

# Жизненный цикл программного обеспечения

Авторы: Ефименкова Екатерина,  
Яковлев Виталий 20П-3

# Введение

**Жизненный цикл (ЖЦ) ПО** – это период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации. Жизненный цикл ПО включает следующие **стадии**:

- Анализ.
- Проектирование.
- Программирование.
- Тестирование и отладка.
- Эксплуатация.



Каждый процесс разделен на набор действий, каждое действие — на набор задач. Каждый процесс, действие или задача инициируется и выполняется другим процессом по мере необходимости, причем не существует заранее определенных последовательностей выполнения (естественно, при сохранении связей по входным данным).

# Основные этапы ЖЦ ПО

**Процесс приобретения** (acquisition process). Он состоит из действий и задач заказчика, приобретающего ПО. Данный процесс охватывает следующие действия:

- инициирование приобретения;
- подготовку заявочных предложений;
- подготовку и корректировку договора;
- надзор за деятельностью поставщика;
- приемку и завершение работ.

**Процесс поставки** (supply process). Он охватывает действия и задачи, выполняемые поставщиком, который снабжает заказчика программным продуктом или услугой. Данный процесс включает следующие действия:

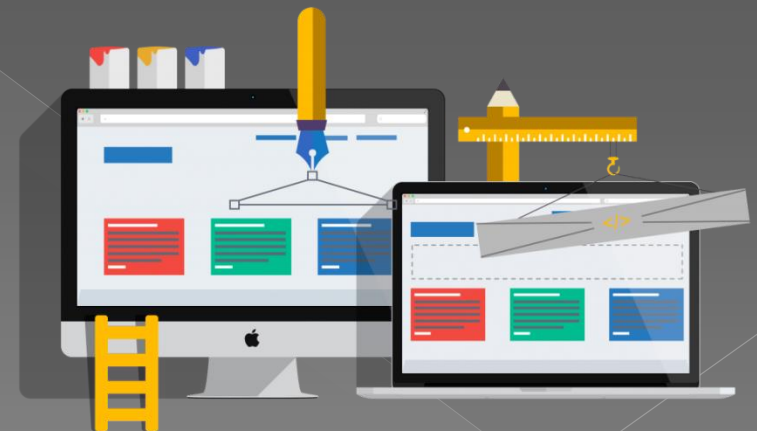
- инициирование поставки;
- подготовку ответа на заявочные предложения;
- подготовку договора;
- планирование;
- выполнение и контроль;
- проверку и оценку;
- поставку и завершение работ.



# Основные этапы ЖЦ ПО

**Процесс разработки** (development process). Он предусматривает действия и задачи, выполняемые разработчиком, и охватывает работы по созданию ПО и его компонентов в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала, и т. д. Процесс разработки включает следующие действия:

- подготовительную работу;
- анализ требований к системе;
- проектирование архитектуры системы;
- анализ требований к ПО;
- проектирование архитектуры ПО;
- детальное проектирование ПО;
- кодирование и тестирование ПО;
- интеграцию ПО;
- квалификационное тестирование ПО;
- интеграцию системы;
- квалификационное тестирование системы;
- установку ПО;
- приемку ПО.



# Основные этапы ЖЦ ПО

**Процесс эксплуатации** (operation process). Он охватывает действия и задачи оператора — организации, эксплуатирующей систему. Данный процесс включает следующие действия:

- подготовительную работу;
- эксплуатационное тестирование;
- эксплуатацию системы;
- поддержку пользователей.



**Процесс сопровождения** (maintenance process). Он предусматривает действия и задачи, выполняемые сопровождающей организацией (службой сопровождения). Данный процесс активизируется **при** изменениях (модификациях) программного продукта и соответствующей документации, вызванных возникшими проблемами или потребностями в модернизации либо адаптации ПО. В соответствии со стандартом IEEE-90 под сопровождением понимается внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

- Изменения, вносимые в существующее ПО, не должны нарушать его целостность. Процесс сопровождения включает перенос ПО в другую среду (миграцию) и заканчивается снятием ПО с эксплуатации.
- Процесс сопровождения охватывает следующие действия:
- подготовительную работу;
- анализ проблем и запросов на модификацию ПО;
- модификацию ПО;
- проверку и приемку;
- перенос ПО в другую среду;
- снятие ПО с эксплуатации.

