# Задание 8

Опишите основные Достоинства, Недостатки и Критерии применимости следующих моделей разработки:

* Waterfall

## Преимущества и недостатки Waterfall

### В число наибольших преимуществ методики Waterfall вошли:

* понятная и простая структура процесса разработки — это снижает порог вхождения для команд
* удобная отчётность — можно легко отследить ресурсы, риски, затраченное время и финансы благодаря строгой этапности процесса разработки и детальной документации проекта
* стабильность задач — задачи, которые стоят перед продуктом, ясны команде с самого начала разработки, и остаются неизменными на протяжении всего процесса
* оценка стоимости и сроков сдачи проекта — сроки выпуска готового продукта, как и его итоговая стоимость могут быть просчитаны до момента запуска разработки.

### Среди недостатков водопадного метода можно выделить:

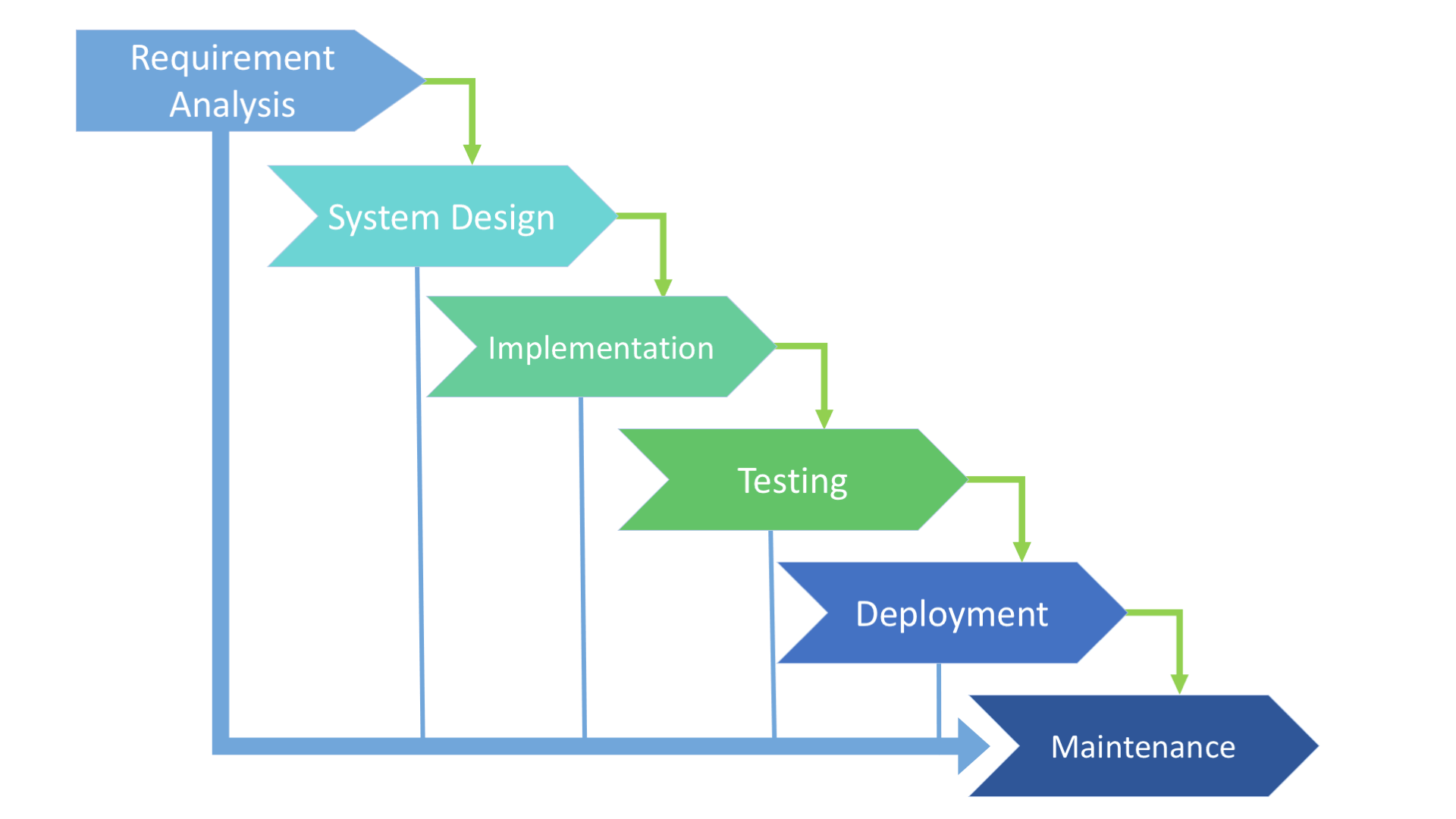
* лишенный гибкости процесс — так, если проект требует больше временных и финансовых ресурсов, чем возможно, то под нож пойдёт фаза тестирования. Согласно исследованиям консалт-группы Rothman, стоимость исправления багов после выпуска продукта выше в среднем в 20 раз, чем во время полноценного многоэтапного тестирования в процессе разработки
* «стойкость» к изменениям — жёсткий каркас из этапов разработки и условие предоставление только готового продукта определяют невозможность вносить изменения во время разработки
* инерционность — на первых стадиях прогноз временных и финансовых трат может измениться в сторону увеличения, но изменить проект в сторону оптимизации затрат, изменения функционала или концепции до выпуска готового продукта невозможно
* повышенный риск — классическая система тестирования подразумевает отдельно тестирование каждого из компонентов проекта, в том числе, во взаимодействии с другими. При использовании Waterfall происходит тестирование готового продукта.

