
ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

-
- 1. Понятие жизненного цикла программного обеспечения
 - 2. Каскадная модель жизненного цикла
 - 3. Спиральная модель жизненного цикла
 - 4. Процессы жизненного цикла и стандарты
 - 5. СТБ ИСО/МЭК 12207-2003
 - 6. ГОСТ Р ИСО/МЭК 15288-2005
 - 7. Выводы
 - 8. Контрольные вопросы
-

- Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде **жизненного цикла (ЖЦ) программного обеспечения (ПО)**, представляя его как некоторую последовательность стадий и выполняемых на них процессов.
- Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д.
- Такое формальное описание ЖЦ ПС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Жизненный цикл ПО можно представить как ряд событий, происходящих с системой в процессе ее создания и использ.

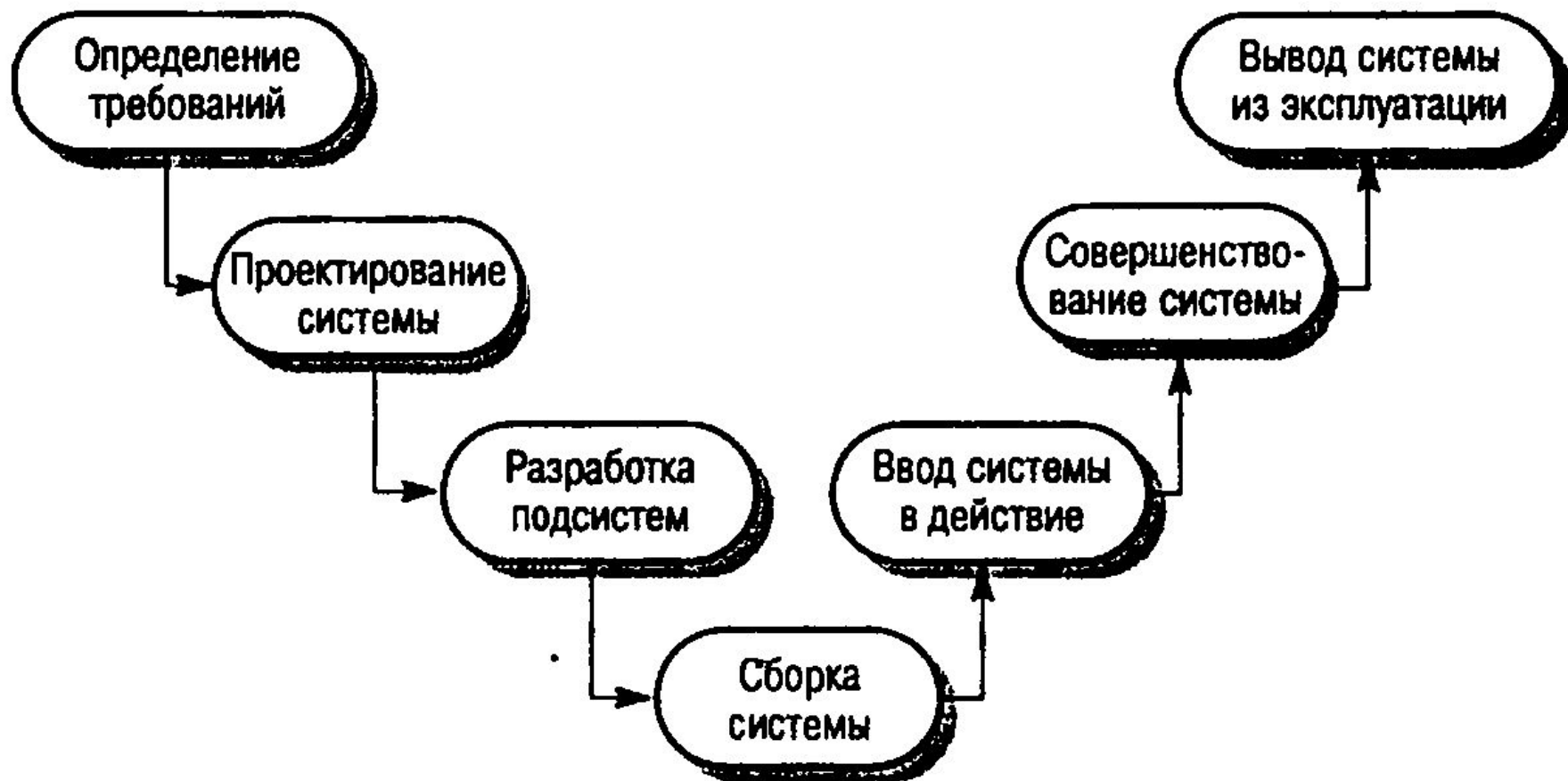


Рис. 4.1. Этапы обобщенного процесса создания системы

- ***Модель жизненного цикла*** отражает различные состояния системы, начиная с момента возникновения необходимости в данной ПО и заканчивая моментом ее полного выхода из употребления.
- ***Модель жизненного цикла*** – структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.
- В настоящее время известны и используются следующие ***модели жизненного цикла:***
 - ***Каскадная модель***
 - ***Поэтапная модель с промежуточным контролем***
 - ***Спиральная модель***

Каскадная модель (рис. 4.1) предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.



Рис. 4.1. Каскадная модель ЖЦ ПС

Поэтапная модель с промежуточным контролем (рис. 4.2).

Разработка ПО ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.



Рис. 4.2. Поэтапная модель с промежуточным контролем

Спиральная модель. На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки - анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).



Рис. 4.3. Спиральная модель ЖЦ ПС

-
- На практике наибольшее распространение получили две основные *модели жизненного цикла*:
 - ▣ *каскадная модель* (характерна для периода 1970-1985 гг.);
 - ▣ *спиральная модель* (характерна для периода после 1986 г.).
-

-
- В ранних проектах достаточно простых ПО каждое приложение представляло собой единый, функционально и информационно независимый блок.
 - Для разработки такого типа приложений **эффективным оказался каскадный способ.**
 - Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.
-

-
- Можно выделить следующие **положительные стороны применения каскадного подхода**:
 - на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
 - выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.
-

- Каскадный подход хорошо зарекомендовал себя при построении относительно простых ИС, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к системе.
- **Основным недостатком** этого подхода является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений.
- В результате реальный процесс создания ПО оказывается соответствующим *поэтапной модели с промежуточным контролем.*

-
- **Основным недостатком каскадной модели** является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений.
 - **Спиральная модель ЖЦ** была предложена для преодоления перечисленных проблем.
 - На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется **путем создания прототипов**.
 - Каждый виток спирали соответствует созданию **работоспособного фрагмента или версии** системы.
-

