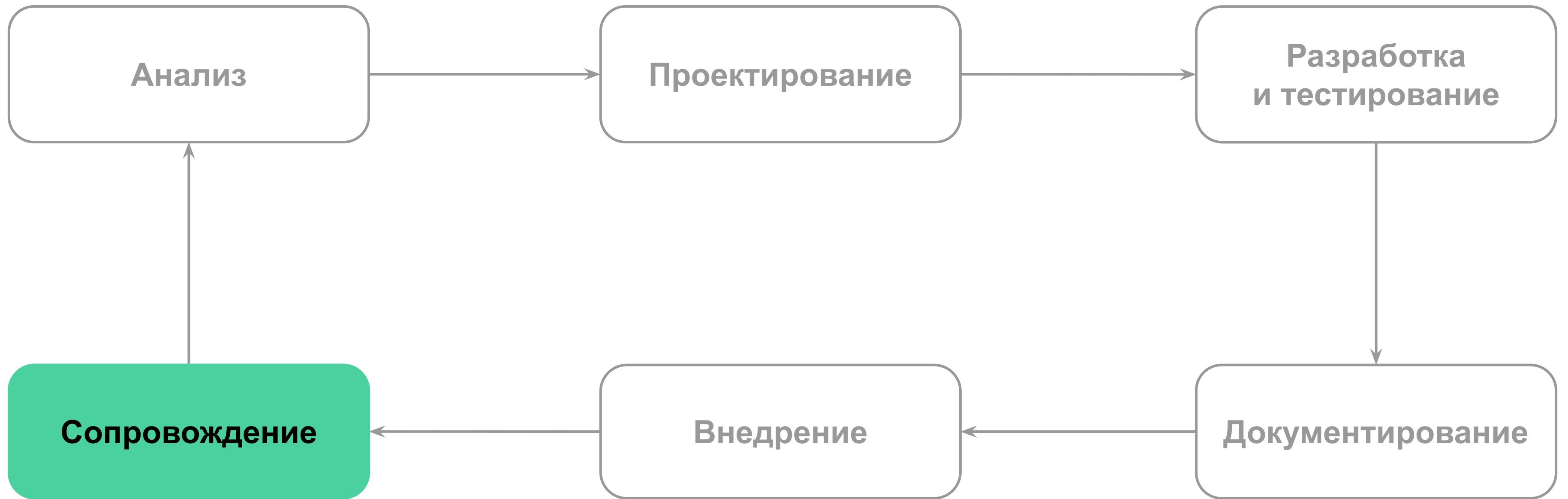
Внедрение. Ответственные за этап

Зависит от вида программного обеспечения:

- для крупных систем, требующих настройки под нужды организации —
специальная команда внедрения
- для онлайн-сервисов и мобильных приложений —
DevOps и команда разработки



Этапы жизненного цикла разработки ПО



Сопровождение

Сопровождение — процесс улучшения, оптимизации и устранения дефектов программного обеспечения после передачи в эксплуатацию



Сопровождение. Ответственные за этап

Зависит от масштаба компании-разработчика
и от договорённостей с заказчиком.

0 линия — обработка телефонных обращений от клиентов, передача сложных проблем техническим специалистам (1-я линия сопровождения).

1-я линия — консультация/настройка/устранение простых ошибок в работе ПО.

2-я линия — функциональное сопровождение, решение неизвестных ранее проблем.

3-я линия — системное сопровождение/проектная деятельность на этапе запуска ПО на оборудовании заказчика.



На каких этапах жизненного цикла аналитик принимает непосредственное участие?



Модели разработки ПО



3



Модель разработки программного обеспечения описывает:

