

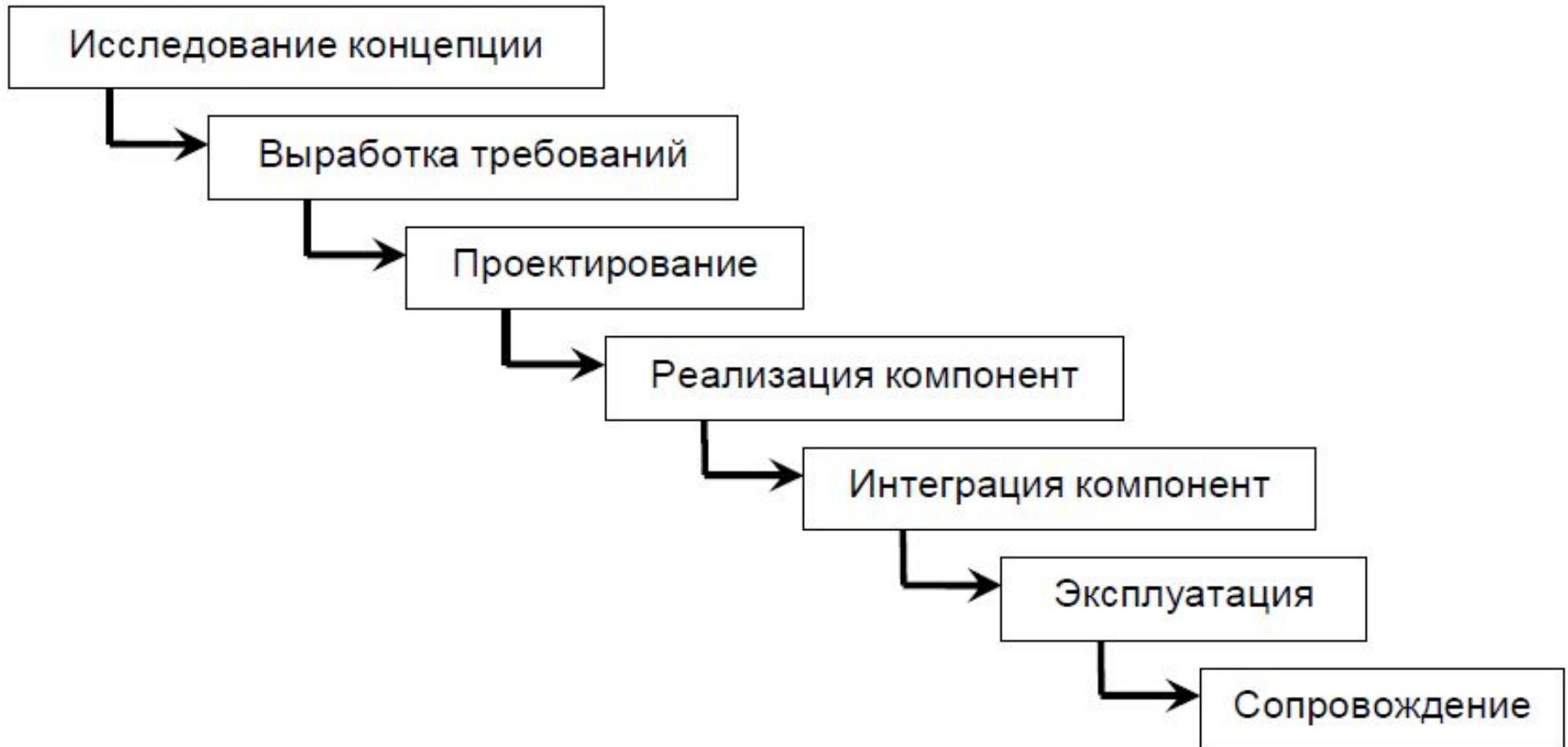
Модели жизненного цикла ПП

Модели жизненного цикла разработки программного продукта

Под моделью жизненного цикла разработки ПП понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении жизненного цикла разработки ПП. Модель жизненного цикла зависит от специфики и сложности выполняемого проекта, а также от условий, в которых создается и будет функционировать ПП. Модель жизненного цикла любого конкретного ПП определяет характер процесса его создания.