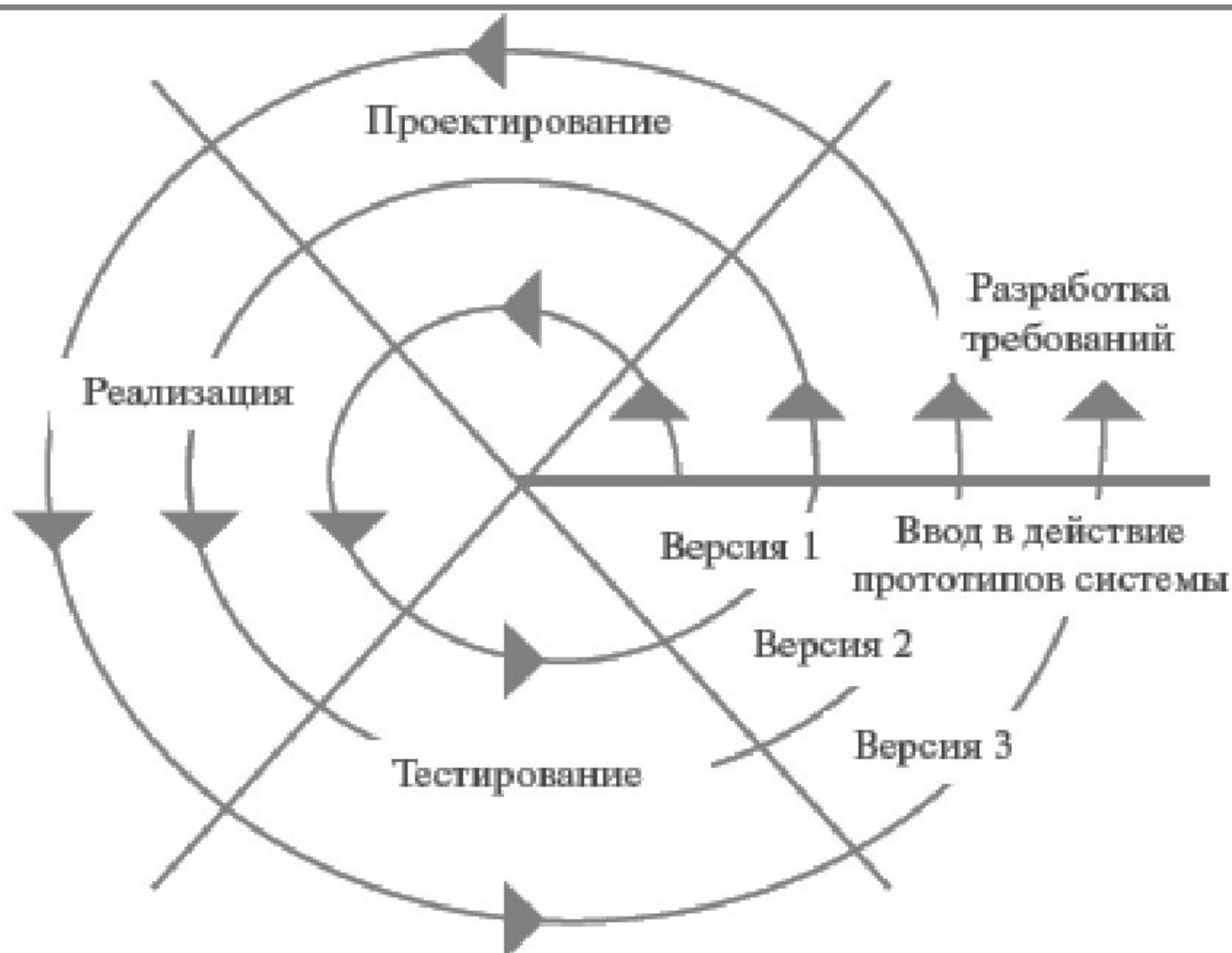


Рис. 4.3. Спиральная модель ЖЦ ПС

- Это позволяет **уточнить** требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали.
- Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается **обоснованный вариант**, который удовлетворяет действительным требованиям заказчика и доводится до реализации.
- **Итеративная разработка** отражает объективно существующий спиральный цикл создания сложных систем.
- Она позволяет **переходить на следующий этап**, не дожидаясь полного завершения работы на текущем и решить **главную задачу** - как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

-
- **Основная проблема** спирального цикла — определение момента перехода на следующий этап.
 - Для ее решения вводятся временные ограничения на каждый из этапов *жизненного цикла*, и переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена.
 - **Планирование производится** на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.
-

- Каждая из стадий создания системы предусматривает выполнение **определенного объема работ**, которые представляются в виде *процессов ЖЦ*.
- **Процесс** определяется как совокупность взаимосвязанных действий, преобразующих входные данные в выходные.
- Описание каждого процесса включает в себя перечень решаемых задач, исходных данных и результатов.
- Существует **целый ряд стандартов**, регламентирующих ЖЦ ПО, а в некоторых случаях и процессы разработки.

-
- Среди наиболее известных стандартов можно выделить следующие:
 - **1. ГОСТ 34.601-90** - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе.
 - Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют *каскадной модели* жизненного цикла.
-

-
- **ISO/IEC 12207:1995** - стандарт на процессы и организацию *жизненного цикла*.
 - Распространяется на все виды заказного ПО.
 - Стандарт не содержит описания фаз, стадий и этапов.
 - Этот стандарт более подробно рассмотрим дальше в нашей лекции.
-

-
- ❑ **СТБ ИСО/МЭК 12207-2003** Информационная технология. Процессы жизненного цикла программных систем
 - ❑ **ГОСТ Р ИСО/МЭК 12207-1999**
 - ❑ Стандарт ISO 12207 равносильно ориентирован на организацию действий каждой из двух сторон:
 - ❑ **поставщик** (разработчик) и **покупатель** (пользователь);
 - ❑ может быть в равной степени применен, когда обе стороны – из одной организации.
-

-
- Каждый процесс ЖЦ разделен на **набор действий**, каждое действие – на **набор задач**.
 - Очень важное отличие ISO:
 - каждый процесс, действие или задача инициируется и выполняется другим процессом по мере необходимости, причем нет заранее определенных последовательностей (естественно, при сохранении логики связей по исходным сведениям задач и т. п.).
 - Все **процессы жизненного цикла** изображены на рис. 4.5.
-



