

Методологии проектирования ПО

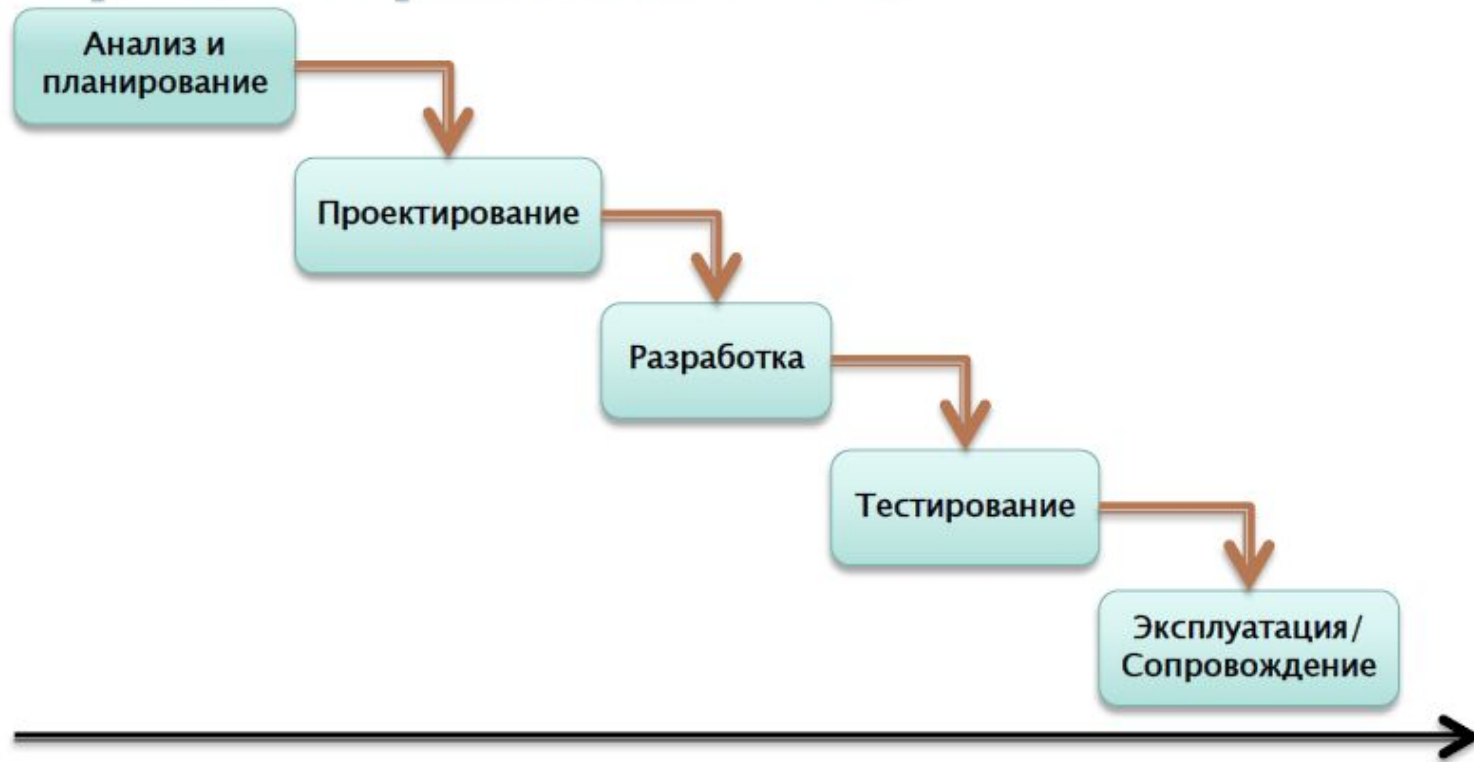
1. Классическая модель разработки ПО: модификации модели, жизненный цикл, достоинства и недостатки

Классическая модель проектирования ПО

- ▶ Предложена в 1960-х годах, впервые описана 1970 г., В. Ройсом
- ▶ Водопадный (однократный) подход
- ▶ Относится к прогнозирующим методологиям
- ▶ Предполагает полное наличие всех требований на момент старта проекта
- ▶ Требования не могут меняться в процессе проектирования
- ▶ Программный продукт появляется по окончании проектирования
- ▶ Промежуточные версии не предусмотрены