Частично недостатки водопадной модели разработки исправлены в модификациях Waterfall: Сашими, Waterfall с субпроектами и водопадная модель разработки с снижением риска.

## **SDLC MODELS**

### **Waterfall SDLC Model**

Waterfall - это каскадная модель SDLC, в которой процесс разработки выглядит как поток, шаг за шагом продвигающийся по этапам анализа, проектирования, реализации, тестирования, внедрения и поддержки. Эта модель SDLC включает постепенное выполнение каждого этапа полностью. Этот процесс строго документирован и предопределен с функциями, ожидаемыми на каждом этапе этой модели жизненного цикла разработки программного обеспечения.

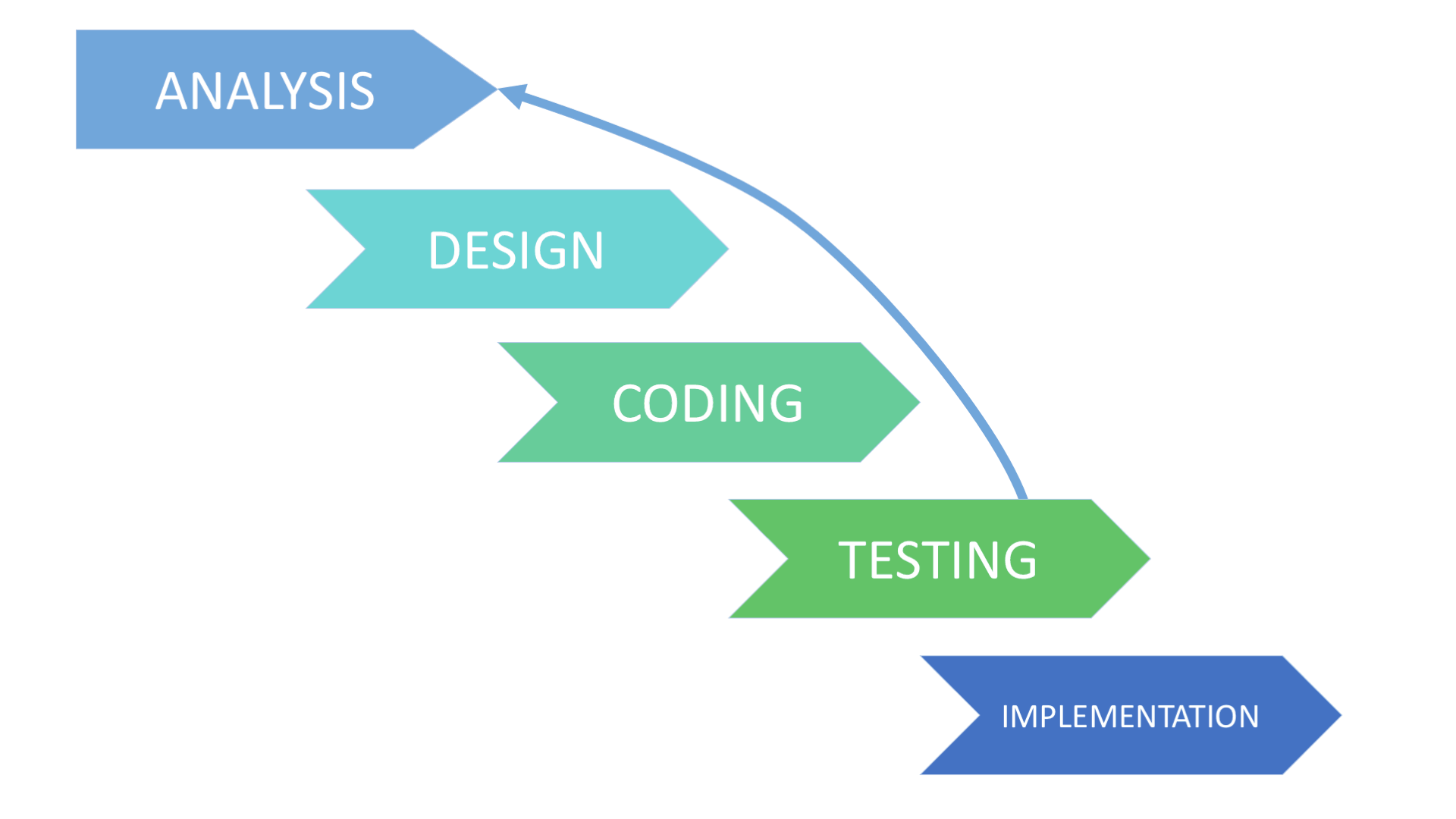


| **ПРЕИМУЩЕСТВА** | **НЕДОСТАТКИ** |
| --- | --- |
| Простой в использовании и понимании | Программа готова только после завершения последнего этапа. |
| Простота управления благодаря жесткости: каждая фаза имеет определенный результат и анализ процесса | Высокие риски и неопределенность |
| Этапы развития идут один за другим | Не лучший выбор для сложных и объектно-ориентированных проектов |
| Идеально подходит для небольших или средних проектов, где требования ясны и не двусмысленны. | Не подходит для долгосрочных проектов. |
| Легко определить ключевые моменты в цикле разработки | Трудно измерить прогресс этапа, пока он все еще находится в разработке. |
| Легко классифицировать и расставлять приоритеты задач | Интеграция производится в самом конце, что не дает возможности заранее выявить проблему. |

**Примеры использования модели SDLC Waterfall:**

* Требования точно задокументированы
* Определение продукта стабильное
* Стек технологий предопределен, что делает его не динамическим.
* Никаких двусмысленных требований
* Проект короткий
* Iterative

### **Iterative SDLC Model**

Итеративная модель SDLC не требует полного списка требований перед запуском проекта. Процесс разработки может начинаться с требований к функциональной части, которые в дальнейшем могут быть расширены. Процесс повторяется, что позволяет создавать новые версии продукта для каждого цикла. Каждая итерация (продолжительностью от двух до шести недель) включает разработку отдельного компонента системы, после чего этот компонент добавляется к ранее разработанному функционалу. Говоря математической терминологией, итерационная модель представляет собой реализацию метода последовательной аппроксимации; это означает постепенное приближение к запланированной форме конечного продукта. 