Рис. 4.5. Процессы жизненного цикла программных средств

-
- В соответствии с базовым международным стандартом ISO/IEC 12207 все *процессы ЖЦ ПО* делятся на три группы:
 - **Основные процессы**
 - **Вспомогательные процессы**
 - **Организационные процессы**

 - **Основные процессы**
 - ❑ приобретение;
 - ❑ поставка;
 - ❑ разработка;
 - ❑ эксплуатация;
 - ❑ сопровождение.
-

■ **Вспомогательные процессы:**

- ❑ документирование;
- ❑ управление конфигурацией;
- ❑ обеспечение качества;
- ❑ разрешение проблем;
- ❑ аудит;
- ❑ аттестация;
- ❑ совместная оценка;
- ❑ верификация.

■ **Организационные процессы:**

- ❑ создание инфраструктуры;
 - ❑ управление;
 - ❑ обучение;
 - ❑ усовершенствование
-

Таблица 4.1

Содержание основных процессов ЖЦ ПО ПС (ISO/IEC 12207)

Процесс (исполнитель процесса)	Действия	Вход	Результат
Приобретение (заказчик)	• Инициирование • Подготовка заявочных предложений • Подготовка договора • Контроль деятельности поставщика • Приемка ИС	• Решение о начале работ по внедрению ПС • Результаты обследования деятельности заказчика • Результаты анализа рынка ИС/ тендера • План поставки/ разработки • Комплексный тест ИС	• Техничко-экономическое обоснование внедрения ПС • Техническое задание на ПС • Договор на поставку/ разработку • Акты приемки этапов работы • Акт приемно-сдаточных испытаний

Таблица 4.1

Содержание основных процессов ЖЦ ПО ПС (ISO/IEC 12207)

Процесс (исполнитель процесса)	Действия	Вход	Результат
Поставка (разработчик ИС)	• Инициирование • Ответ на заявочные предложения • Подготовка договора • Планирование исполнения • Поставка ПС	• Техническое задание на ПС • Решение руководства об участии в разработке • Результаты тендера • Техническое задание на ПС • План управления проектом • Разработанная ПС и документация	• Решение об участии в разработке • Коммерческие предложения/ конкурсная заявка • Договор на поставку/ разработку • План управления проектом • Реализация/ корректировка • Акт приемно-сдаточных испытаний

Таблица 4.1

Содержание основных процессов ЖЦ ПО ПС (ISO/IEC 12207)

Процесс (исполнитель процесса)	Действия	Вход	Результат
Разработка (разработчик ИС)	- Подготовка - Анализ требований к ПС - Проектирование архитектуры ПС - Разработка требований к ПО - Проектирование архитектуры ПО - Детальное проектирование ПО - Кодирование и тестирование ПО - Интеграция ПО и квалификационное тестирование ПО - Интеграция ПС и квалификационное тестирование ИС	- Техническое задание на ПС - Техническое задание на ИС, модель ЖЦ - Техническое задание на ПС - Подсистемы ПС - Спецификации требования к компонентам ПО - Архитектура ПО - Материалы детального проектирования ПО - План интеграции ПО, тесты - Архитектура ИС, ПО, документация на ИС, тесты	- Используемая модель ЖЦ, стандарты разработки - План работ - Состав подсистем, компоненты оборудования - Спецификации требования к компонентам ПО - Состав компонентов ПО, интерфейсы с БД, план интеграции ПО - Проект БД, спецификации интерфейсов между компонентами ПО, требования к тестам - Тексты модулей ПО, акты автономного тестирования - Оценка соответствия комплекса ПО требованиям ТЗ - Оценка соответствия ПО, БД, технического комплекса и комплекта документации требованиям ТЗ

-
- **Вспомогательные процессы** предназначены для поддержки выполнения основных процессов, обеспечения качества проекта, организации верификации, проверки и тестирования ПО.
 - **Организационные процессы** определяют действия и задачи, выполняемые как заказчиком, так и разработчиком проекта для управления своими процессами.
 - Для поддержки практического применения стандарта ISO/IEC 12207 разработан **ряд технологических документов**:
 - ГОСТ Р ИСО/МЭК ТО 15271-2002 «Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств)»
 - ГОСТ Р ИСО/МЭК ТО 16326-2002 «Программная инженерия. Руководство по применению ГОСТ Р ИСО/МЭК 12207 при управлении проектом»
-

