

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

## **Лекция**

по дисциплине

*«Основы программной инженерии»*

№1 Жизненный цикл ПО

**Выполнил:** Батаргин Егор Александрович

**Группа:** Р3332

**ИТМО.ID:** 335189

**г. Санкт-Петербург, 2024 г.**

## Оглавление

Характеристики современного ПО.....	3
Жизненный цикл ПО.....	3
Модели жизненного цикла .....	3
Методология разработки (Водопадная) .....	4
Методология Ройса.....	4
Методология V-образная 1977 года .....	4
Многопроходная модель .....	4
Модель прототипирования .....	4
Методология RAD (Casесистемы).....	5
Видеолекция.....	5

## Характеристики современного ПО

- Сложность
  - API
  - Алгоритмы и методы
  - Неспособность одного человека удерживать все детали в голове
- Необходимость реализовать «вчера»
  - Требование бизнеса быть в конкурентной среде
- Низкий уровень повторного использования ПО
- Необходимость интеграции с внешними системами
- Распределенная и неоднородная среда функционирования

## Жизненный цикл ПО

Под жизненным циклом ПО подразумевается время от идеи до вывода из эксплуатации.

Этапы ЖЦ:

- Разработка требований
- Анализ
- Проектирование
- Разработка
- Тестирование
- Внедрение
- Эксплуатация
- Вывод из эксплуатации

## Модели жизненного цикла

Модель жизненного цикла - это структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении всего жизненного цикла

- Последовательная
  - Определены все требования и один этап разработки
- Инкрементная
  - Определены все требования и несколько этапов
- Эволюционная
  - Определены не все требования и несколько этапов
- Формальных преобразований

## Методология разработки (Водопадная)

Разработана в 60-х годах. Водопадная модель называется «Каскадная».

Этапы

- Требования к системы
- Требования к частям ПО
- Анализ требований
- Кодирование
- Тестирование
- Использование системы

Минус в том, что лишь на этапе тестирования вы сможете понять, что вы делаете то, что вам не надо. А сил уже затрачено. Иногда определить требования непросто. Часто бывает, что заказчик знает, что хочет, но ничего не знает о реализации.

## Методология Ройса

Ройс выдвинул идею о вовлечении пользователя в разработку

Этапы

- Предварительный дизайн (Параллельно основным процессам, сделать процесс для создания прототипа)
- Документирование
- Тестирование
- Подключение пользователя

## Методология V-образная 1977 года

Предложение было таким, чтобы на против каждой стадии из каскадной модели ставить тестирование. Для начала тестирования, надо определить корректное поведение программы. Здесь присутствуют два вида тестирования. Динамическое и статическое тестирование. Динамическое тестирование включает компьютерное исполнение тестов, а статическое – выполняет проверку артефактов без их компьютерного исполнения

## Многопроходная модель

Представляет собой V-образную модель, но разработка и интеграция происходила в несколько этапов

## Модель прототипирования

Делают быстрые прототипы для пользователи с тем, что он получит. Пока, пользователь не одобрит. То есть идет создание базы данных, анализ и

разработка основного функционала. Затем это все интегрируется в прототип и

## Спиральная методология

Идея в этой методологии заключается в том, что каждая спираль — это новая фаза разработки ПО. И на каждой фазе создается свой прототип. И после каждого прототипа проводится анализ риска, проектирование. На последней фазе идет окончательный прототип, кодирование, тестирование, выпуск и сопровождение.

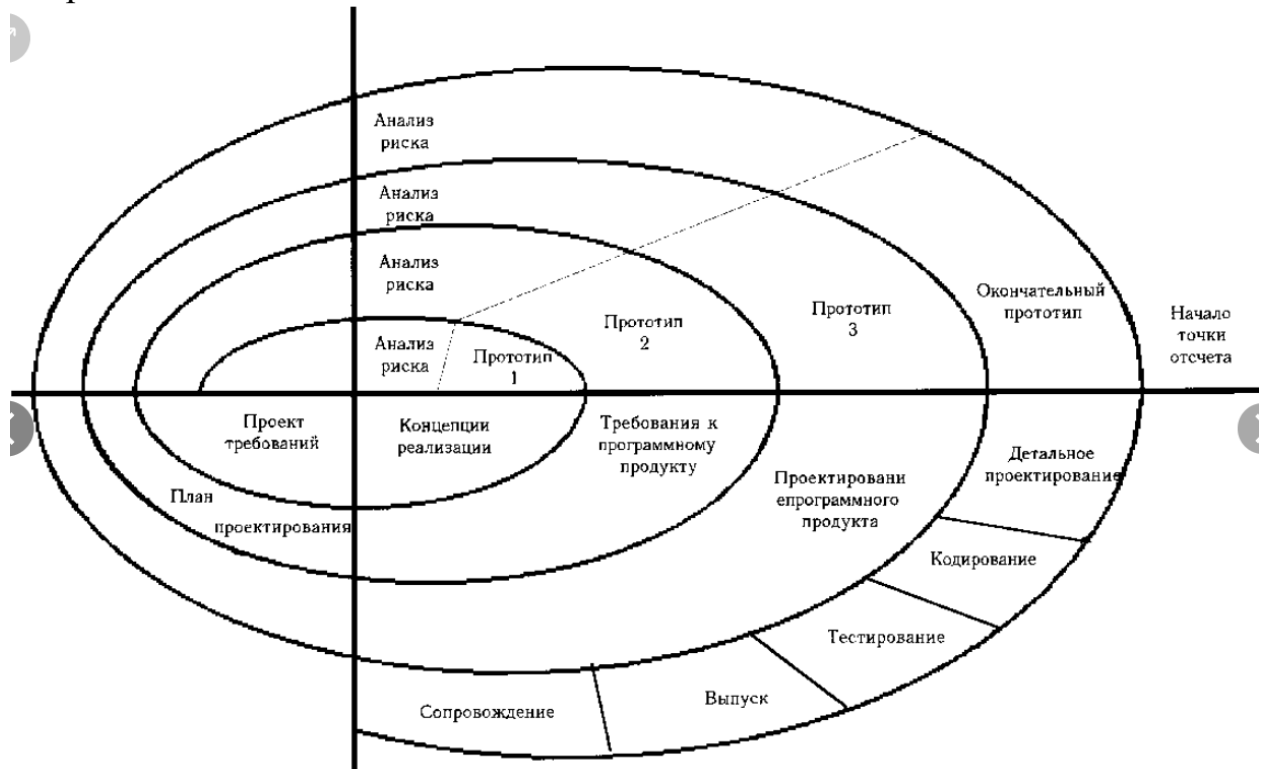


Рис. 1.8. Модель "спираль"

## Методология RAD (Cасесистемы)

Описана компанией IBM.

Идея заключается в том, что в разработку вовлекают пользователя. Как я понял, создается заранее прототипы и показываются пользователям. Все ПО собирается из стандартных блоков.

CASE-системы. Позволяют строить модели и превращать в заготовки кода и обрабатывать требования. Они активно используются в RAD

## Видеолекция

Видеолекции есть на двух платформах:

- [RuTube](#)
- [YouTube](#)

СПАСИБО ЗА ВНИМАНИЕ !