- каскадная модель
- Спиральная
- Итерационная
- V-образная
- Инкрементная
- Модель быстрого прототипирования

Каскадная модель жизненного цикла ПО



Каскадная модель включает выполнение следующих фаз:

- **исследование концепции** — происходит исследование требований, разрабатывается видение продукта и оценивается возможность его реализации.
- **определение требований** — определяются программные требования для информационной предметной области системы, предназначение, линии поведения, производительность и интерфейсы.
- **разработка проекта** — разрабатывается и формулируется логически последовательная техническая характеристика программной системы, включая структуры данных, архитектуру ПО, интерфейсные представления и процессуальную (алгоритмическую) детализацию;

- **реализация** — эскизное описание ПО превращается в полноценный программный продукт. Результат: исходный код, база данных и документация. В реализации обычно выделяют два этапа: реализацию компонент ПО и интеграцию компонент в готовый продукт. На обоих этапах выполняется кодирование и тестирование, которые тоже иногда рассматривают как два подэтапа.
- **эксплуатация и поддержка** - подразумевает запуск и текущее обеспечение, включая предоставление технической помощи, обсуждение возникших вопросов с пользователем, регистрацию запросов пользователя на модернизацию и внесение изменений, а также корректирование или устранение ошибок;
- **сопровождение** — устранение программных ошибок, неисправностей, сбоев, модернизация и внесение изменений. Состоит из итераций разработки.

Основными принципами каскадной модели являются:

1. Строго последовательное выполнение фаз

-Каждая последующая фаза начинается лишь тогда, когда полностью

завершено выполнение предыдущей фазы

-Каждая фаза имеет определенные критерии входа и выхода:

входные и

выходные данные.

-Каждая фаза полностью документируется

-Переход от одной фазы к другой осуществляется посредством формального обзора с участием заказчика

2. Основа модели – сформулированные требования (ТЗ), которые меняться не должны

3. Критерий качества результата – соответствие продукта установленным требованиям.

Спиральная модель жизненного цикла ПО



На практике, при решении достаточно большого количества задач, разработка ПО имеет циклический характер, когда после выполнения некоторых стадий приходится возвращаться на предыдущие. Можно указать две основные причины таких возвратов:

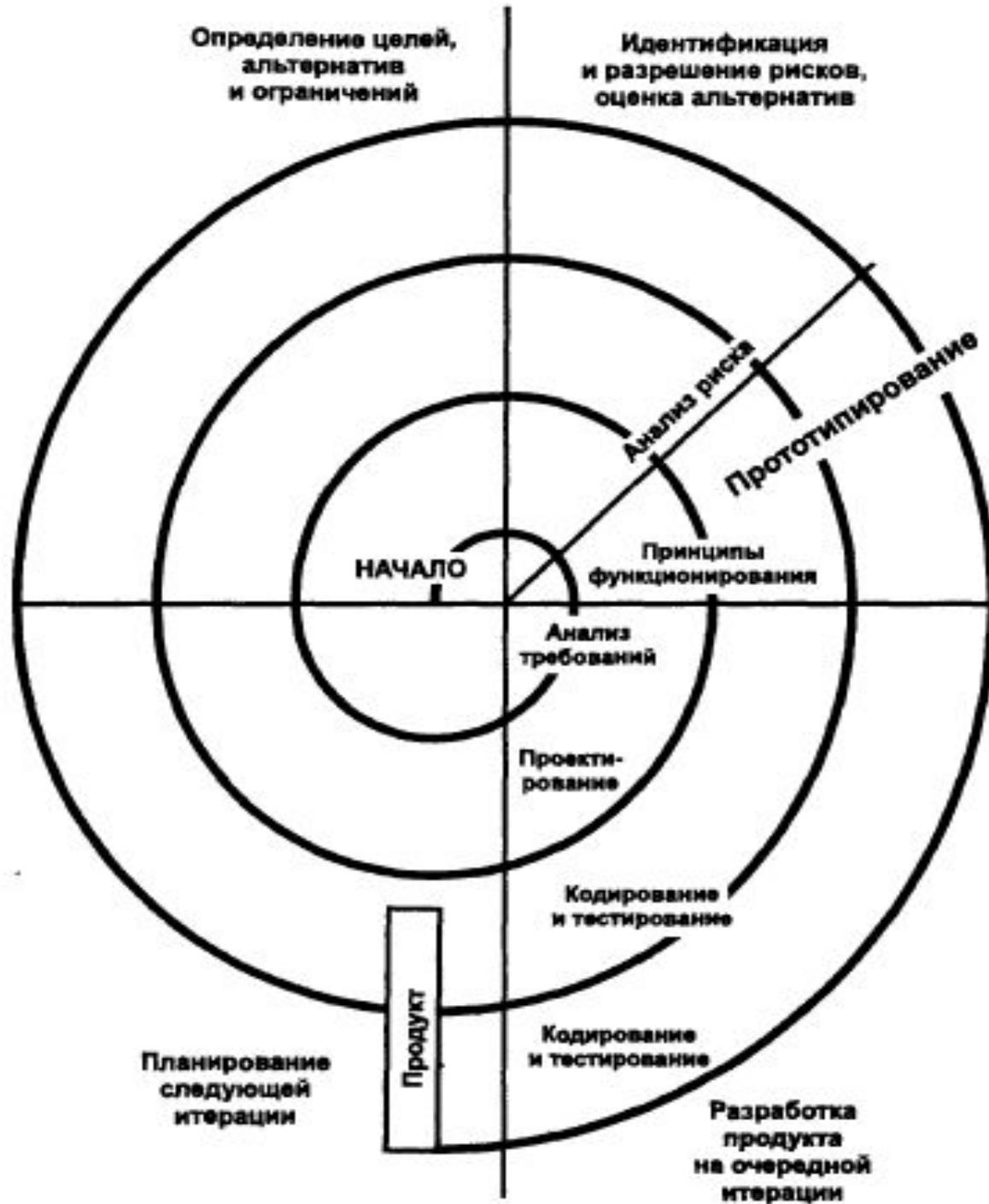
- Ошибки разработчиков, допущенные на ранних стадиях и выявленные на поздних стадиях – ошибки анализа, проектирования, кодирования, выявляемые, как правило, на стадии тестирования.
- Изменение требований в процессе разработки («ошибки» заказчиков). Это или неготовность заказчиков сформулировать требования («Сказать, что должна делать программа я смогу только после того, как увижу как она работает»), или изменения требований, вызванные изменениями ситуации в процессе разработки (изменения рынка, новые технологии, ...).

Основные принципы спиральной модели можно сформулировать следующим образом:

- Разработка вариантов продукта, соответствующих различным вариантам требований с возможностью вернуться к более ранним вариантам
- Создание прототипов ПО как средства общения с заказчиком для уточнения и выявления требований
- Планирование следующих вариантов с оценкой альтернатив и анализом рисков, связанных с переходом к следующему варианту
- Переход к разработке следующего варианта до завершения предыдущего в случае, когда риск завершения очередного варианта (прототипа) становится неоправданно высок.
- Использование каскадной модели как схемы разработки очередного варианта
- Активное привлечение заказчика к работе над проектом. Заказчик участвует в оценке очередного прототипа ПО, уточнении требований при переходе к следующему, оценке предложенных альтернатив очередного варианта и оценке рисков.