Классическая модель проектирования ПО



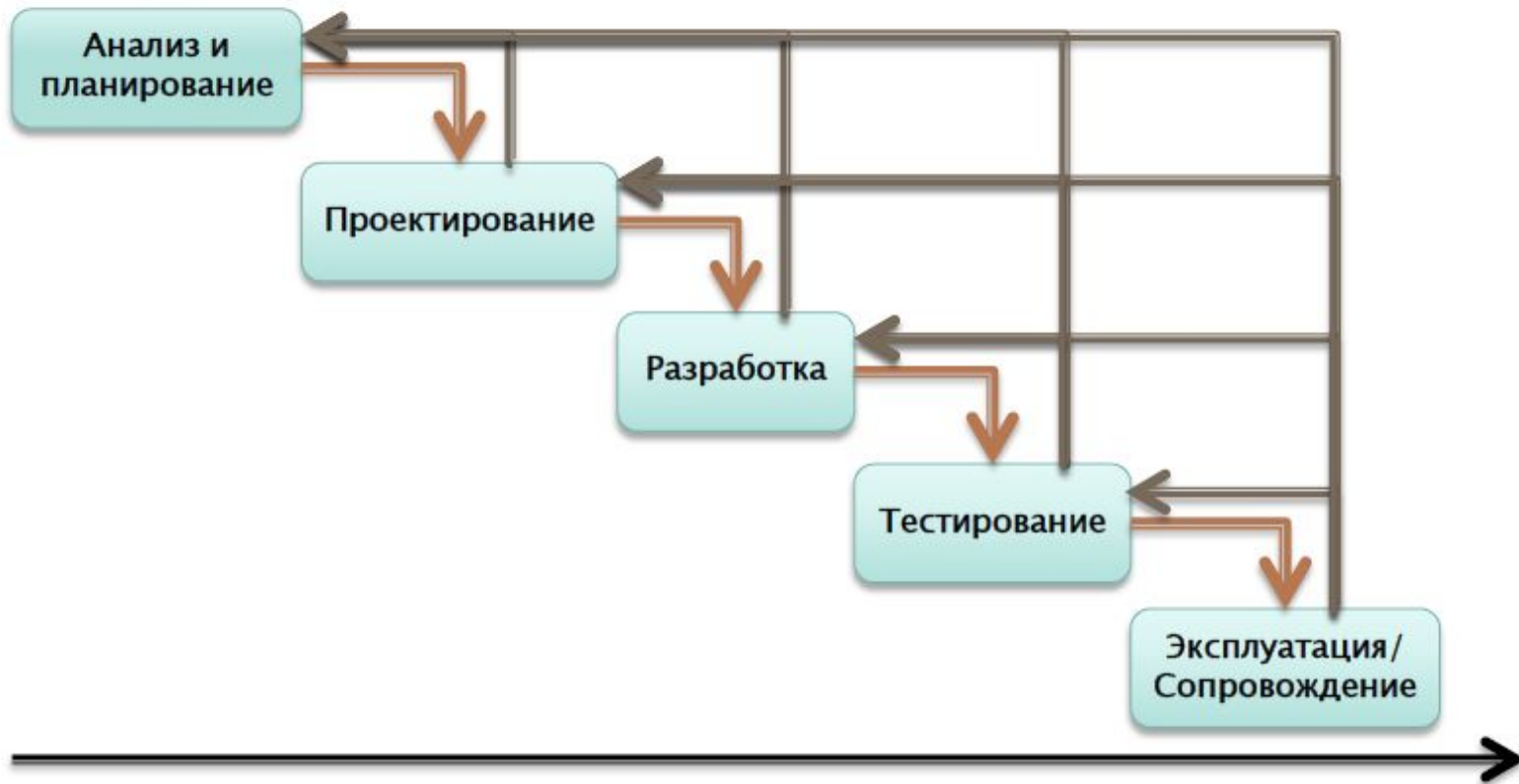
Классическая модель проектирования ПО

- ▶ Анализ и планирование
 - Сбор требований
 - Анализ требований
 - Планирование проекта
- ▶ Проектирование
 - Разработка архитектуры
 - Разработка моделей данных
 - Разработка алгоритмов
- ▶ Реализация
 - Кодирование
 - Отладка
- ▶ Тестирование/верификация
- ▶ Сопровождение
 - Внедрение
 - Эксплуатация
 - Внесение изменений