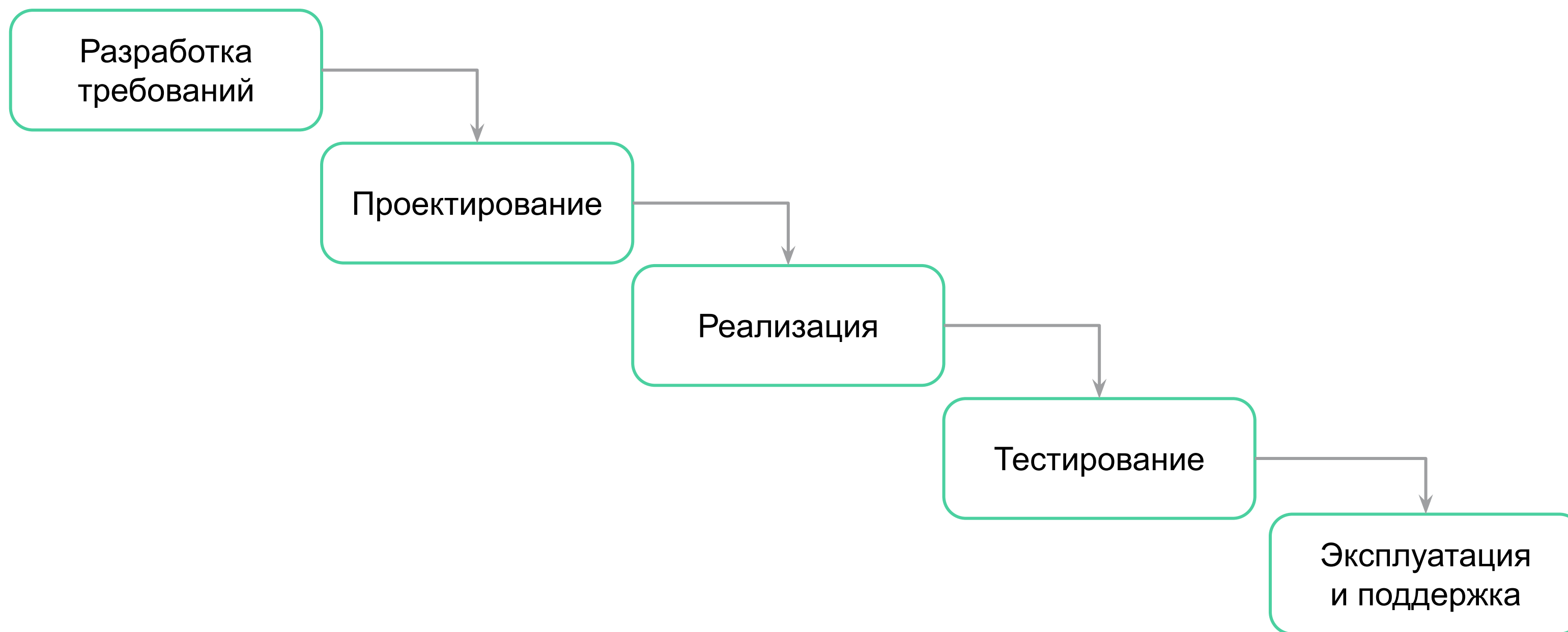
- какие **стадии жизненного цикла** должна проходить разработка
- какие **процессы** должны происходить на каждой стадии
- какие **артефакты** должны быть получены в результате их завершения



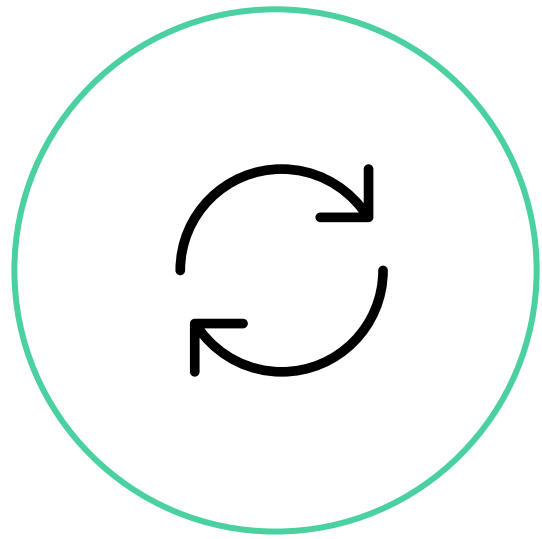
Водопадная модель (Waterfall)

Разработана в 1976 г.