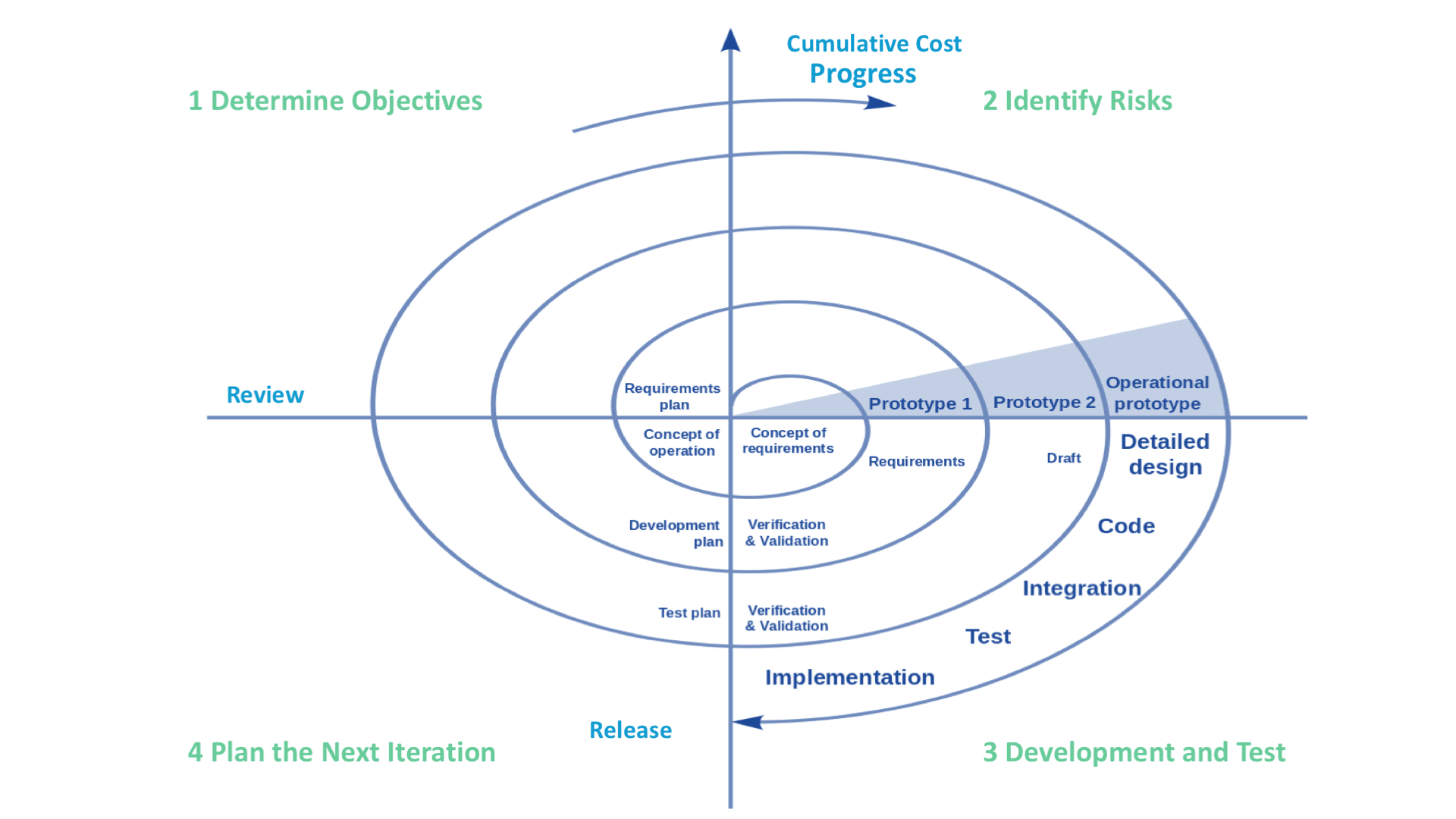
| **ПРЕИМУЩЕСТВА** | **НЕДОСТАТКИ** |
| --- | --- |
| Некоторые функции можно быстро разработать в начале жизненного цикла разработки. | Итерационная модель требует больше ресурсов, чем водопадная модель |
| Параллельная разработка может применяться | Требуется постоянное управление |
| Прогресс легко измерить | Проблемы с архитектурой или дизайном могут возникнуть из-за того, что не все требования предусмотрены на короткой стадии планирования. |
| Чем короче итерация - тем проще этапы тестирования и отладки. | Плохой выбор для небольших проектов |
| Риски легче контролировать, так как задачи с высоким риском выполняются в первую очередь | Процесс сложно управлять |
| Проблемы и риски, определенные в рамках одной итерации, можно предотвратить в следующих спринтах. | Риски могут быть не определены полностью даже на завершающей стадии проекта. |
| Гибкость и готовность к изменению требований | Анализ рисков требует привлечения высококвалифицированных специалистов. |