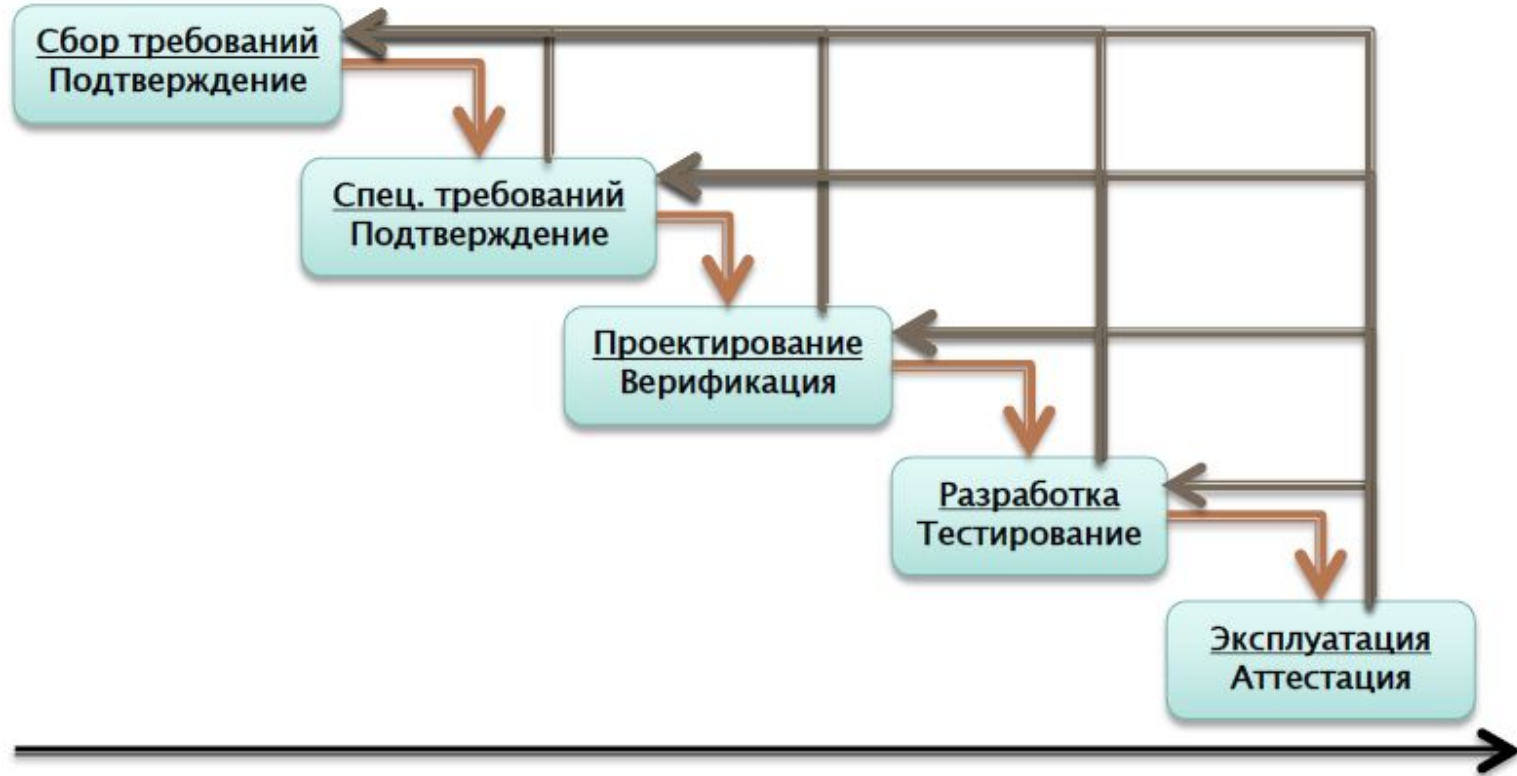
Классическая модель проектирования ПО

- ▶ Имеется несколько модификаций
 - Общепринятая линейная модель
 - Классическая итерационная
 - Предложена В. Ройсом, 1970 г.
 - Обратная связь после каждого этапа
 - Каскадная модель
 - Завершение каждого этапа проверкой
 - Строгая каскадная модель
 - Минимизация возвратов к пройденным этапам