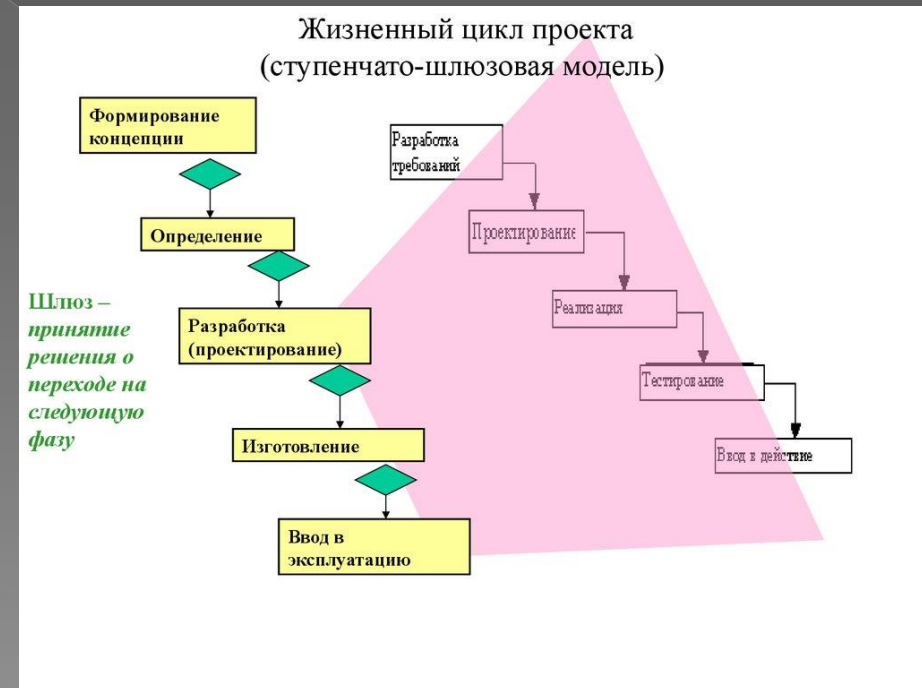


# Модели жизненного цикла ПО

Под моделью ЖЦ ПО понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении ЖЦ. Модель ЖЦ зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует.

Модель ЖЦ любого конкретного ПО ЭИС определяет характер процесса его создания, который представляет собой совокупность упорядоченных во времени, взаимосвязанных и объединенных в стадии работ, выполнение которых необходимо и достаточно для создания ПО, соответствующего заданным требованиям. Под стадией создания ПО понимается часть процесса создания ПО, ограниченная некоторыми временными рамками и заканчивающаяся выпуском конкретного продукта (моделей ПО, программных компонентов, документации), определяемого заданными для данной стадии требованиями. Стадии создания ПО выделяются по соображениям рационального планирования и организации работ, заканчивающихся заданными результатами. В состав жизненного цикла ПО обычно включаются следующие стадии:

- Формирование требований к ПО.
- Проектирование.
- Реализация.
- Тестирование.
- Ввод в действие.
- Эксплуатация и сопровождение.
- Снятие с эксплуатации.



