

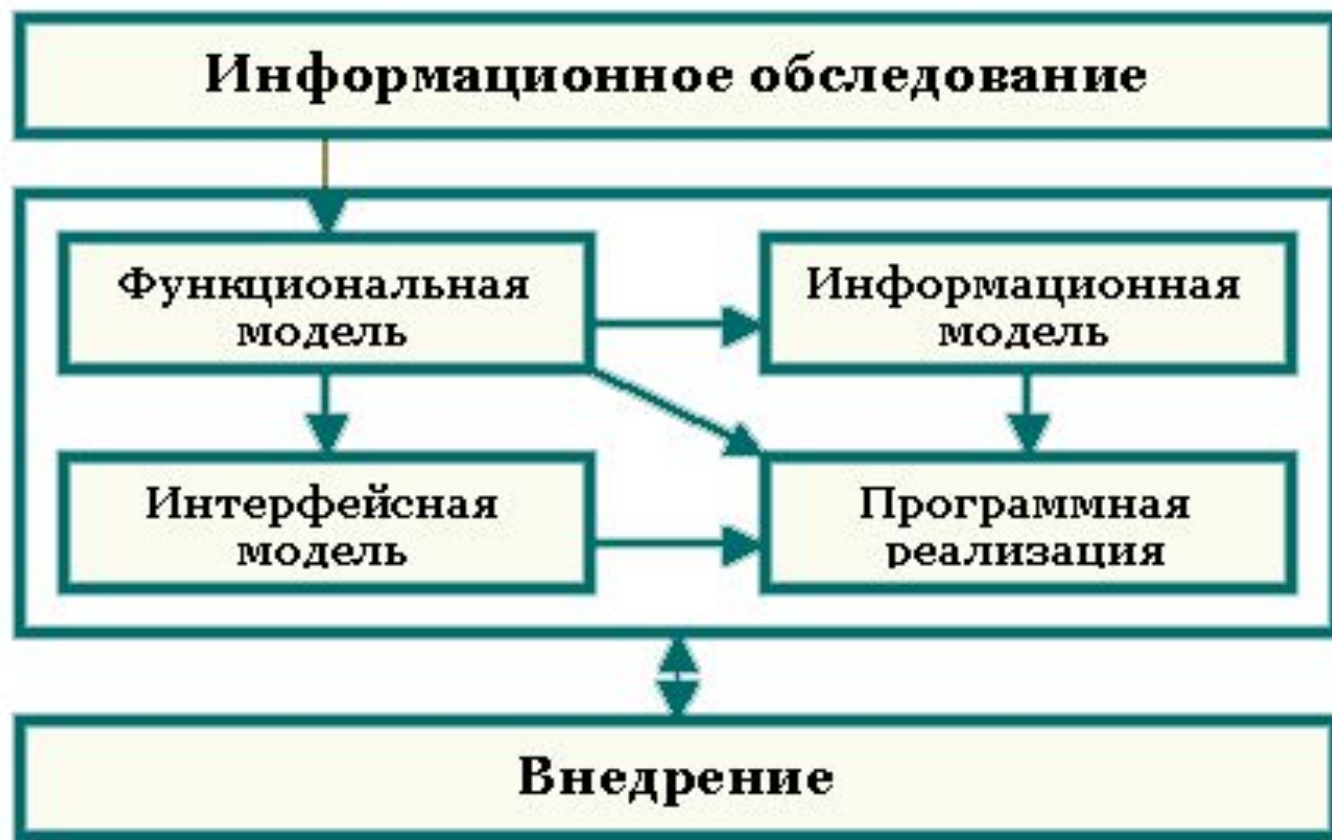
**Тема урока:
Жизненный цикл
информационных
систем**



Жизненный цикл информационной системы — период времени, который начинается с момента принятия решения о необходимости создания информационной системы и заканчивается в момент ее полного изъятия из эксплуатации.



Первый этап жизни ИС



Стадия — часть процесса создания ИС, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей, программных компонентов, документации), определяемого заданными для данной стадии требованиями.