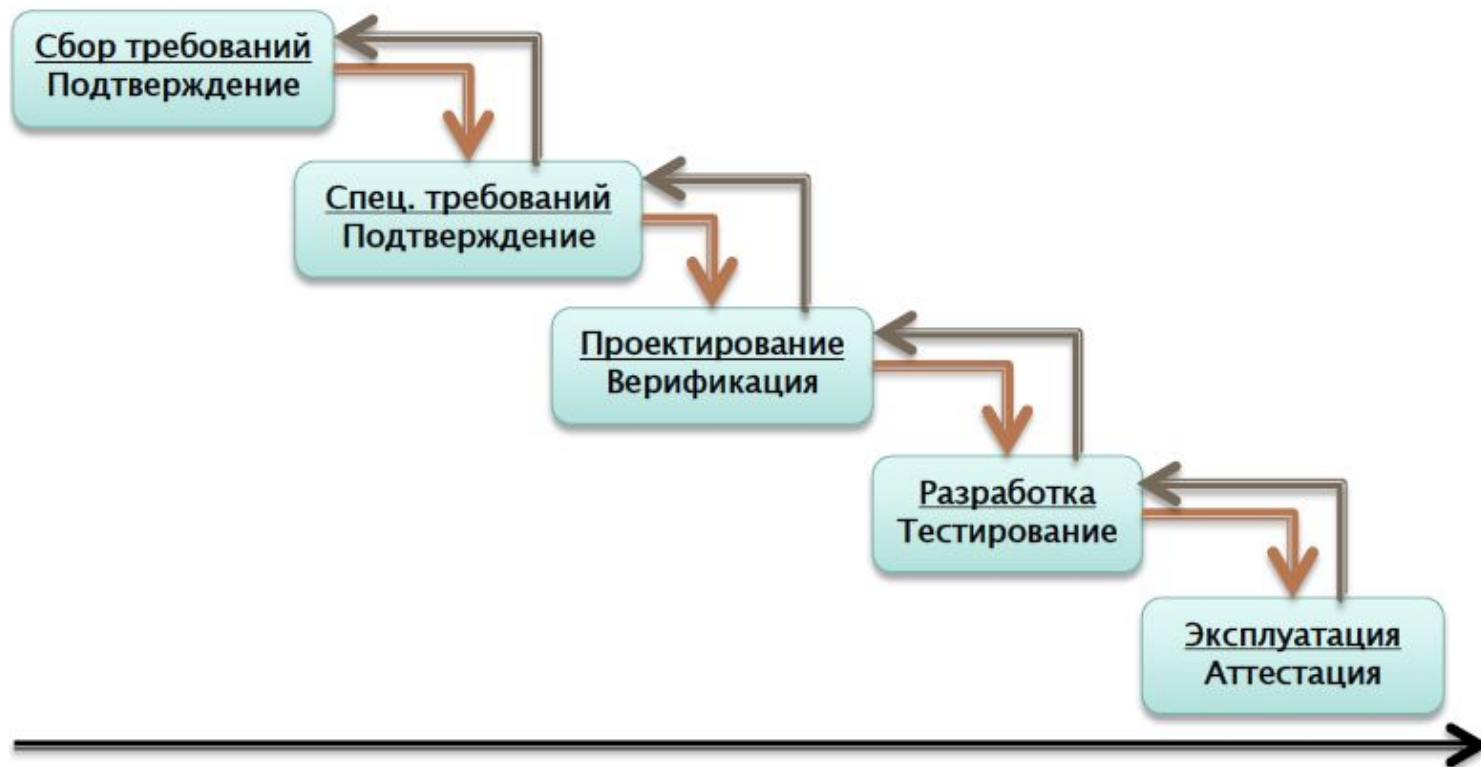
Классическая итерационная модель ППО



Каскадная модель



Строгая каскадная модель

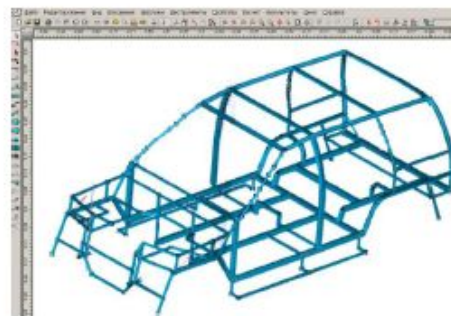


Классическая модель проектирования ПО

- ▶ **Достоинства:**
 - Имеется план и график по всем этапам конструирования
 - Ход конструирования – упорядочен
 - Имеется богатый опыт использования
- ▶ **Недостатки:**
 - Не всегда соответствует реальным проектам (отсутствует гибкость)
 - Часто всех требований на начальном этапе нет
 - Результат доступен только в конце

Прототипирование (макетирование)

- ▶ Применяется, когда имеются не все требования
- ▶ Позволяет быстро увидеть некоторые свойства продукта
 - Удобство
 - Внешний вид
 - Применимость
- ▶ Часто применяется при проектировании
 - Информационных систем
 - Программных продуктов с ГПИ
- ▶ Используются средства быстрой разработки приложений



Прототипирование

1. Сбор и уточнение требований
2. Быстрое проектирование
3. Построение макета
4. Оценка макета заказчиком
 - Заказчик не удовлетворен
 - Уточнение требований
 - Переход к п. 2
 - Заказчик удовлетворен
 - Переход к п. 5
5. Конструирование продукта