# Стадия формирования требований к ПО.

Она является одной из важнейших, поскольку определяет успех всего проекта. Данная стадия включает следующие этапы:

- *планирование работ*, предваряющее работы над проектом. Основными задачами этапа являются: определение целей разработки, предварительная экономическая оценка проекта, построение плана-графика выполнения работ, создание и обучение совместной рабочей группы;
- *проведение обследования деятельности автоматизируемого объекта (организации)*, в рамках которого осуществляются: предварительное выявление требований к будущей системе; определение структуры организации; определение перечня целевых функций организации; анализ распределения функций по подразделениям и сотрудникам; выявление функциональных взаимодействий между подразделениями, информационных потоков внутри подразделений и между ними, внешних по отношению к организации объектов и внешних информационных взаимодействий; анализ существующих средств автоматизации деятельности организации;
- *построение моделей деятельности организации*, предусматривающее обработку материалов обследования и построение двух видов моделей:
- модели "AS-IS" ("как есть"), отражающей существующее на момент обследования положение дел в организации и позволяющей понять, каким образом функционирует данная организация, а также выявить узкие места и сформулировать предложения по улучшению ситуации;
- модели "TO-BE" ("как должно быть"), отражающей представление о новых технологиях работы организации.





# Стадия проектирования.

- разработка системного проекта. На этом этапе дается ответ на вопрос: "Что должна делать будущая система?", а именно: определяются архитектура системы, ее функции, внешние условия функционирования, интерфейсы и распределение функций между пользователями и системой, требования к программным и информационным компонентам, состав исполнителей и сроки разработки. Основу системного проекта составляют модели проектируемой ЭИС, которые строятся на основе модели "ТО-БЕ". Документальным результатом этапа является техническое задание;
- разработка технического проекта. На этом этапе на основе системного проекта осуществляется собственно проектирование системы, включающее проектирование архитектуры системы и детальное проектирование. Таким образом, дается ответ на вопрос: "Как построить систему, чтобы она удовлетворяла предъявленным к ней требованиям?". Модели проектируемой ЭИС при этом уточняются и детализируются до необходимого уровня.



# Дополнительные модели жизненного цикла

## Каскадная модель жизненного цикла программного обеспечения (водопад)

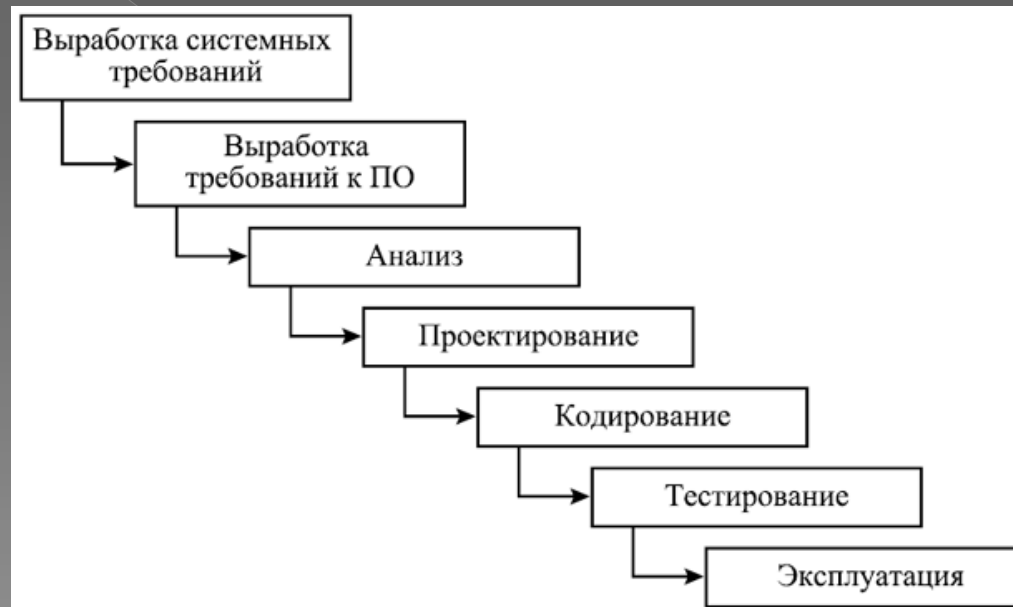
Алгоритм данного метода, который я привожу на схеме, имеет ряд преимуществ перед алгоритмом предыдущей модели, но также имеет и ряд **весомых** недостатков.