Жизненный цикл информационной системы подразделяется на стадии:

1. анализ предметной области и формирование требований;
2. проектирование;
3. реализация;
4. тестирование;
5. ввод в эксплуатацию;
6. эксплуатация (сопровождение проекта).

Процессы жизненного цикла ИС

Процесс определяется как совокупность взаимосвязанных действий, преобразующих входные данные в выходные.



Группы процессов жизненного цикла ПО

Процессы ЖЦ ПО

Основные

приобретение
поставка
разработка
эксплуатация
сопровождение

Вспомогательные

документирование
управление конфигурацией
обеспечение качества
верификация
аттестация
совместная оценка
аудит
разрешение проблем

Организационные

управление
усовершенствование
задание инфраструктуры
обучение

Модели жизненного цикла

— структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла



Модель ЖЦ ИС включает в себя

- стадии;
- результаты выполнения работ на каждой стадии;
- ключевые события — точки завершения работ и принятия решений.

Типы моделей жизненного цикла информационных систем

- Каскадная модель.
- Поэтапная модель с промежуточным контролем
- Спиральная модель.

Каскадная модель



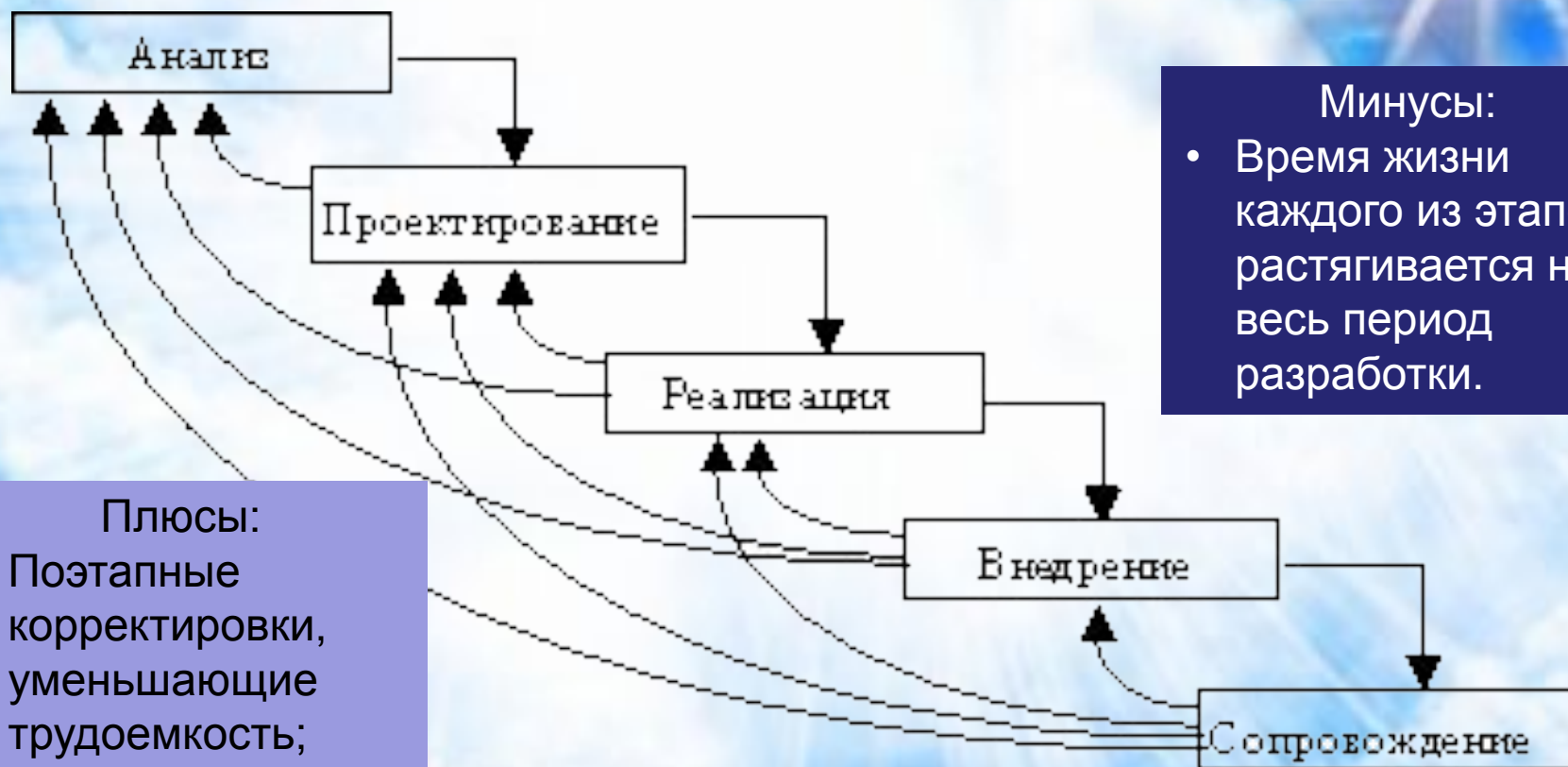
Плюсы:

