

# Разработка программного обеспечения (Software Engineering)

Создание ПО

# Терминология

**Процесс создания ПО** – совокупность процессов, приводящих к созданию программного продукта.

Фундаментальные процессы, присущие любому проекту создания ПО:

- ▶ Разработка спецификации требований на ПО (Определяют функциональные характеристики системы и обязательны для выполнения)
- ▶ Создание программного обеспечения (создание ПО согласно спецификации)
- ▶ Аттестация ПО (Созданное ПО должно пройти аттестацию для подтверждения соответствию требованиям заказчика)
- ▶ Модернизация ПО (совершенствование ПО согласно измененным требованиям потребителя)

# Базовые процессы создания ПО

- ▶ Разработка спецификации
- ▶ Проектирование и реализация
- ▶ Аттестация
- ▶ Эволюция

**Жизненный цикл ПО** – совокупность процессов, протекающих от момента принятия решения о создании ПО до его полного вывода из эксплуатации

# Каскадная модель



# Каскадная модель

## Достоинства:

- Документирование каждого этапа

## Недостатки:

- «негибкое» разбиение процесса создания на отдельные этапы

## Применение:

Подход хорошо зарекомендовал себя при построении информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования с целью предоставить разработчикам свободу реализовать их технически как можно лучше.