Прототипирование



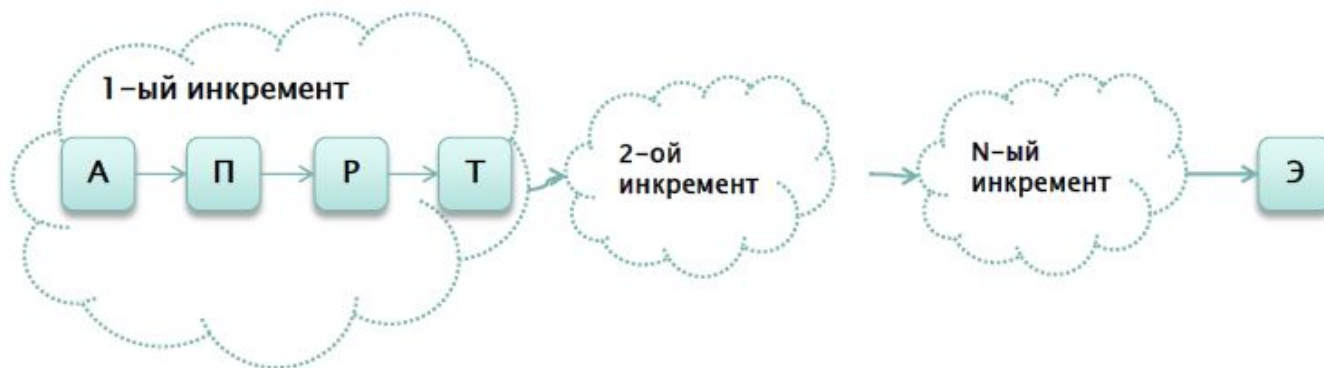
Прототипирование

- ▶ **Достоинства:**
 - Обеспечивает определение полных требований к ПО
- ▶ **Недостатки:**
 - По сути не является полным ЖЦ
 - Заказчик может принять макет за продукт
 - Разработчик может принять макет за продукт

Инкрементная модель

- ▶ Объединяет классический подход и макетирование
- ▶ Весь проект делится на инкременты – версии продукта с определенной функциональностью
- ▶ Для каждого инкремента выполняется:
 - Анализ
 - Проектирование
 - Разработка
 - Тестирование
- ▶ Результат каждого инкремента – работающий продукт

Инкрементная модель

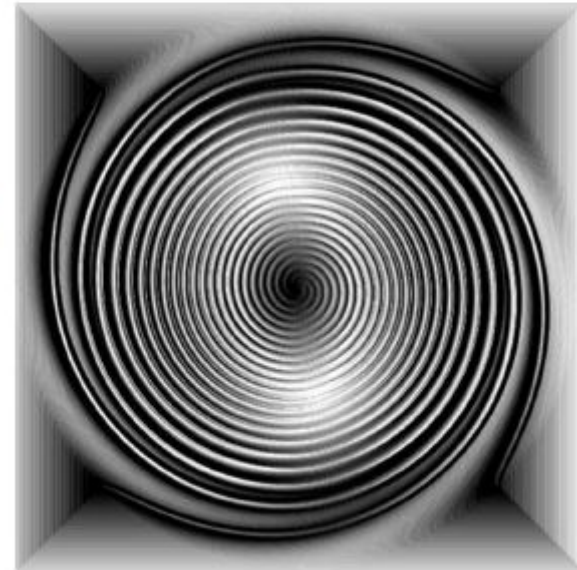


Инкрементная модель

- ▶ Достоинства:
 - Имеется план и график по всем этапам конструирования
 - Промежуточные версии доступны заказчику
- ▶ Недостатки:
 - Часто всех требований на начальном этапе нет
 - Не всегда можно заранее спланировать содержание версий
 - Отсутствует гибкость

Спиральная модель

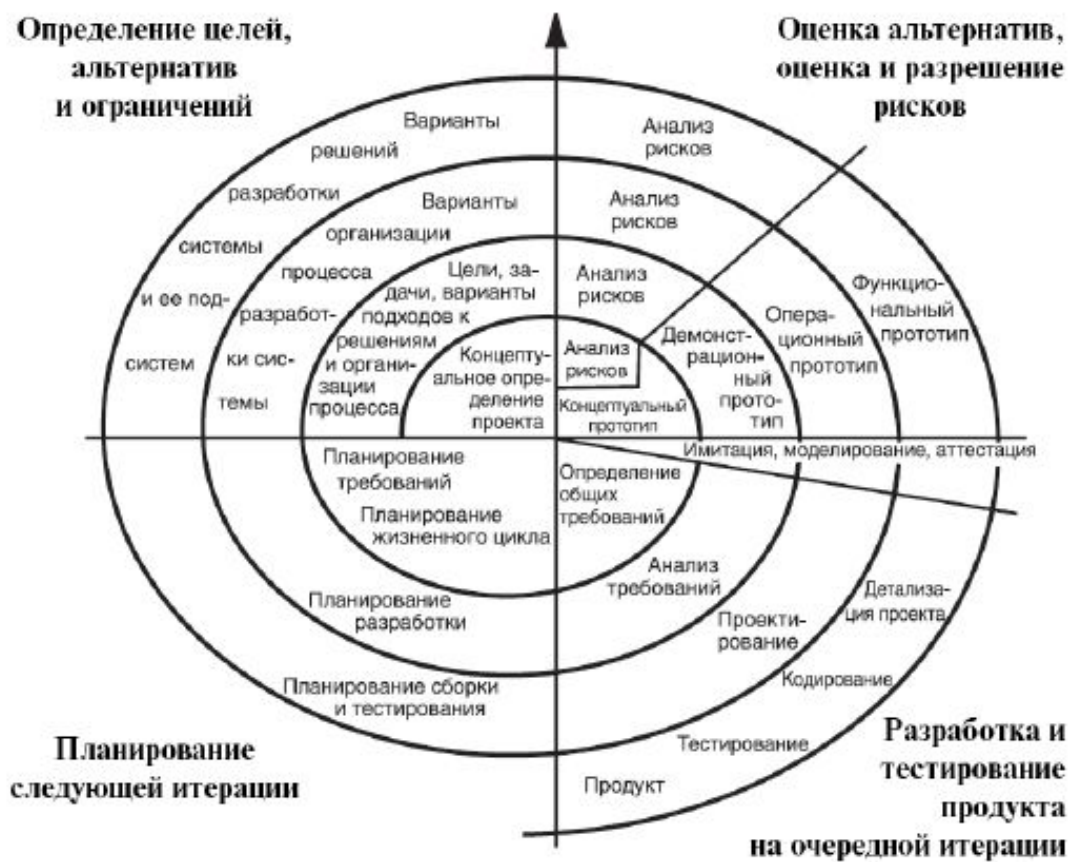
- ▶ Предложена Б. Боемом, 1988г
- ▶ Базируется:
 - На классическом ЖЦ
 - На макетировании
- ▶ Дополнена анализом рисков
- ▶ Основные компоненты
 - Планирование
 - Анализ
 - Конструирование
 - Оценивание