- Возможность планирования времени осуществления всех этапов проекта;
- Упорядочение хода конструирования.

Минусы:

- Недостаточно гибкая модель;
- Все требования не всегда формируются в начале проекта;
- Результаты проекта доступны заказчику только в конце работы.

Поэтапная модель с промежуточным контролем (итерационная модель)



Минусы:

- Время жизни каждого из этапов растягивается на весь период разработки.

Плюсы:

- Поэтапные корректировки, уменьшающие трудоемкость;
- Эффективная обратная связь с потребителем.

Спиральная модель ЖЦ ИС

Плюсы:

- Отображает объективно существующую ИС;
- Последовательно конкретизирует детали проекта;
- Возможность перехода на следующий этап, не дожидаясь полного завершения работы на текущем;
- Быстрота получения работоспособного продукта пользователем.



Минусы:

- Повышенные требования к заказчику;
- Трудности контроля и управления временем разработки.

На каждом витке спирали выполняется создание очередной версии ИС следующим образом:

1. Начальный сбор требований и планирование проекта;
2. Та же работа, но на основе рекомендаций заказчика;
3. Анализ риска на основе начальных требований;
4. Анализ риска на основе реакции заказчика;
5. Переход к комплексной системе;
6. Начальный макет системы;
7. Следующий уровень макета;
8. Сконструированная система;
9. Оценивание заказчиком.



На практике наибольшее распространение получили две основные модели жизненного цикла:

каскадная модель

(характерна для периода 1970-1985 г.);

спиральная модель

(характерна для периода после 1986.г.).



Роли, выделяемые в коллективе разработчиков:

- **Заказчик** – представитель организации, заказавшей разрабатываемое ПО.
- **Эксперт предметной области** – отвечает за изучение сферы приложения.
- **Планировщик ресурсов** – выдвигает и координирует требования к проектам в организации, осуществляющей данную разработку; развивает и направляет план выполнения проекта с точки зрения организации.
- **Менеджер проекта** – отвечает за развитие проекта в целом; гарантирует, что работы и предъявленные результаты идут по графику, что результаты соответствуют требованиям.

Роли, выделяемые в коллективе разработчиков (продолжение):

- **Руководитель команды - производит техническое руководство командой в процессе выполнения проекта.**
- **Архитектор – отвечает за проектирование архитектуры системы, согласовывает развитие работ, связанных с проектом.**
- **Разработчик – реализует проектируемые компоненты, владеет и создает специфические классы и методы, осуществляет кодирование и автономное тестирование, строит продукт.**
- **Специалист по пользовательскому интерфейсу – отвечает за удобство применения системы. Работает с заказчиком.**

Роли, выделяемые в коллективе разработчиков (окончание):

- **Специалист по тестированию – проверяет функциональность, качество и эффективность продукта.**
- **Специалист по внедрению и сопровождению – участвует в анализе особенностей площадки заказчика.**
- **Специалист по безопасности – ответственен за весь спектр вопросов безопасности создаваемого продукта.**
- **Технический писатель – выполняет обязанности по подготовке документации к разработанному продукту, финального описание функциональных возможностей.**
- **Библиотекарь – отвечает за создание и ведение общей библиотеке проекта.**