- Позднее был разработан и в 2002 г. опубликован стандарт на процессы **жизненного цикла систем: ISO/IEC 15288-2002 «Information technology. System engineering. System life cycle processes»**.
- В России он был введен с 1 января 2007 г.: ГОСТ Р ИСО/МЭК 15288-2005 «Информационная технология. Системная инженерия. Процессы жизненного цикла систем».
- К разработке стандарта были привлечены специалисты различных областей: системной инженерии, программирования, управления качеством, человеческими ресурсами, безопасностью и пр.
- Был учтен практический опыт создания систем в правительственных, коммерческих, военных и академических организациях.
- Стандарт применим для широкого класса систем, но его основное предназначение — **поддержка создания компьютеризированных систем**.

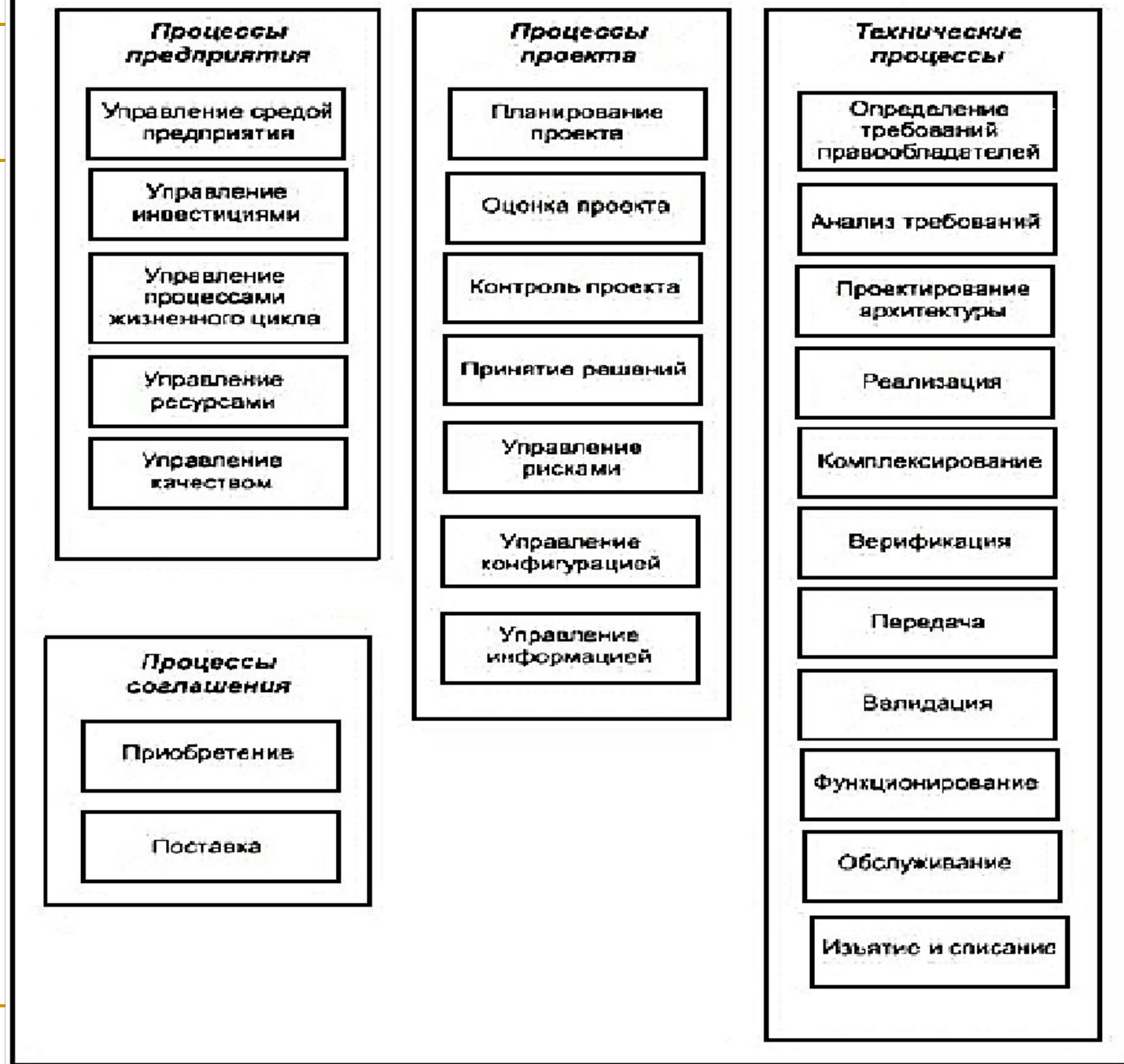


Рис. 4.5. Процессы жизненного цикла системы

-
- Каждый процесс жизненного цикла при необходимости **может быть начат в любой момент жизненного цикла**, при этом нет определенного порядка в их использовании.
 - Любой процесс может **выполняться** одновременно с любыми другими процессами жизненного цикла и может быть реализован **на любом уровне иерархии** структуры системы.
 - Стадии создания системы, предусмотренные в стандарте ISO/IEC 15288, несколько отличаются от рассмотренных выше.
 - Перечень **стадий и основные результаты**, которые должны быть достигнуты к моменту их завершения, приведены в таблице 4.2.
-

Таблица 4.2.**Стадии создания систем (ISO/IEC 15288)**

№ п/п	Стадия	Описание
1	Формирование концепции	Анализ потребностей, выбор концепции и проектных решений
2	Разработка	Проектирование системы
3	Реализация	Изготовление системы
4	Эксплуатация	Ввод в эксплуатацию и использование системы
5	Поддержка	Обеспечение функционирования системы
6	Снятие с эксплуатации	Прекращение использования, демонтаж, архивирование системы

-
- **Жизненный цикл ПО** можно представить, как ряд событий, происходящих с системой в процессе ее создания и использования
 - *Модель жизненного цикла* – структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.
 - В настоящее время известны и используются следующие *модели жизненного цикла*:
 - *Каскадная модель*
 - *Поэтапная модель с промежуточным контролем*
 - *Спиральная модель*
-

- Можно выделить следующие **положительные стороны** применения **каскадного подхода**:
на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.
- Основным **недостатком этого подхода** является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений.