Спиральная модель



Инструментальная спиральная модель



Спиральная модель ППО

▶ Достоинства:

- Адекватно отражает эволюционный характер проектирования
- Позволяет явно учитывать риски на каждой витке эволюции
- Использует моделирование

▶ Недостатки:

- Высокие требования к заказчику
- Трудность контроля времени разработки и управления им

Быстрая разработка приложений

Быстрая разработка приложений (RAD)

RAD = Rapid Application Development

- ▶ Инкрементная стратегия конструирования
- ▶ Использование компонентно-ориентированного конструирования
- ▶ Обеспечение очень короткого цикла разработки (60-90 дней)
- ▶ Ориентирована в основном на разработку ИС

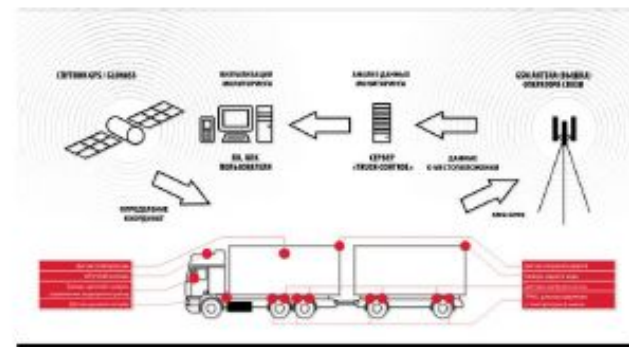


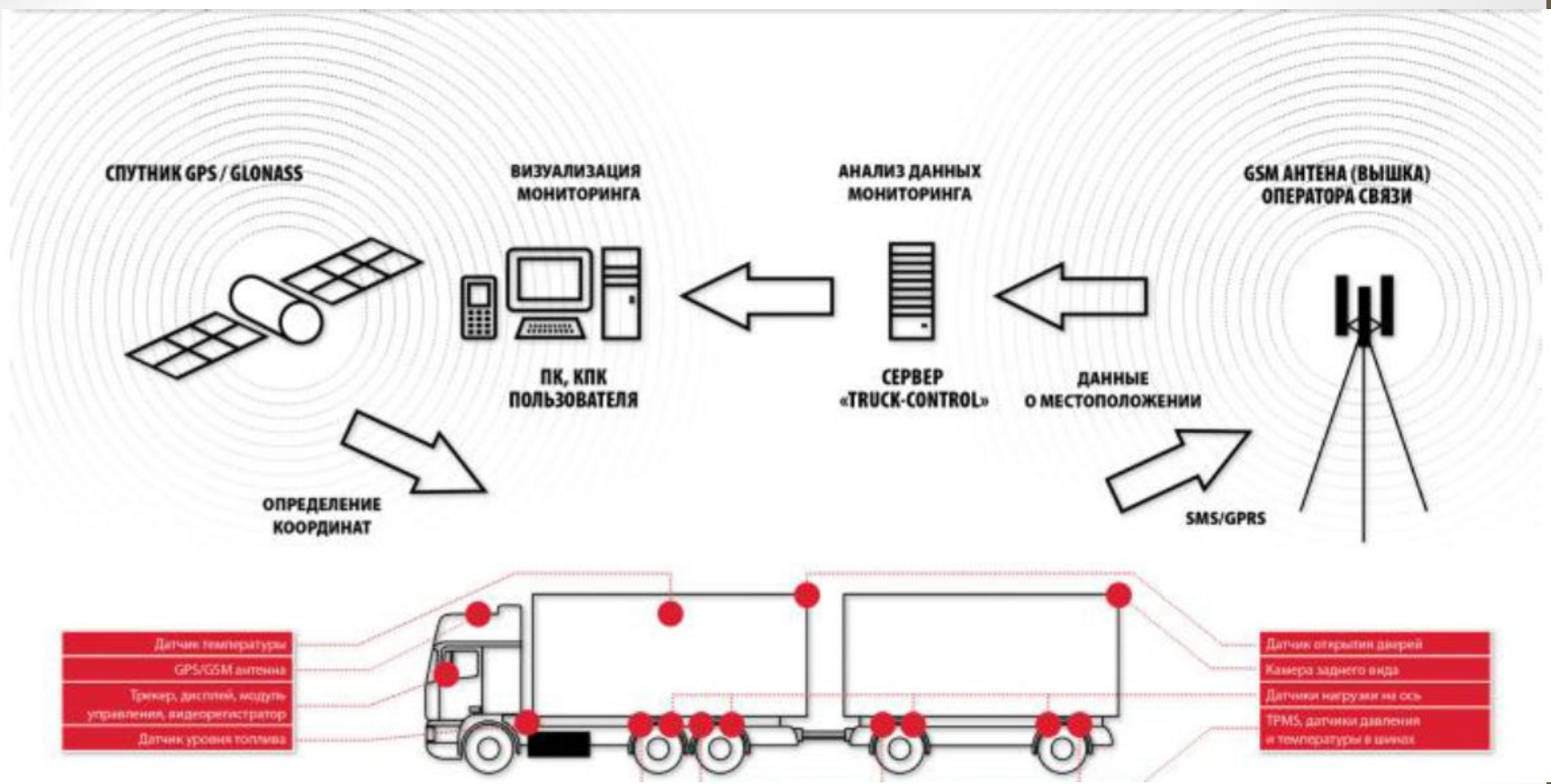
RAD. Основные этапы

- ▶ Бизнес-моделирование
- ▶ Моделирование данных
- ▶ Моделирование обработки
- ▶ Генерация приложения
- ▶ Тестирование и объединение

RAD. Бизнес-моделирование

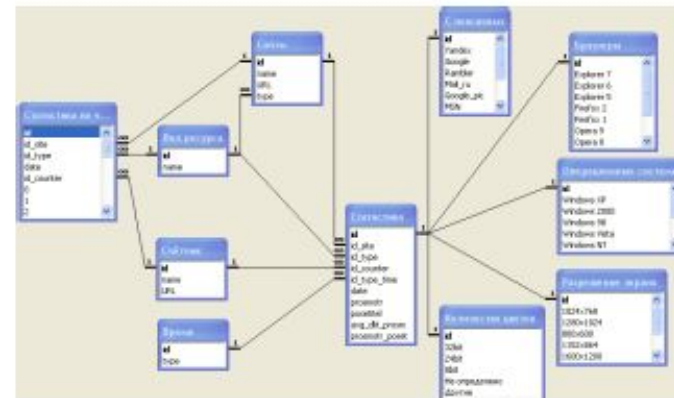
- ▶ Моделируется информационный поток между бизнес-функциями
- ▶ Определяется:
 - Какая информация создается
 - Кто ее создает
 - Кто ее обрабатывает
