

# Жизненный цикл программного обеспечения

---

Выполнено студентами: Зиннуров Эмиль,  
Умиленов Богдан

Группа: 20П-3

# Жизненный цикл

Жизненный цикл программного обеспечения охватывает промежуток времени с момента возникновения потребности в ПО до вывода его из эксплуатации  
В зависимости от используемой модели жизненный цикл протекать может по-разному

Модели отличаются между собой по таким параметрам как:

- этапность (фазы, стадии, этапы)
- последовательность прохождения этапов (линейная или циклическая)
- гибкость (возможность подстраивать процесс под конкретные условия)
- связь с определёнными методологиями разработки ПО
- использование специализированных инструментальных средств
- другие

# Модель жизненного цикла программного обеспечения

структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

Эти модели можно разделить на 3 основных группы:

- Инженерный подход

- С учетом специфики задачи

- Современные технологии быстрой разработки

# Методологии разработки ПО

Методология разработки ПО – это совокупность принципов, идей, понятий, методов, способов и средств, определяющих стиль разработки ПО.

Выбор методологии зависит от:  
размера команды;  
специфики и сложности проекта;  
стабильности процессов в компании  
личных качеств сотрудников.

# Модель кодирования и устранения ошибок

Совершенно простая модель, характерная для студентов ВУЗов. Именно по этой модели большинство студентов разрабатывают, ну скажем лабораторные работы.

Данная модель имеет следующий алгоритм:

Постановка задачи

Выполнение

Проверка результата

При необходимости переход к первому пункту

Алгоритм данного метода, который я привожу на схеме, имеет ряд преимуществ перед алгоритмом предыдущей модели, но также имеет и ряд весомых недостатков.

### Алгоритм каскадной модели



#### Преимущества:

Последовательное выполнение этапов проекта в строгом фиксированном порядке  
Позволяет оценивать качество продукта на каждом этапе

#### Недостатки:

Отсутствие обратных связей между этапами  
Не соответствует реальным условиям разработки программного продукта

## Каскадная модель с промежуточным контролем (водоворот)

Данная модель является почти эквивалентной по алгоритму предыдущей модели, однако при этом имеет обратные связи с каждым этапом жизненного цикла, при этом порождает очень весомый недостаток: 10-ти кратное увеличение затрат на разработку. Относится к первой группе моделей.

## V модель (разработка через тестирование)

Данная модель имеет более приближенный к современным методам алгоритм, однако все еще имеет ряд недостатков. Является одной из основных практик экстремального программирования.





# Модель на основе разработки прототипа

Данная модель основывается на разработки прототипов и прототипирования продукта.

Прототипирование используется на ранних стадиях жизненного цикла программного обеспечения:

Прояснить не ясные требования (прототип UI)

Выбрать одно из ряда концептуальных решений (реализация сценариев)

Проанализировать осуществимость проекта

Классификация прототипов:

- Горизонтальные и вертикальные

- Одноразовые и эволюционные

- бумажные и раскадровки

Горизонтальные прототипы — моделирует исключительно UI не затрагивая логику обработки и базу данных.

Вертикальные прототипы — проверка архитектурных решений.

Одноразовые прототипы — для быстрой разработки.

Эволюционные прототипы — первое приближение эволюционной системы.

# Спиральная модель жизненного цикла программного обеспечения

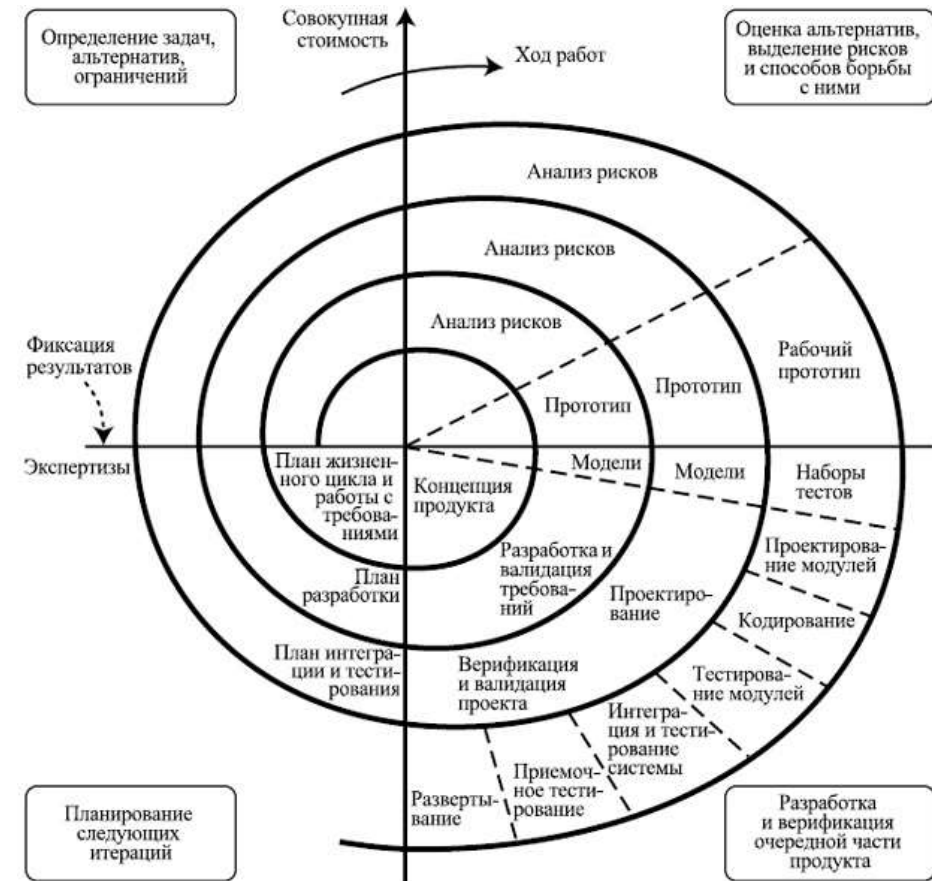
Спиральная модель представляет собой процесс разработки программного обеспечения, сочетающий в себе как проектирование, так и поэтапное прототипирование с целью сочетания преимуществ восходящей и нисходящей концепции.

Преимущества:

- Быстрое получение результата
- Повышение конкурентоспособности
- Изменяющиеся требования — не проблема

Недостатки:

- Отсутствие регламентации стадий



# «Agile Model» (гибкая методология разработки)

## Преимущества:

подходит для больших или нацеленных на длительный жизненный цикл проектов, постоянно адаптируемых к условиям рынка;  
лучше всего подходит для руководителей, которым свойственно генерировать, выдавать и опробовать новые идеи еженедельно или даже ежедневно;  
после каждой итерации заказчик может наблюдать результат и понимать, удовлетворяет он его или нет.

## Недостатки:

из-за отсутствия конкретных формулировок результатов сложно оценить трудозатраты и стоимость, требуемые на разработку.