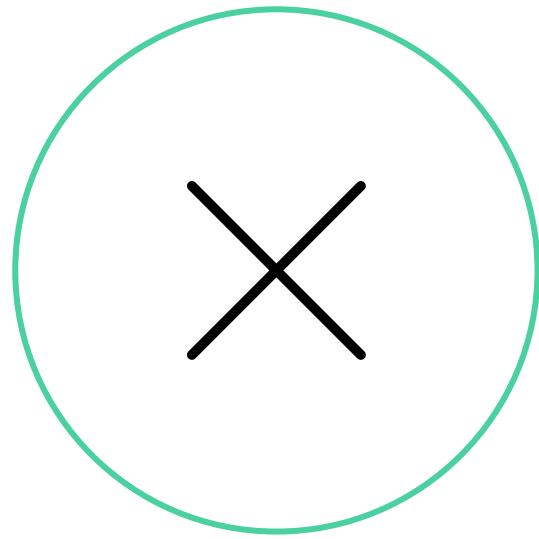
Представляет собой линейный процесс,
состоящий из следующих этапов:



Принципы Waterfall:



Следующий этап работы не начинается, пока не закончится предыдущий



Пропускать этапы нельзя



Каждый этап должен быть тщательно задокументирован



Водопадная модель (Waterfall)

Преимущества

- Стоимость проекта определяется на начальном этапе
- Легко контролировать процесс разработки

Недостатки

- При необходимости внесения каких-либо изменений в функциональность необходимо пройти весь процесс заново
- Код тестируется в самом конце, что приводит к большому числу исправлений в коде документации на финальных этапах проекта
- Велика вероятность получить неиспользованный продукт, т. к. заказчик может дать обратную связь только после завершения всех работ по проекту
- Разработчики пишут много технической документации, что увеличивает время и стоимость разработки

Когда стоит использовать подход?

- В относительно небольших проектах
- Когда требования известны, понятны и непротиворечивы
- При наличии заказчика, требующего прозрачность работ и исполнение в точные сроки
- При разработке проекта, не имеющего конкуренции на рынке



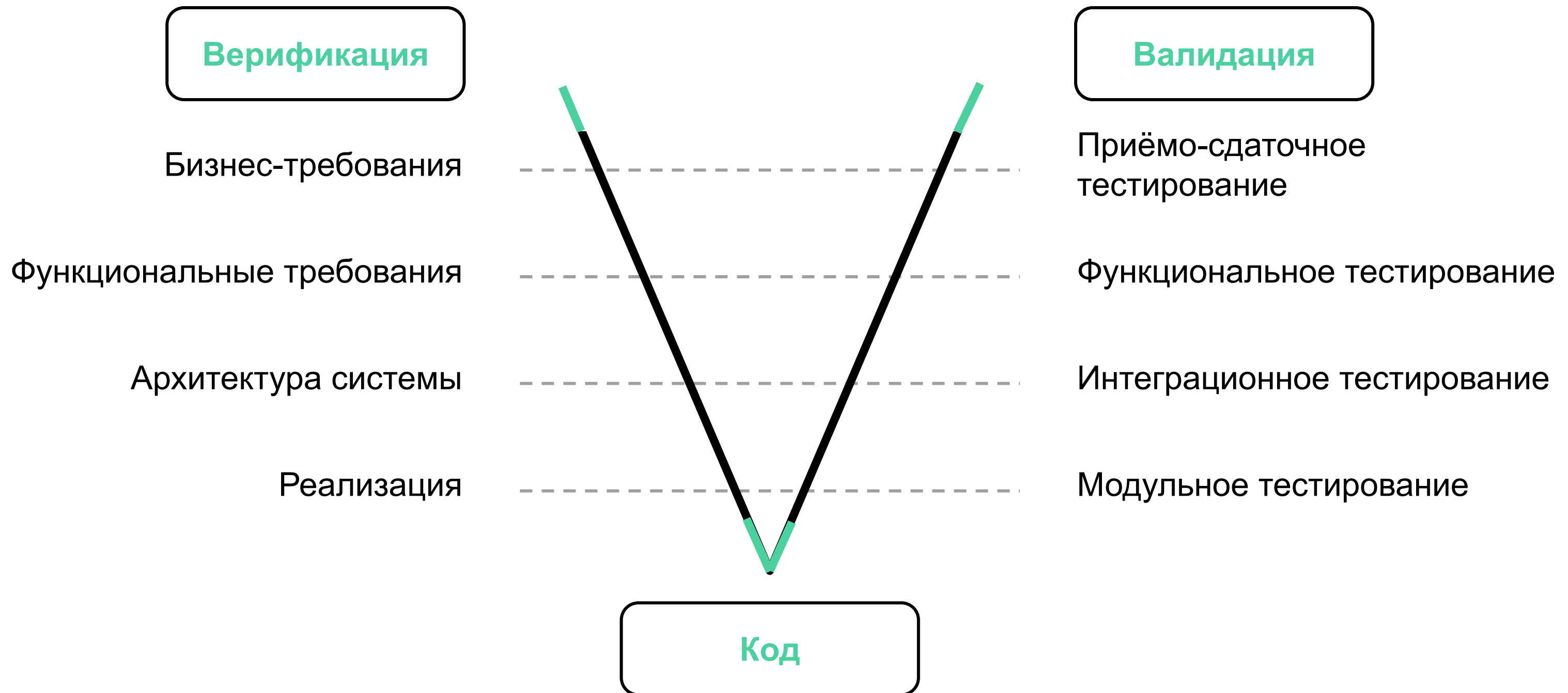
Особенности этапа анализа (Waterfall)