# Модель прототипирования

Прототип – действующий ПП, реализующий отдельные функции и внешние интерфейсы разрабатываемого ПО.

Достоинства:

- ▶ Взаимодействие заказчика с разрабатываемой системой начинается на раннем этапе;
- ▶ Благодаря реакции заказчика на прототип сводится к минимуму число неточностей в требованиях;
- ▶ В процессе разработки всегда можно учесть новые, даже неожиданные требования заказчика;
- ▶ Заказчик всегда видит прогресс в процессе разработки ПП;
- ▶ Уменьшается число доработок, что снижает стоимость разработки;
- ▶ Заказчики принимают участие в процессе разработки на протяжении всего ЖЦ и в конечном итоге в большей степени довольны результатом работы;

# Модель прототипирования

## Недостатки:

- ▶ Решение сложных задач может отодвигаться на будущее;
- ▶ Заказчик может предпочесть получить прототип, а незаконченную полную версию ПП;
- ▶ Прототипирование может неоправданно затянуться;
- ▶ Перед началом работы неизвестно, сколько итераций придется выполнить;

# Технология RAD

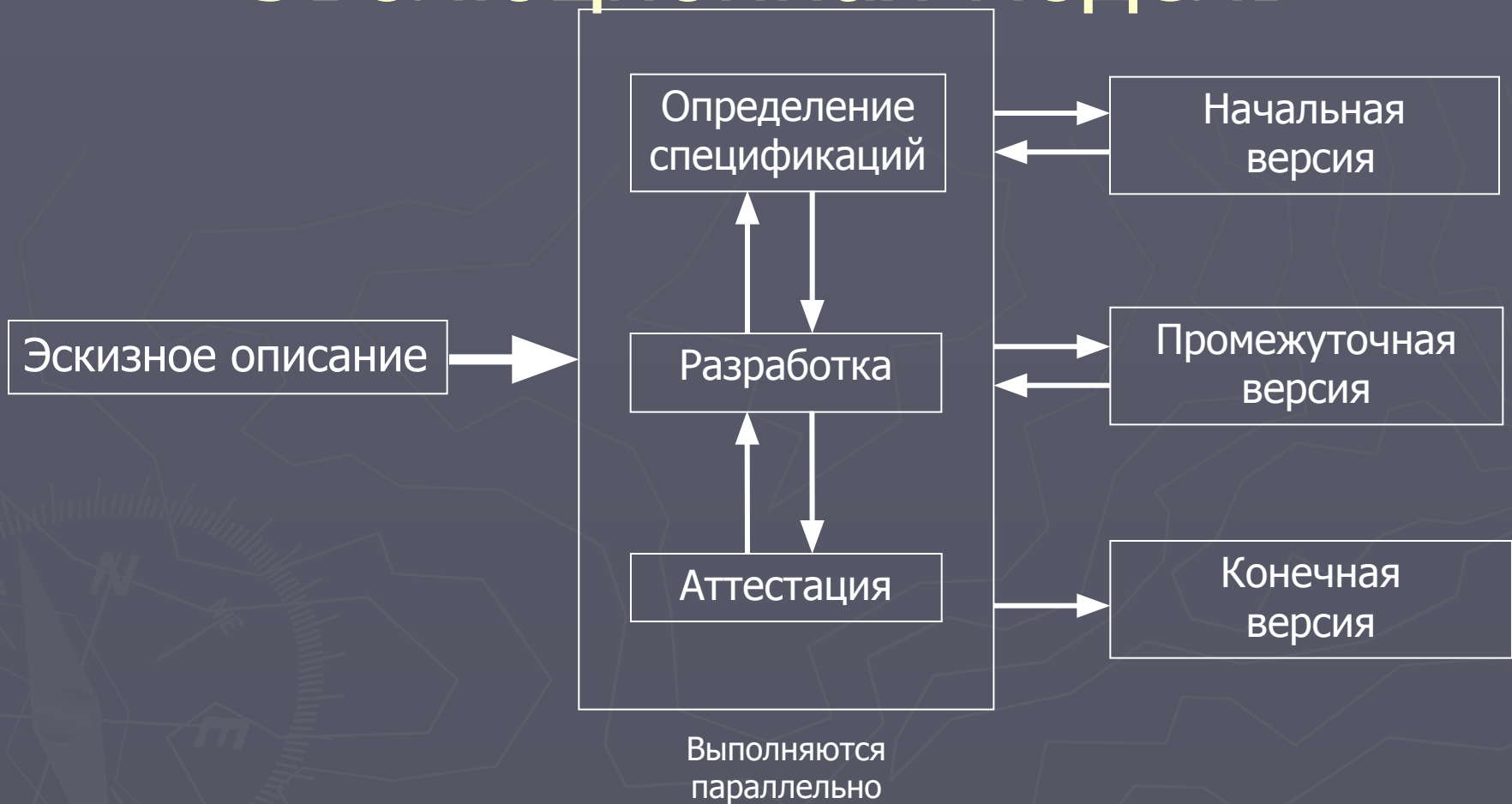
Rapid Application Development — быстрая разработка приложений.