- В результате реальный процесс создания ПС оказывается соответствующим *поэтапной модели с промежуточным контролем*

-
- В **спиральной модели** ЖЦ каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы.
 - Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали.
 - Таким образом **углубляются и последовательно конкретизируются детали проекта** и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.
-

-
- **СТБ ИСО/МЭК 12207-2003** Информационная технология. Процессы жизненного цикла программных систем
 - **ГОСТ Р ИСО/МЭК 12207-1999**
 - Стандарт ISO 12207 равносильно ориентирован на организацию действий каждой из двух сторон: поставщик (разработчик) и покупатель (пользователь); может быть в равной степени применен, когда обе стороны – из одной организации.
 - Каждый процесс ЖЦ разделен на набор действий, каждое действие – на набор задач.
 - ***Очень важное отличие ISO:*** каждый процесс, действие или задача инициируется и выполняется другим процессом по мере необходимости, причем нет заранее определенных последовательностей (естественно, при сохранении логики связей по исходным сведениям задач и т. п.).
-

-
- **ISO/IEC 15288-2002** «Information technology. System engineering. System life cycle processes».
 - **ГОСТ Р ИСО/МЭК 15288-2005** «Информационная технология. Системная инженерия. Процессы жизненного цикла систем».
 - К разработке стандарта были привлечены специалисты различных областей: системной инженерии, программирования, управления качеством, человеческими ресурсами, безопасностью и пр.
 - Был учтен практический опыт создания систем в правительственных, коммерческих, военных и др. организациях.
 - Стандарт применим для широкого класса систем, но его основное предназначение – **поддержка создания компьютеризированных систем.**
-

-
- 1. Дайте определение жизненного цикла ПО.
 - 2. Определите модели жизненного цикла: каскадная, модель с промежуточным контролем, спиральная.
 - 3. Что такое каскадная модель жизненного цикла?
 - 4. Что такое поэтапная модель ЖЦ с промежуточным контролем?
 - 5. Что такое спиральная модель жизненного цикла?
 - 6. Основные причины популярности каскадной модели.
 - 7. Процессы жизненного цикла и стандарты.
 - 8. Регламентация процессов проектирования в отечественных и международных стандартах.
 - 9. Процессы жизненного цикла в СТБ ИСО/МЭК 12207-2003: основные, вспомогательные, организационные.
 - 10. Содержание и взаимосвязь процессов жизненного цикла ПС в СТБ ИСО/МЭК 12207-2003.
 - 11. Процессы и стадии жизненного цикла в ГОСТ Р ИСО/МЭК 15288-2005
-