**Примеры использования итерационной модели:**

* Требования к конечному продукту строго предопределены.
* Применяется к масштабным проектам
* Основная задача предопределена, но детали могут со временем продвигаться.
* Spiral

### **Spiral SDLC Model**

Спиральная модель - это модель SDLC, которая поэтапно объединяет архитектуру и прототипирование. Это комбинация моделей Iterative и Waterfall SDLC со значительным акцентом на анализе рисков. Главный вопрос спиральной модели - это определение подходящего момента для перехода на следующий этап. Предварительно установленные временные рамки рекомендуются как решение этой проблемы. Переход на следующий этап происходит по плану, даже если работа на предыдущем этапе еще не завершена. План внедряется на основе статистических данных, полученных в предыдущих проектах, даже из личного опыта разработчика.



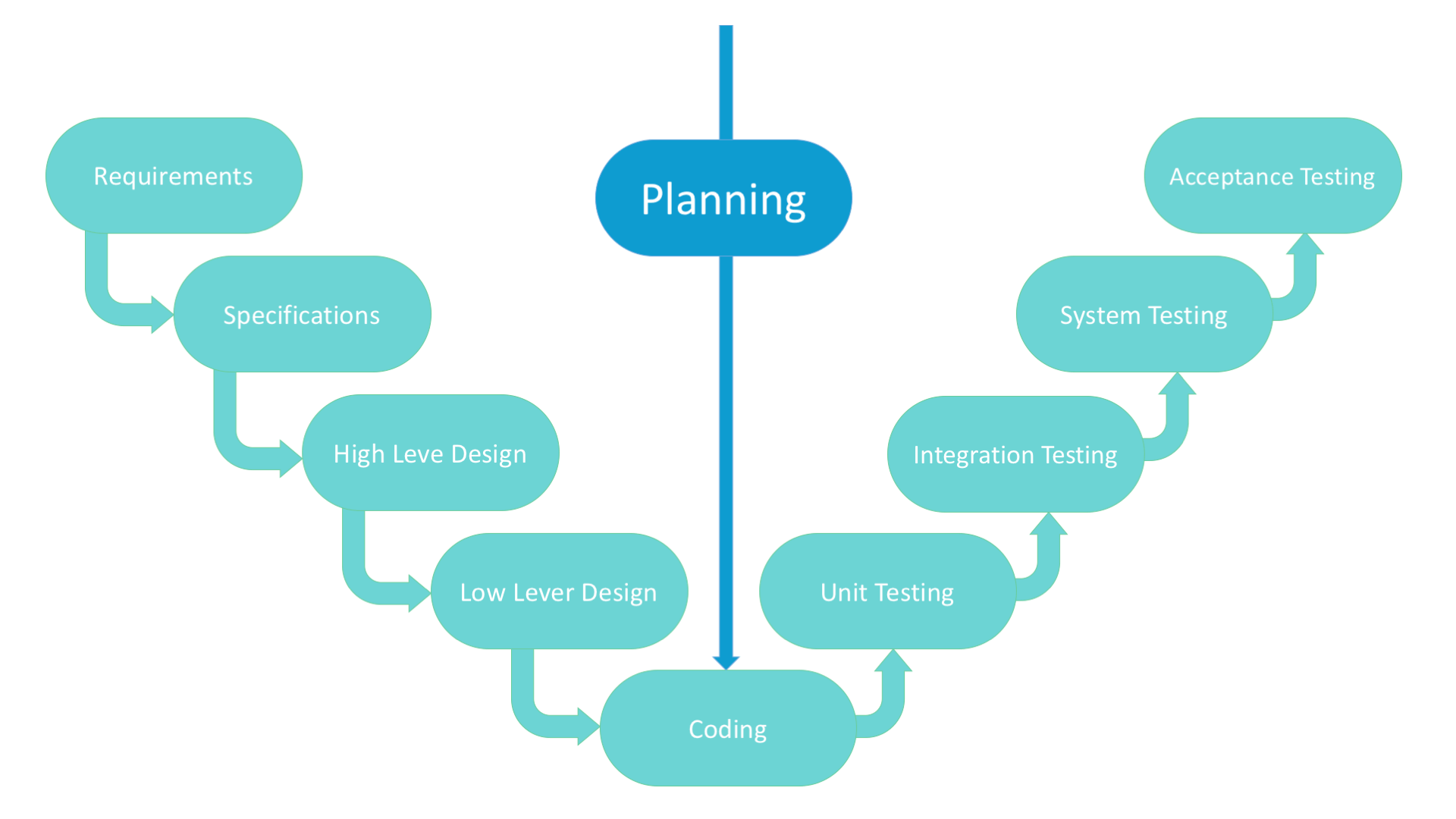
| **ПРЕИМУЩЕСТВА** | **НЕДОСТАТКИ** |
| --- | --- |
| Жизненный цикл делится на небольшие части, и если концентрация риска выше, этап можно завершить раньше, чтобы решить проблемы. | Может быть довольно дорого |
| Процесс разработки точно задокументирован, но масштабируется с учетом изменений. | Контроль рисков требует привлечения высококвалифицированных специалистов. |
| Масштабируемость позволяет вносить изменения и добавлять новый функционал даже на относительно поздних этапах. | Может быть неэффективным для небольших проектов |
| Более ранний рабочий прототип готов - быстрее пользователи смогут указать на недостатки | Большое количество промежуточных этапов требует чрезмерной документации |