Ориентирована на максимально быстрое получение первых версий разрабатываемого ПО. Она предусматривает:

- ведение разработки **небольшими группами** (3-7 человек), каждая из которых проектирует и реализует **отдельные подсистемы**, позволяет улучшить управляемость проекта;
  - использование **готовых компонентов** способствует уменьшению времени получения работоспособного прототипа;
  - наличие четко проработанного **графика** цикла, рассчитанного не более чем на три месяца, существенно увеличивает эффективность работы.
- ▶ RAD хорошо зарекомендовал себя для относительно небольших **стандартных проектов**, разрабатываемых для конкретного заказчика.
  - ▶ RAD – модель можно применять при разработке ПП, которые хорошо поддаются моделированию, когда требования к ПП хорошо известны, а заказчик может принять непосредственное участие в процессе разработки.



# Эволюционная модель



Прототип – действующий программный модуль, реализующий отдельные функции создаваемого ПО.

# Эволюционная модель

## Достоинства:

- Спецификация разрабатывается постепенно, по мере требования заказчика.

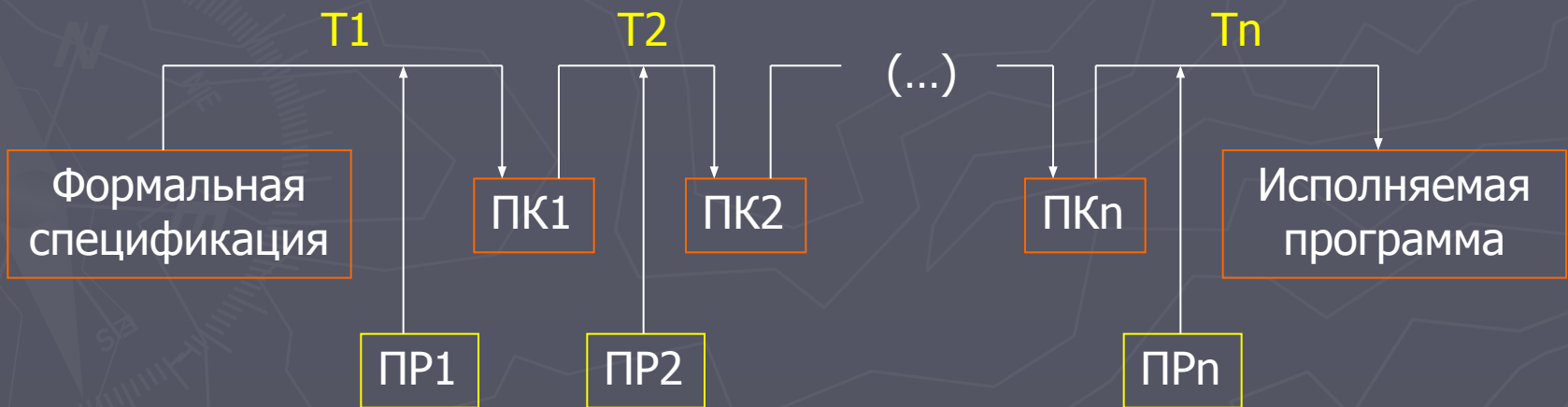
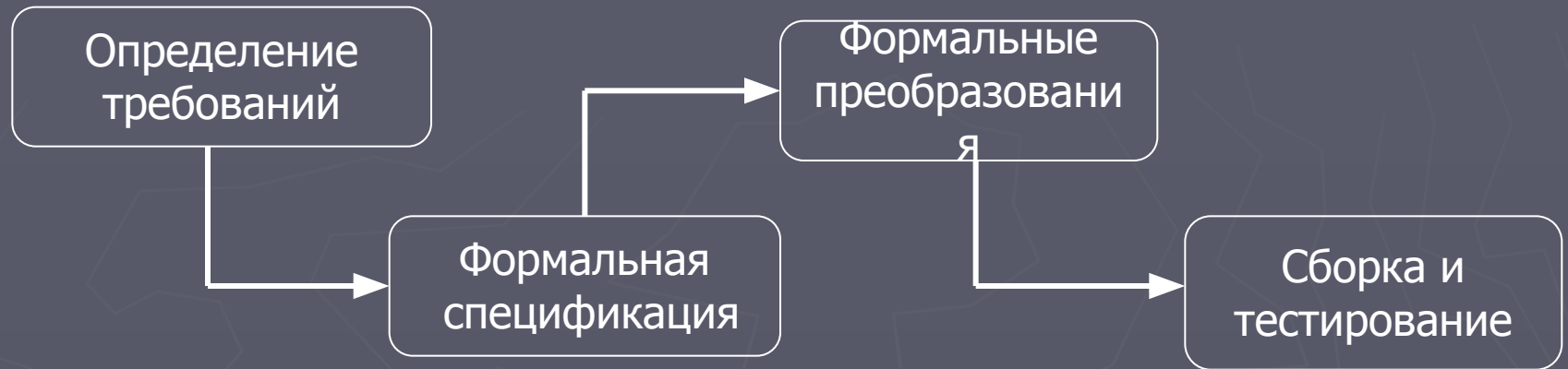
## Недостатки:

- Многие этапы создания ПО не документированы.
- Система часто получается плохо структурированной.
- Требуются специальные средства и технологии разработки ПО.

## Применение:

- Разработка небольших систем (<100 000 строк) или средних (<500 000 строк) с относительно коротким сроком жизни.

# Формальная разработка



Процесс формальных преобразований

# Формальная разработка

## Преимущества:

- Точное соответствие программы спецификации.
- Отказ от тестирования отдельных модулей.
- Тестирование всей системы только после ее сборки.

