

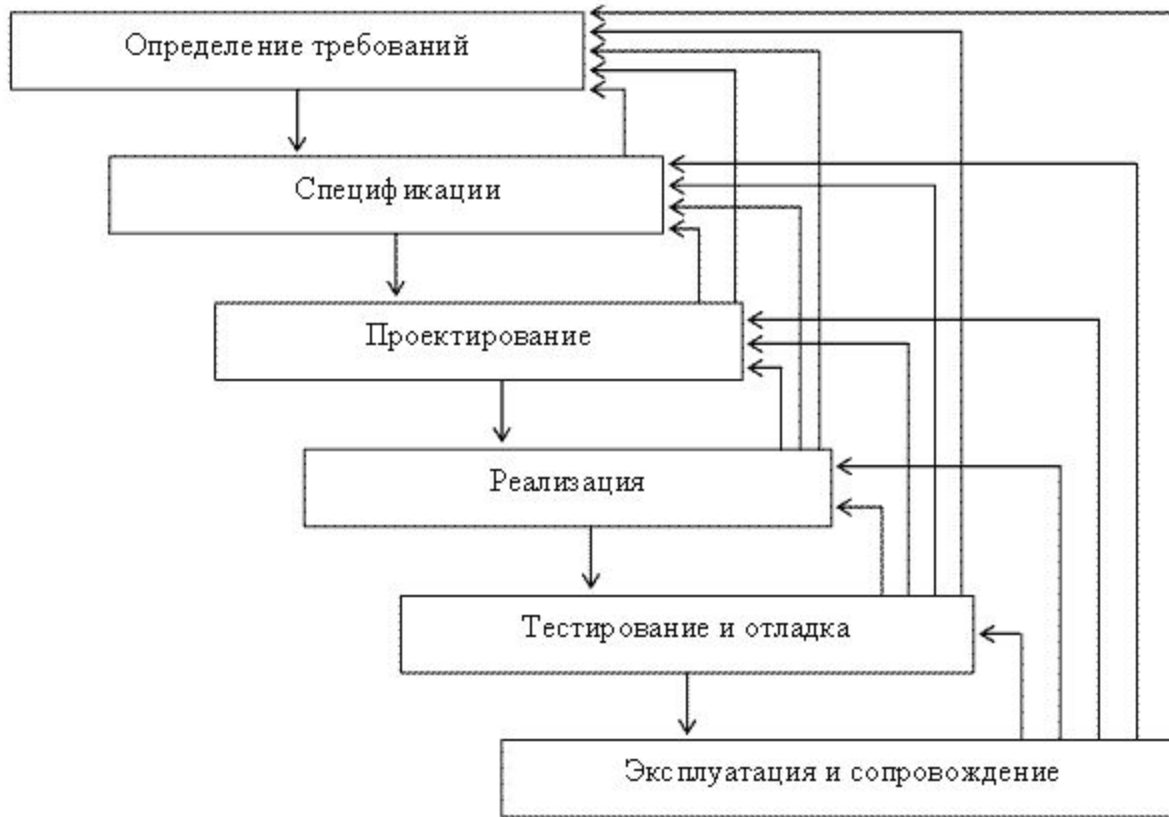
# *Фазы проекта*

**Владимир Макаров**  
**08.02.08**

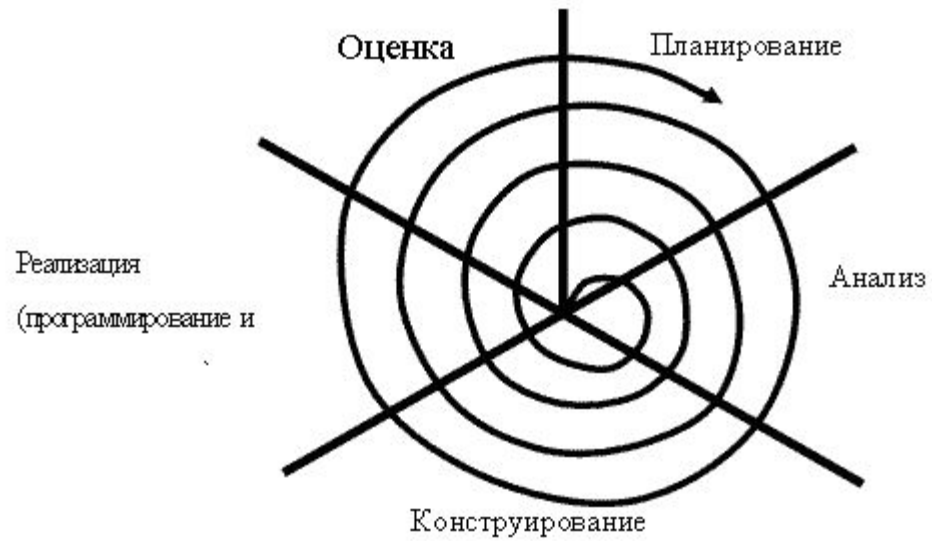
## *Выбор модели жизненного цикла*

- Каскадная модель
- Циклическая (спиральная ) модель
  
- Нужен ли прототип ?

# *Каскадная модель*



# Циклическая модель



# *Фазы проекта*

- Определение проекта (назначение и цель)
- Выработка требований
- Разработка спецификаций
- Проектирование и планирование
  - Проектирование архитектуры
  - Детальное проектирование
- Прототипирование (конструирование)
- Кодирование
- Интеграция
- Тестирование (сертификация)
- Внедрение
- Эксплуатация (сопровождение).

# *Определение проекта*

- Цели и задачи
- Назначение
- Замысел системы в окончательном варианте

# *Выработка требований*

Требования для решения *проблемы* или достижения цели.

При описании требований используются понятия качества, анализируется использование системы в конкретной среде.

# *Спецификации*

Спецификация - это описание на языке разработчика внешних характерных особенностей поведения системы.

Спецификация включает в себя:

- граничные условия
- описание функции (что программа должна делать, что возможно будет делать)
- спецификацию входных и выходных данных
- верификационные требования (установление тестовых случаев)
- тип и количество документов.



# *Проектирование*

Входной информацией для проектирования являются спецификации, написанные по требованиям пользователя.

Два вида проектирования: архитектурное и детальное проектирование.

# *Проектирование архитектуры*

Первая стадия проектирования, заканчивается декомпозицией спецификаций в структуру системы.