**Примеры использования спиральной модели**

* Заказчик не уверен в требованиях
* Ожидается, что основные правки будут внесены в ходе цикла разработки.
* Проекты со средним или высоким уровнем риска, где важно предотвратить эти риски
* Новый продукт, который должен быть выпущен в несколько этапов, чтобы получить достаточное количество отзывов клиентов.
* V-Model

### **V-shaped SDLC Model**

V-образная модель SDLC является расширением классической модели водопада и основана на соответствующем этапе тестирования для каждого этапа разработки. Это очень строгая модель, и следующий этап начинается только после предыдущего. Это также называется моделью «валидация и верификация». Каждый этап имеет текущий контроль процесса, чтобы убедиться, что переход на следующий этап возможен.



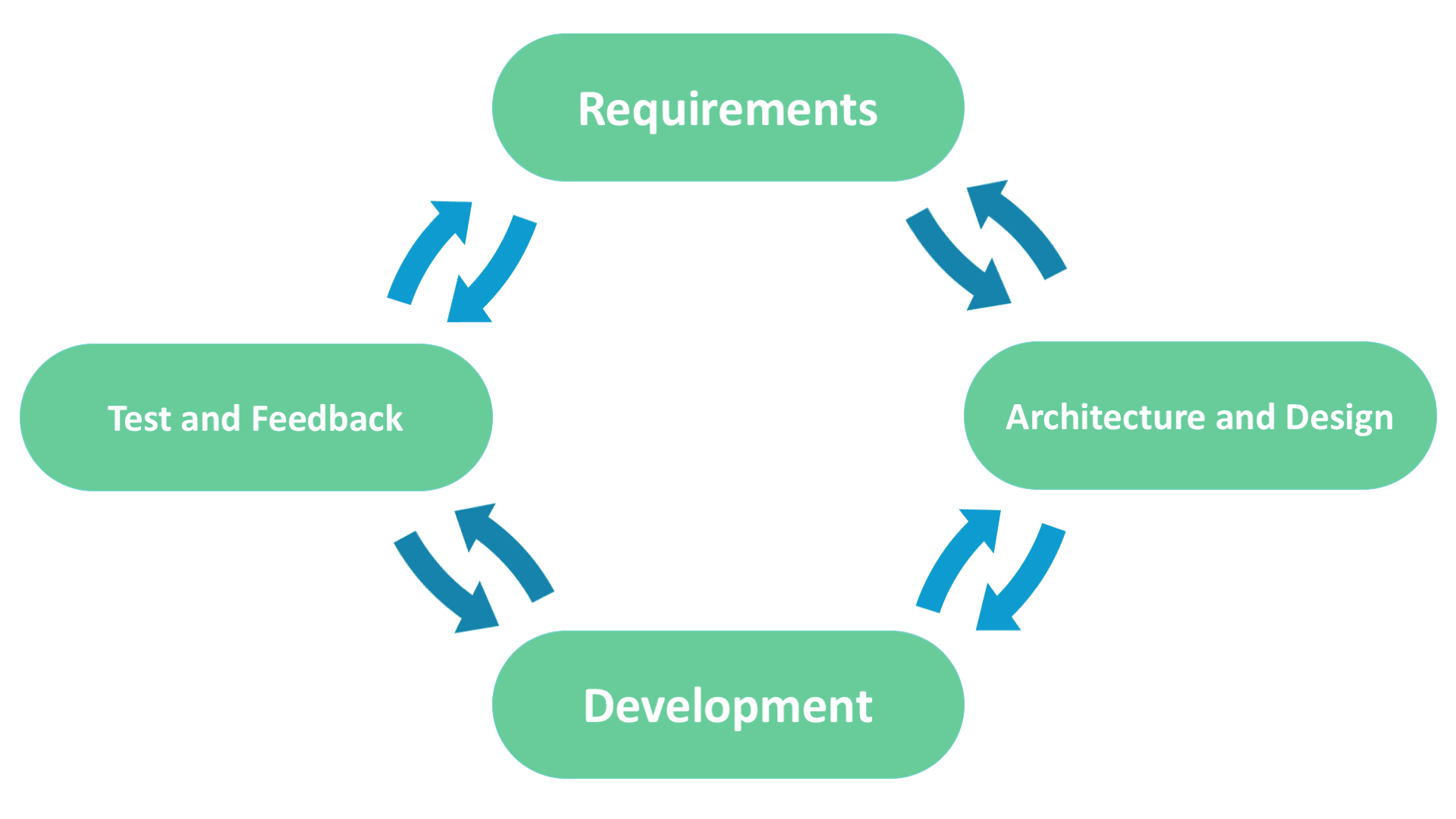
| **ПРЕИМУЩЕСТВА** | **НЕДОСТАТКИ** |
| --- | --- |
| Каждая ступень V-образной модели имеет строгие результаты, поэтому ее легко контролировать. | Отсутствие гибкости |
| Тестирование и проверка происходят на ранних этапах | Плохой выбор для небольших проектов |
| Подходит для небольших проектов, где требования статичны и ясны. | Сравнительно большие риски |

**Варианты использования V-образной модели:**

* Для проектов, где требуется точное тестирование продукта
* Для малых и средних проектов, где требования строго предопределены.
* Инженеры необходимой квалификации, особенно тестировщики, находятся в пределах легкой досягаемости.