Разработка вариантов продукта представляется как набор циклов раскручивающейся спирали. Каждому циклу спирали соответствует такое же количество стадий, как и в модели каскадного процесса. При этом, начальные стадии, связанные с анализом и планированием представлены более подробно с добавлением новых элементов. В каждом цикле выделяются четыре базовые фазы:

- определение целей, альтернативных вариантов и ограничений.
- оценка альтернативных вариантов, идентификация и разрешение рисков.
- разработка продукта следующего уровня.
- планирование следующей фазы.



Определение целей,
альтернатив
и ограничений

Идентификация
и разрешение рисков,
оценка альтернатив

НАЧАЛО

Принципы
функционирования

Анализ
требований

Проекти-
рование

Кодирование
и тестирование

Кодирование
и тестирование

Разработка
продукта
на очередной
итерации

Планирование
следующей
итерации

Продукт

Анализ риска

Прототипирование

Другие типы моделей жизненного цикла ПО

Итерационная модель

Итерационная модель жизненного цикла является развитием классической каскадной модели и предполагает возможность возвратов на ранее выполненные этапы. При этом, причинами возвратов в классической итерационной модели являются выявленные ошибки, устранение которых и требует возврата на предыдущие этапы в зависимости от типа (природы) ошибки – ошибки кодирования, проектирования, спецификации или определения требований. Реально итерационная модель является более жизненной, чем классическая (строгая) каскадная модель, т.к. создание ПО всегда связано с устранением ошибок.

V-образная модель

V-образная модель была создана как итерационная разновидность каскадной модели.

Целями итераций в этой модели является обеспечение процесса тестирования. Тестирование продукта обсуждается, проектируется и планируется на ранних этапах жизненного цикла разработки. План испытания приемки заказчиком разрабатывается на этапе планирования, а компоновочного испытания системы - на фазах анализа, разработки проекта и т. д. Этот процесс разработки планов испытания обозначен пунктирной линией между прямоугольниками V-образной модели. Помимо планов, на ранних этапах разрабатываются также и тесты, которые будут выполняться при завершении параллельных этапов.

Инкрементная (пошаговая) модель

Инкрементная разработка представляет собой процесс поэтапной реализации всей системы и поэтапного наращивания (приращения) функциональных возможностей. На первом шаге необходим полный заранее сформулированный набор требований, которые делятся по некоторому признаку на части. Далее выбирается первая группа требований и выполняется полный проход по каскадной модели. После того, как первый вариант системы, выполняющий первую группу требований сдан заказчику, разработчики переходят к следующему шагу (второму инкременту) по разработке варианта, выполняющего вторую группу требований.

Особенностью инкрементной модели является разработка приемочных тестов на этапе анализа требований, что упрощает приемку варианта заказчиком и устанавливает четкие цели разработки очередного варианта системы. Инкрементная модель особенно эффективна в случае, когда задача разбивается на несколько относительно независимых подзадач (разработка подсистем «Зарплата», «Бухгалтерия», «Склад», «Поставщики»).

Модель быстрого прототипирования

Модель быстрого протитипирования предназначена для быстрого создания прототипов продукта с целью уточнения требований и поэтапного развития прототипов в конечный продукт. Скорость (высокая производительность) выполнения проекта обеспечивается планированием разработки прототипов и участием заказчика в процессе разработки.

Начало жизненного цикла разработки помещено в центре эллипса. Совместно с пользователем разрабатывается предварительный план проекта на основе предварительных требований. Результат начального планирования - документ, описывающий в общих чертах примерные графики и результативные данные.

Следующий уровень – создание исходного прототипа на основе быстрого анализа, проекта база данных, пользовательского интерфейса и некоторых функций. Затем начинается итерационный цикл быстрого прототипирования. Разработчик проекта демонстрирует очередной прототип, пользователь оценивает его функционирование, совместно определяются проблемы и пути их преодоления для перехода к следующему прототипу. Этот процесс продолжается до тех пор, пока пользователь не согласится, что очередной прототип в точности отображает все требования.

Получив одобрение пользователя, быстрый прототип преобразуют в детальный проект, и систему настраивают на производственное использование. Именно на этом этапе настройки ускоренный прототип становится полностью действующей системой.