Спецификации на *модуль*:

- имя и цели
- неформальное описание
- ССЫЛКИ
- ТОЧКИ ВХОДОВ И ВЫХОДОВ
- алгоритмы
- комментарии

# *Проектирование архитектуры*

- Имя и цель. Что делает модуль в целевом смысле.
- Неформальное описание. обзор действий модуля и над какими данными.
- Ссылки. С кем работает модуль ( дает и принимает)
- Точки входов и выходов.
- Алгоритм. Алгоритм должен быть структурирован.
- Комментарии . Например, временные характеристики, необычные ситуации, приводящие к ошибкам.

## *Детальное проектирование*

Системная структура программы трансформируется в процедурное описание (логику) программы.

Происходит выбор и *оценка* алгоритма для реализации каждого модуля. Все детали и решения по каждому модулю должны быть хорошо определены.

# *Прототипирование*

**Прототип** - это целостный объект, содержащий в себе реализованную базовую функциональность отдельного элемента системы или системы целиком.

**Прототипирование** *важный* этап разработки. На этапе создается работающая система (м.б. неэффективно, с ошибками, и не в полной мере). Во время прототипирования разработчики начинают «видеть», как система должна быть устроена.

Быстрая «черновая» реализация базовой функциональности проекта как для анализа ее работы в целом, так и отдельно взятых ее элементов.

# *Реализация*

- Кодирование
- Интеграция
- Тестирование (сертификация)

# *Внедрение*

Интеграция системы в реальную среду.

Подтверждение соответствия требованиям.

# Эксплуатация

Оценка работы системы и поддержание ее работы в приемлемых границах.