1. Выделенный этап анализа
2. Детальная спецификация, описанная универсальным абстрактным языком
3. Требования собраны в единый документ и описаны в терминах системы
4. Требования являются сильно связанными, что осложняет внесение изменений
5. Долгий этап согласования



V-модель

Была разработана в конце 1980-х годов, но окончательно модифицирована под разработку ПО только в 2005 г. Воплощает идею «validation and verification».



Цели V-модели:

минимизация рисков

гарантии качества

**уменьшение общей
стоимости проекта**



V-модель

Преимущества

- в модели особое значение придаётся планированию, направленному на верификацию каждого шага разработки продукта
- количество ошибок в архитектуре ПО сводится к минимуму

Недостатки

- внесение изменений на всех этапах жизненного цикла модели не предусмотрено
- тестирование требований в жизненном цикле происходит слишком поздно, вследствие чего невозможно внести изменения, не повлияв при этом на график разработки

Когда стоит использовать подход?

- В малых и средних проектах, где требования чётко определены
- В проектах, для которых важна надёжность, а цена ошибки очень высока
- В условиях доступности инженеров необходимой квалификации, особенно тестировщиков



Инкрементная модель

Процесс работы над проектом разделяется на несколько полных циклов разработки, в результате которых реализуется часть системы



Инкрементная модель

Преимущества

- Заказчик может оплатить только создание основных функций и принять решение о необходимости развития продукта после сбора обратной связи на рынке
- Каждая итерация — простая в управлении контрольная точка проекта
- Стоимость исправления ошибок, допущенных при проектировании системы, значительно ниже, чем в «водопаде» или V-модели

Недостатки

- Есть риск неправильно разбить требования к продукту на инкременты и получить версию, функциональность которой не позволит сделать корректный вывод о необходимости развития продукта