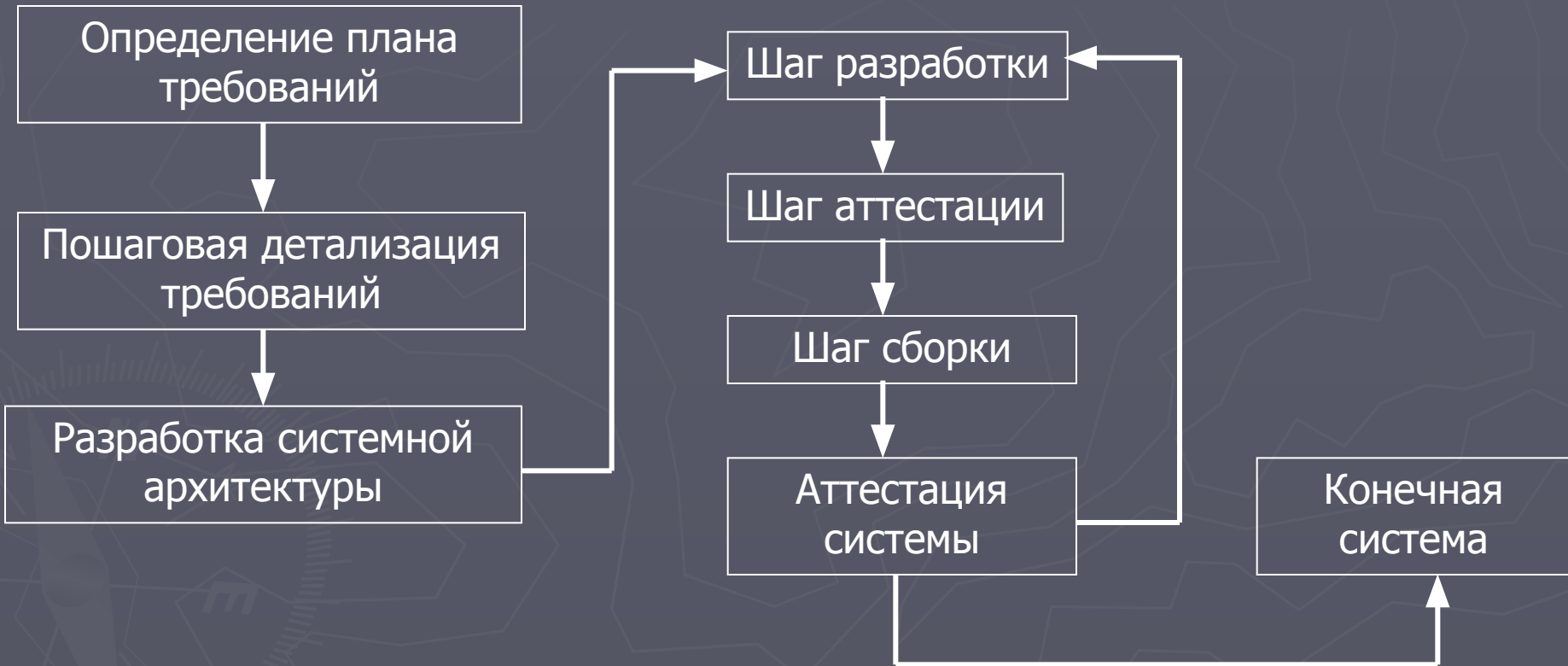
## Недостатки:

- Требуют специальных знаний и опыта использования.
- Не дают существенного выигрыша в стоимости разработки.
- Большинство сложных систем с трудом поддаются формальному описанию.

## Применение:

- Метод «Чистой комнаты» (IBM).

# Модель пошаговой разработки



На каждом шаге отсутствует требование использования одного и того же подхода к процессу разработки!

# Модель пошаговой разработки

## Достоинства:

- Нет необходимости ждать полного завершения разработки системы.
- Можно использовать компоненты, полученные на первых шагах, как прототипы.
- Уменьшается риск общесистемных ошибок.
- Системные сервисы с высоким приоритетом разрабатываются первыми, а все последующие интегрируются с ними. Это позволяет снизить вероятность программных ошибок в особо важных частях системы.

## Недостатки:

- Компоненты, получаемые на каждом шаге, имеют небольшой размер.
- Сложно определить на первых этапах общесистемные функции.
- Невозможно сразу определить набор базовых свойств, которые зачастую разрабатываются совместно с другими частями системы.