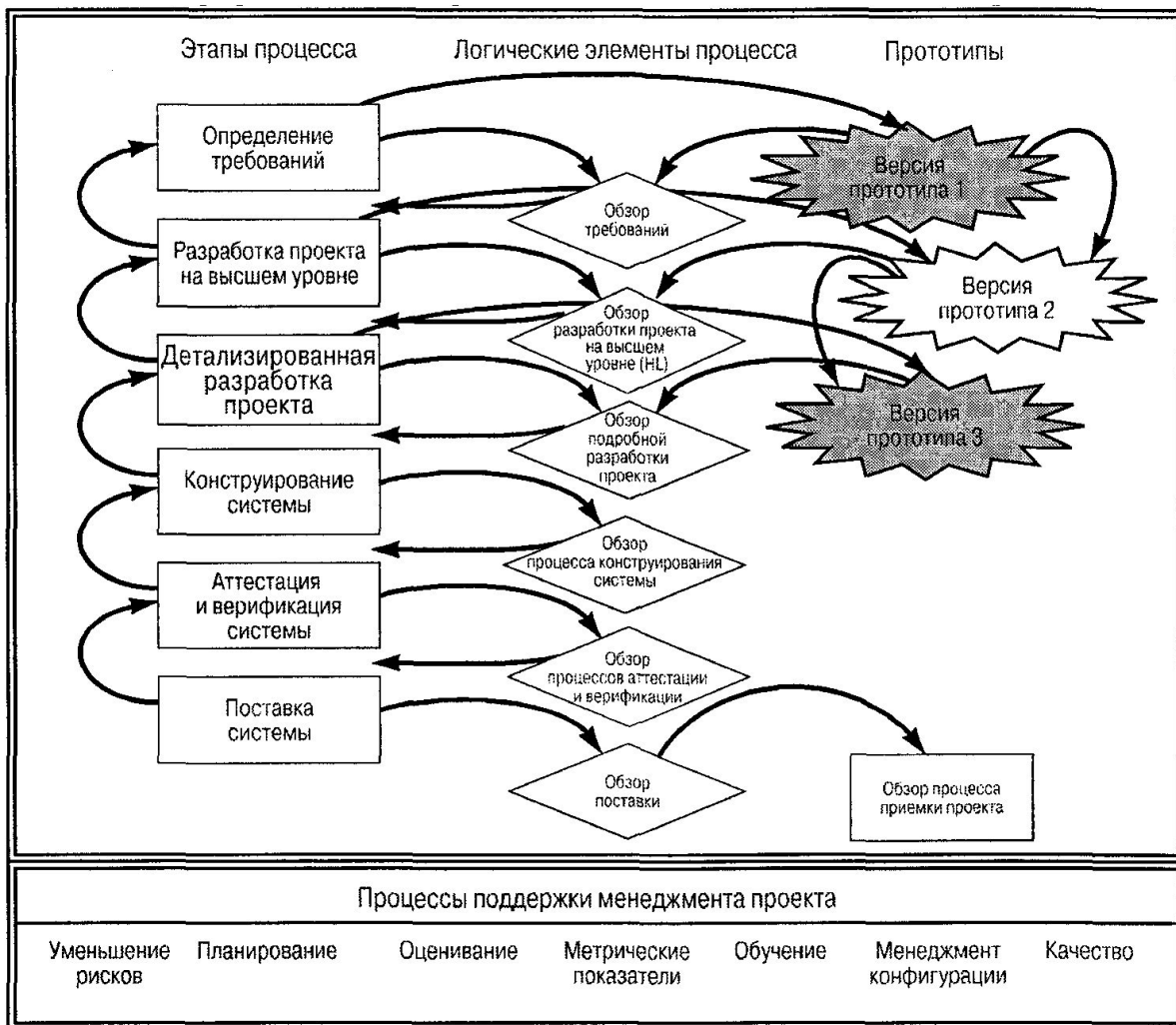
Суть процесса сопровождения:

- нахождении и исправлении ошибок
- добавлении новых функций и модификацией существующих
- включение программы в новую систему
- улучшении показателей работы.

Сопровождение программы может стоить в 2-3 раза *дороже*, чем ее разработка.



# Модель процесса разработки



## *Статистика из опыта*

- Определение требований  
- 27 %
- Проектирование высокого уровня  
- 15 %
- Детальное проектирование  
- 3 %
- Кодирование и отладка  
- 36 %
- Тестирование и верификация  
- 10 %
- Внедрение системы  
- 9 %

# *Прибыльные задачи*

- Определение требований
- Разработка проекта
- Документирование
- Поставка
- Установка и внедрение
- Обновление

# *Неприбыльные задачи 1*

- Тестирование адекватности требований
- Планирование
- Организация процесса разработки
- Обучение
- Менеджмент

## *Неприбыльные задачи 2*

- Корректировка проекта
- Задержки планов
- Анализ ошибок
- Дополнительная бумажная работа
- Реализация дополнительных функций
- Установка среды проектирование
- Определение метрик процесса
- Модификация и инсталлирование системы
- Повторное проектирование
- Повторное тестирование
- Другие внеплановые работы

## *Итог*

- Прибыльные задачи 65 – 70 %
- Неприбыльные задачи 30 – 35 %