 - Где информация применяется





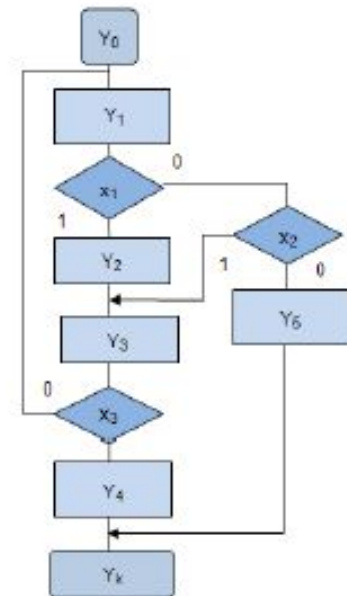
RAD. Моделирование данных

- ▶ По информационному потоку формируется набор объектов данных
- ▶ Определяются свойства объектов
- ▶ Специфицируются отношения между объектами



RAD. Моделирование обработки

- ▶ Определение преобразований объектов данных
- ▶ Создаются описания для
 - добавления объектов данных
 - модификации объектов данных
 - удаления объектов данных
 - поиска объектов данных



RAD. Генерация приложения

- ▶ Использование ЯП 4-го поколения
- ▶ Использование готовых компонентов
- ▶ Создание повторно используемых компонентов
- ▶ Использование средств автоматизации



RAD. Тестирование и объединение

- ▶ Тестирование упрощается из-за повторного использования компонентов
 - Они не требуют автономного тестирования
- ▶ Используется интеграционное тестирование



RAD. Ограничения

- ▶ Область применения – проектирование информационных систем
- ▶ Производительность не является критичной
 - Неприменимо для задач реального времени
- ▶ Можно привлечь достаточно разработчиков
- ▶ Отсутствуют технические риски