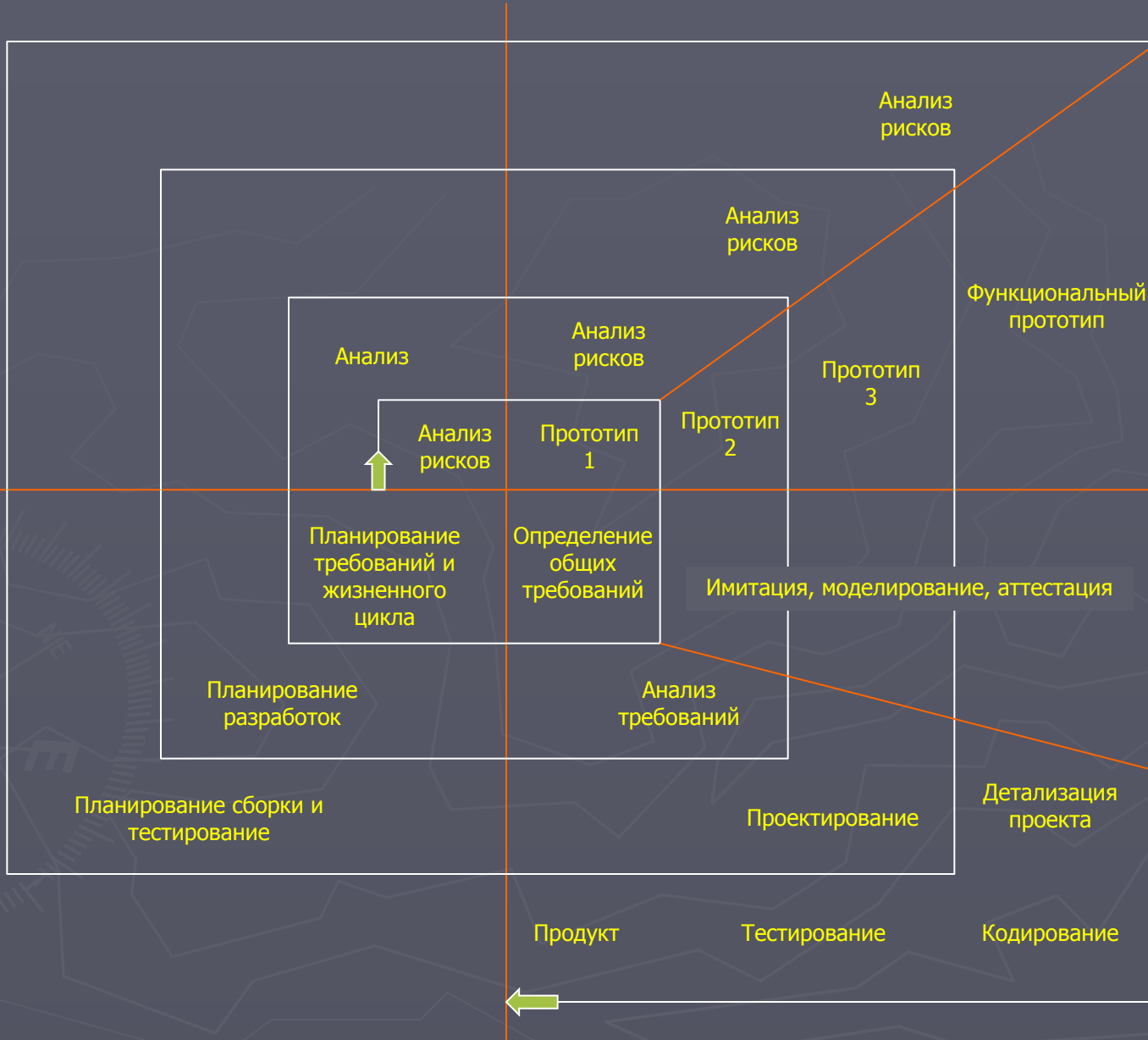
# Спиральная модель

Определение целей, альтернатив и ограничений

Планирование следующей итерации

Оценка альтернатив, оценка и разрешение рисков

Разработка и тестирование продукта на очередной итерации



# Спиральная модель

## Достоинства:

- Нет фиксированных этапов.
- Эта модель может включать в себя любые другие модели на каждом витке спирали –
  - ▶ прототипирование может использоваться при нечетком определении требований;
  - ▶ Каскадная модель в случае последовательного выполнения некоторых этапов;
  - ▶ Модель формальных преобразований – если четко сформулированы требования.

## Недостатки:

- Сложна автоматизация процессов разработки.
- Огромная роль при разработке системы отводится управлению проектом.



# V-образная модель



**V – образная модель демонстрирует комплексный подход к определению фаз процесса разработки ПО. В ней подчеркнуты взаимосвязи, существующие между аналитическими фазами и фазами проектирования, которые предшествуют кодированию, после которого следуют фазы тестирования. Пунктирные линии означают, что эти фазы необходимо рассматривать параллельно.**

### **Преимущества:**

- ▶ Большая роль придается верификации и аттестации ПП, начиная с ранних стадий его разработки, все действия планируются;
- ▶ Предполагаются аттестация и верификация не только самого ПП, но и всех полученных внутренних и внешних данных;
- ▶ Ход выполнения работы может легко отслеживаться, так как завершение каждой фазы является контрольной точкой;
- ▶ модель проста в использовании (относительно проекта, для которого она является приемлимой).

### **Недостатки:**

- ▶ с ее помощью непросто справиться с параллельными событиями;
- ▶ в ней не учтены итерации между фазами;
- ▶ Нельзя вносить изменения на разных этапах жизненного цикла;
- ▶ Тестирование требований происходит слишком поздно, поэтому внесение изменений влияет на выполнение графика работ.