### Agile SDLC Model **Гибкая модель SDLC**

В гибкой методологии после каждой итерации разработки заказчик может увидеть результат и понять, доволен он им или нет. Это одно из преимуществ гибкой модели жизненного цикла разработки программного обеспечения. Одним из его недостатков является то, что при отсутствии определенных требований сложно оценить ресурсы и стоимость разработки. Экстремальное программирование - одно из практических применений гибкой модели. В основе такой модели - короткие еженедельные встречи - спринты, которые являются частью подхода Scrum.



| **ПРЕИМУЩЕСТВА** | **НЕДОСТАТКИ** |
| --- | --- |
| В процесс разработки вносятся корректировки функциональных требований для обеспечения конкурентоспособности. | Трудности с определением конечной стоимости из-за постоянных изменений |
| Проект разбит на короткие и прозрачные итерации. | Команда должна быть высокопрофессиональной и клиентоориентированной. |
| Риски сведены к минимуму благодаря гибкому процессу внесения изменений | Новые требования могут противоречить существующей архитектуре |
| Быстрый выпуск первой версии продукта | Со всеми исправлениями и изменениями есть вероятность того, что проект превысит ожидаемые сроки. |

**Примеры использования Agile-модели:**

* Потребности пользователей меняются динамически
* Меньшая цена за внесенные изменения из-за большого количества итераций
* В отличие от модели Waterfall, для запуска проекта требуется только первоначальное планирование.