Преимущества:

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

Недостатки:

- Не соответствует реальным условиям разработки программного продукта
- Относится к первой группе моделей
- Отсутствие обратных связей между этапами

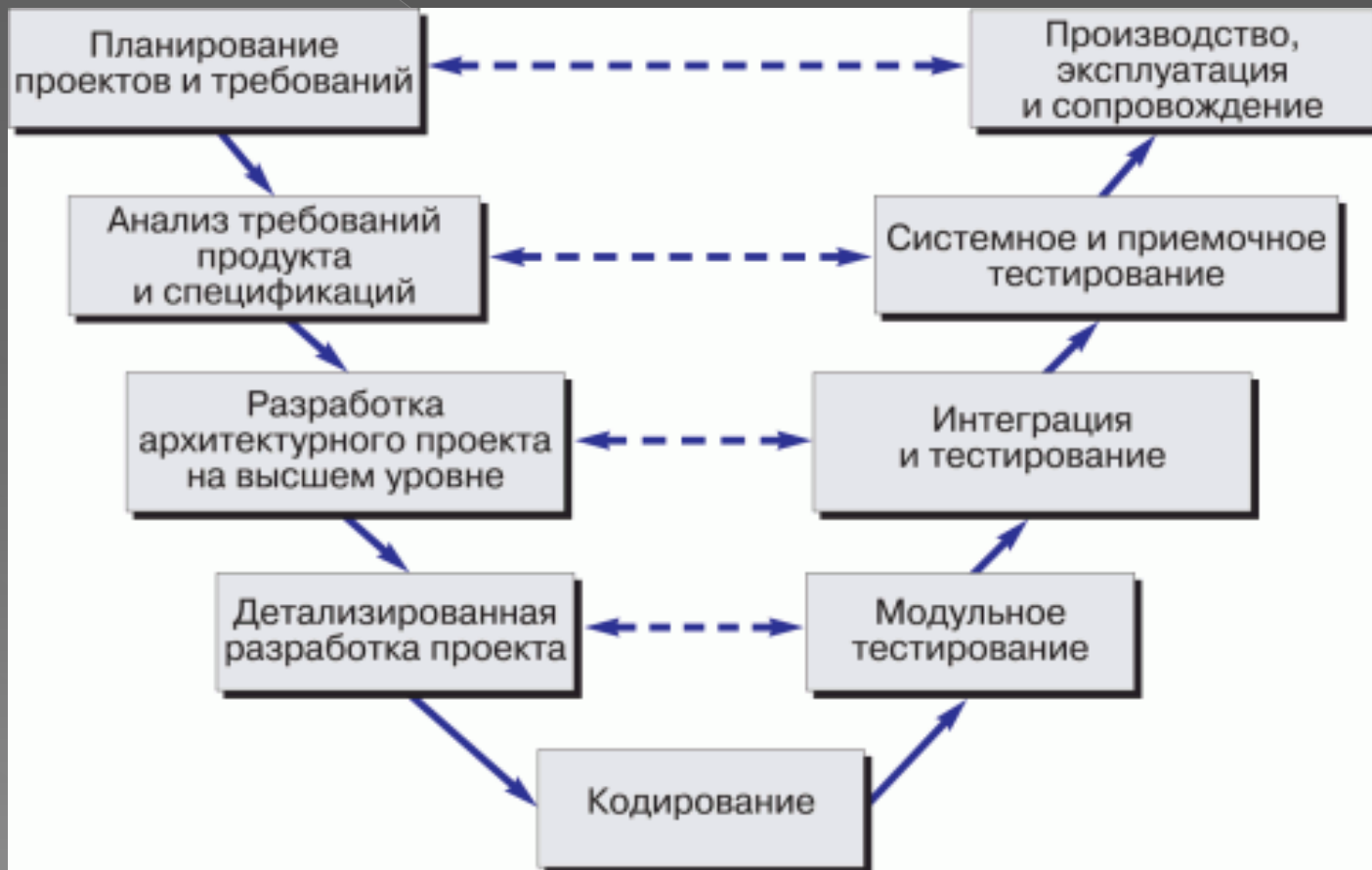


# Каскадная модель с промежуточным контролем (водоворот)

- 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.