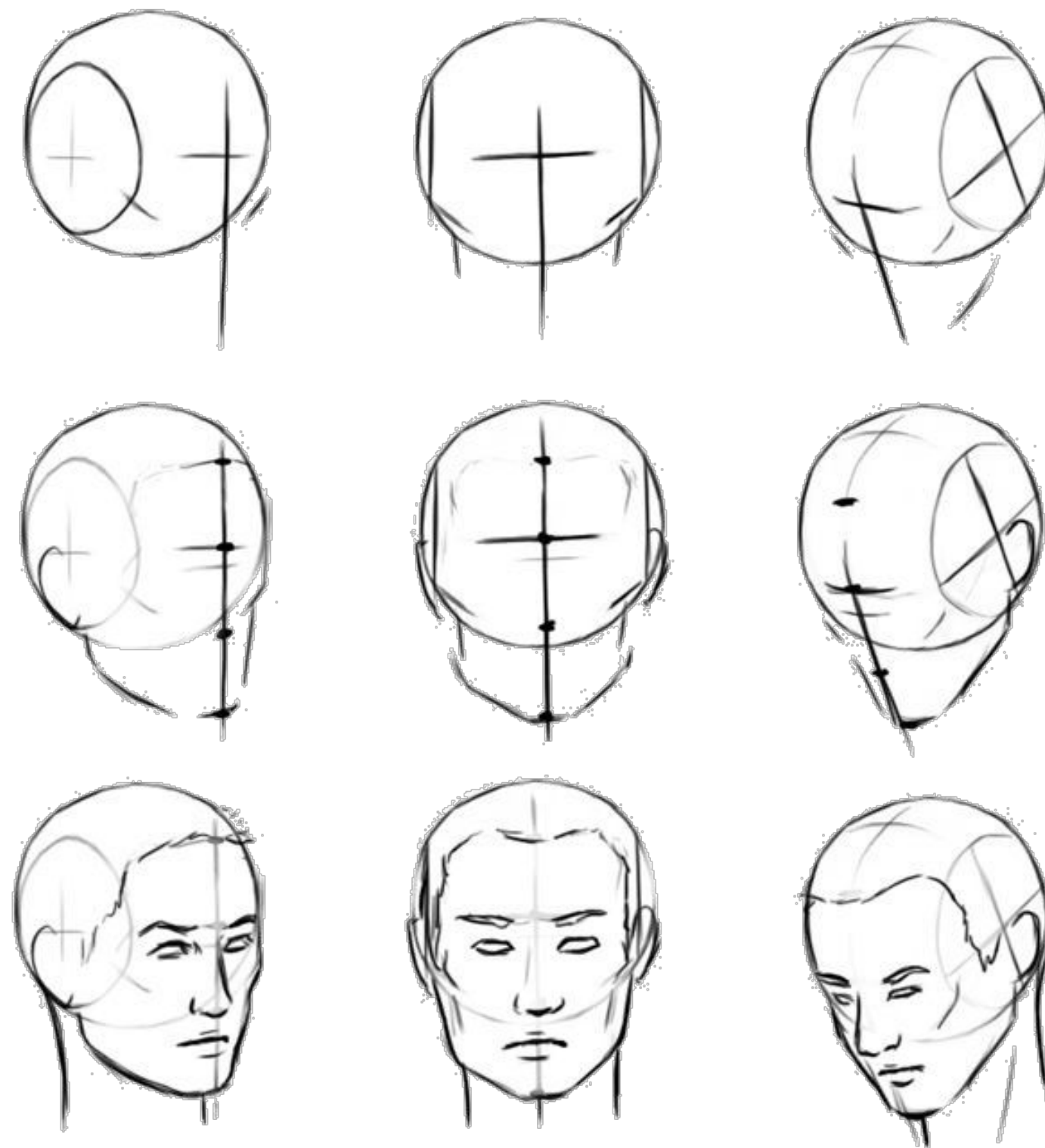
Когда использовать подход?

- Основные требования к системе чётко определены и понятны
- Требуется ранний вывод продукта на рынок



Итерационная модель

Осуществляется небольшими этапами, в ходе которых ведётся анализ полученных промежуточных результатов, выдвигаются новые требования и корректируются предыдущие этапы работы



Итерационная модель

Преимущества

- Снижение рисков за счёт раннего обнаружения противоречий в требованиях
- Организация эффективной обратной связи проектной команды с пользователями продукта
- Быстрый выпуск минимально ценного продукта (MVP)

Недостатки

- Из-за отсутствия чётких требований сложно продумать правильную расширяемую архитектуру. При изменении требований к продукту могут понадобиться дополнительные ресурсы
- Нет фиксированного бюджета и сроков, а также нужна сильная вовлечённость заказчика в процесс

Когда стоит использовать подход?

- Для больших проектов
- Для проектов с высоким уровнем неопределённости
- Для программных продуктов, которые носят инновационный характер и основаны на бизнес-гипотезах, требующих проверки



Итоги занятия

1

Жизненный цикл разработки ПО –
это период времени от момента
принятия решения о необходимости
разработки программного продукта до
момента его полного изъятия
из эксплуатации

2

Этапы жизненного цикла:

- анализ
- проектирование
- разработка и тестирование
- документирование
- внедрение
- сопровождение

3

4 базовых модели разработки ПО:

- водопадная
- V-модель
- инкрементная
- итерационная

4

**Системный аналитик принимает
непосредственное участие
на каждом этапе жизненного
цикла разработки**



Домашнее задание



Цель

Закрепить знания об этапах жизненного цикла разработки программного обеспечения, и о моделях разработки программного обеспечения

Задание

Пройти тестирование

Результат

Тест пройден



Жизненный цикл разработки ПО

Ксения Шипина
Системный аналитик