Rational Unified Process

- ▶ Авторы:
 - А. Якобсон
 - Г. Буч
 - Дж. Рембо
- ▶ Продвигается IBM Rational
- ▶ Начало разработки - 1995 г.
- ▶ Первая версия RUP - 1998 г.
- ▶ Наиболее глубоко проработанная методология

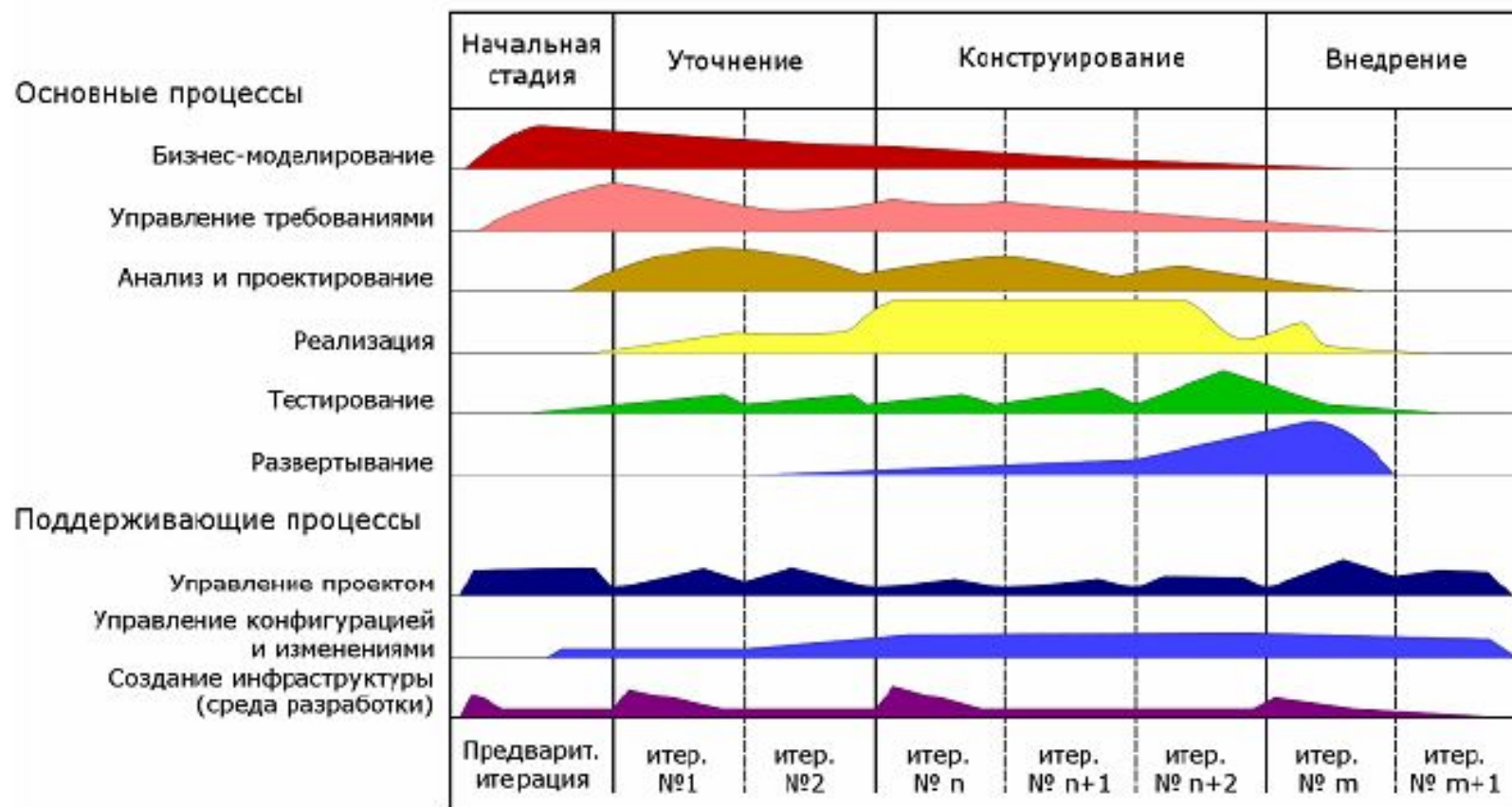


Rational Unified Process

- ▶ Инкрементная и эволюционная итеративная методология
- ▶ Базируется на широком использовании UML
- ▶ На всех стадиях используются программные метрики
- ▶ **Процесс** делится на этапы (стадии)
- ▶ Каждый **этап** состоит из итераций
- ▶ **Итерация** – законченный цикл разработки, вырабатывающий промежуточный продукт

Рабочие процессы

Стадии



Итерации

RUP. Рабочие потоки процесса

- ▶ Бизнес-моделирование
- ▶ Управление требованиями
- ▶ Анализ и проектирование
 - Создание статического и динамического представления системы
- ▶ Реализация
 - Создание программного кода
- ▶ Тестирование
 - Проверка системы в целом