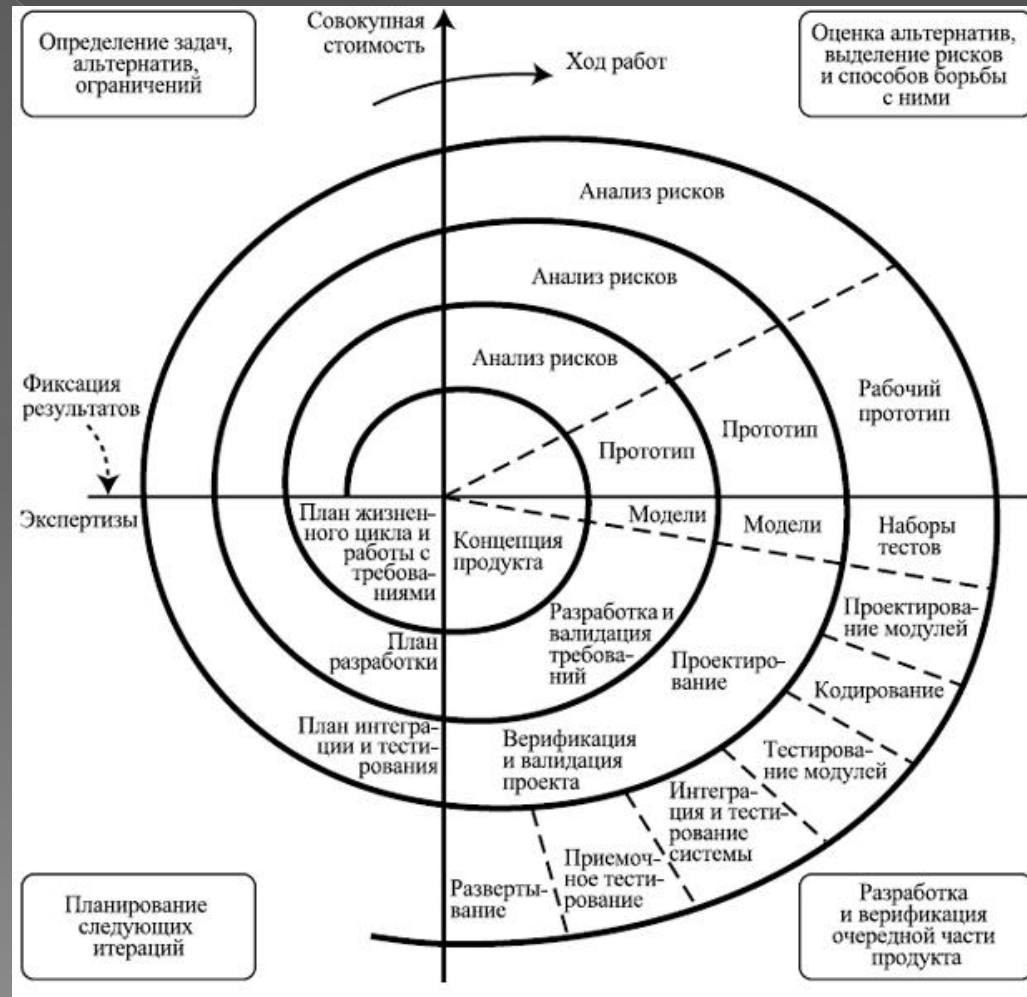
**Вертикальные** прототипы — проверка архитектурных решений.

**Одноразовые** прототипы — для быстрой разработки.

**Эволюционные** прототипы — первое приближение эволюционной системы.

Модель принадлежит второй группе.

# Спиральная модель жизненного цикла программного обеспечения



# 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

*Третьей группе принадлежат такие модели как **экстремальное***

**программирование (XP), SCRUM, инкрементальная модель (RUP),** но о них я бы хотел рассказать в отдельном топике