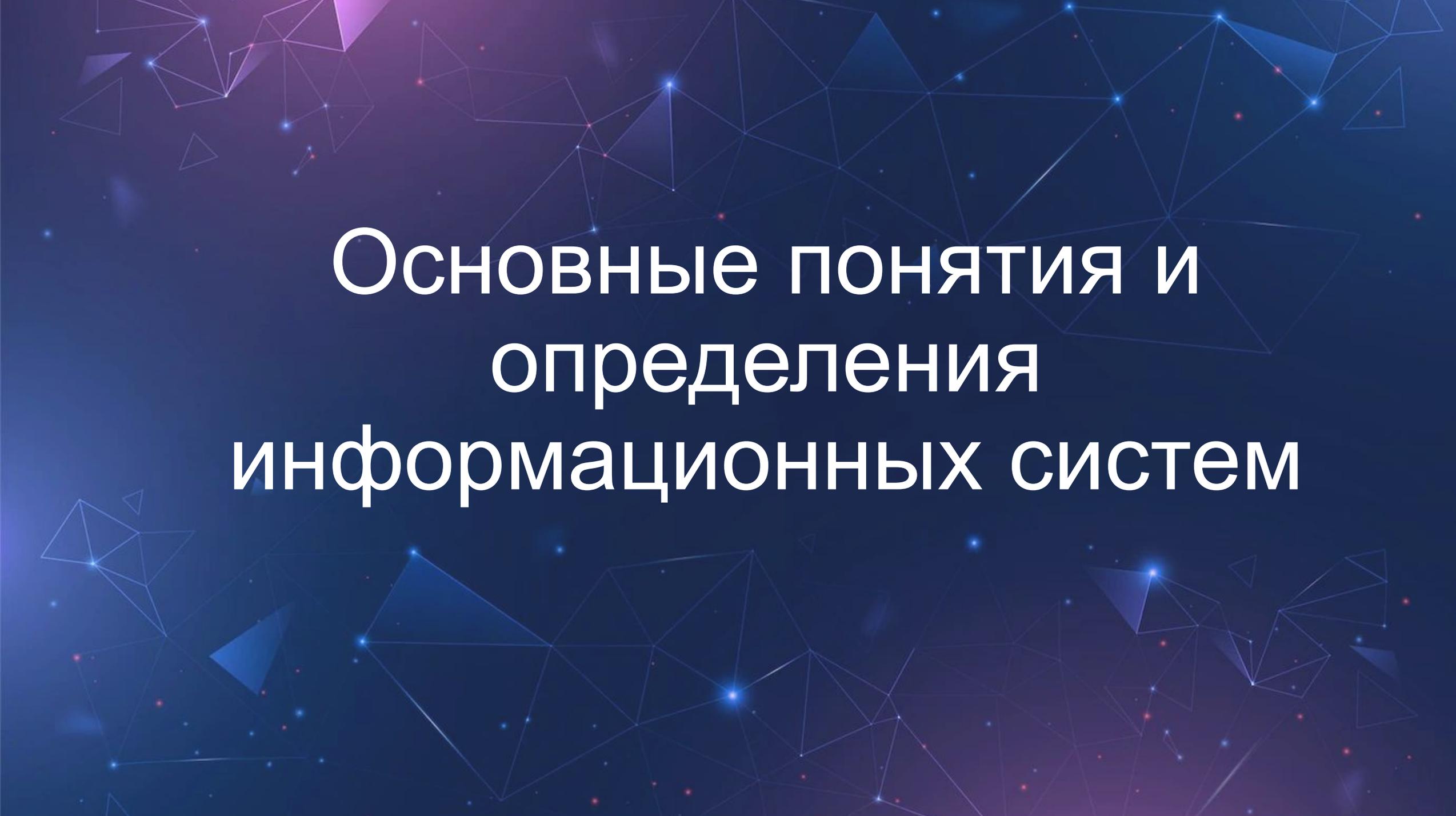


**ПМ. 05 Проектирование и
разработка информационных
систем**

**МДК 05.01 Проектирование и дизайн
информационных систем**



Основные понятия и определения информационных систем

Понятие информационной системы (ИС)

Прикладная программная подсистема, ориентированная на сбор, хранение, поиск и обработку текстовой и/или фактографической информации.



Свойства информационных систем

любая ИС может быть подвергнута анализу, построена и

управляема на основе общих принципов построения сложных систем

необходимо

использовать

ИС является подход

воспринимать как динамичной и

развивающейся системы,

состоящую из

входной продукцией компьютерных и

ИС является

информация, на основе которой

реализованную на

части человека, и в которой

зависит от сложности технологий

системы, типов и

выполнение рутинных наборов данных, операций

степени

формализации

решаемых задач

Процессы в информационной системе



Классификация информационных систем

По областям применения

- Информационные системы в экономике (АСЭ – автоматизированные системы в экономике). В образовании (АСО). В

По характеру информации, которой оперирует ИС

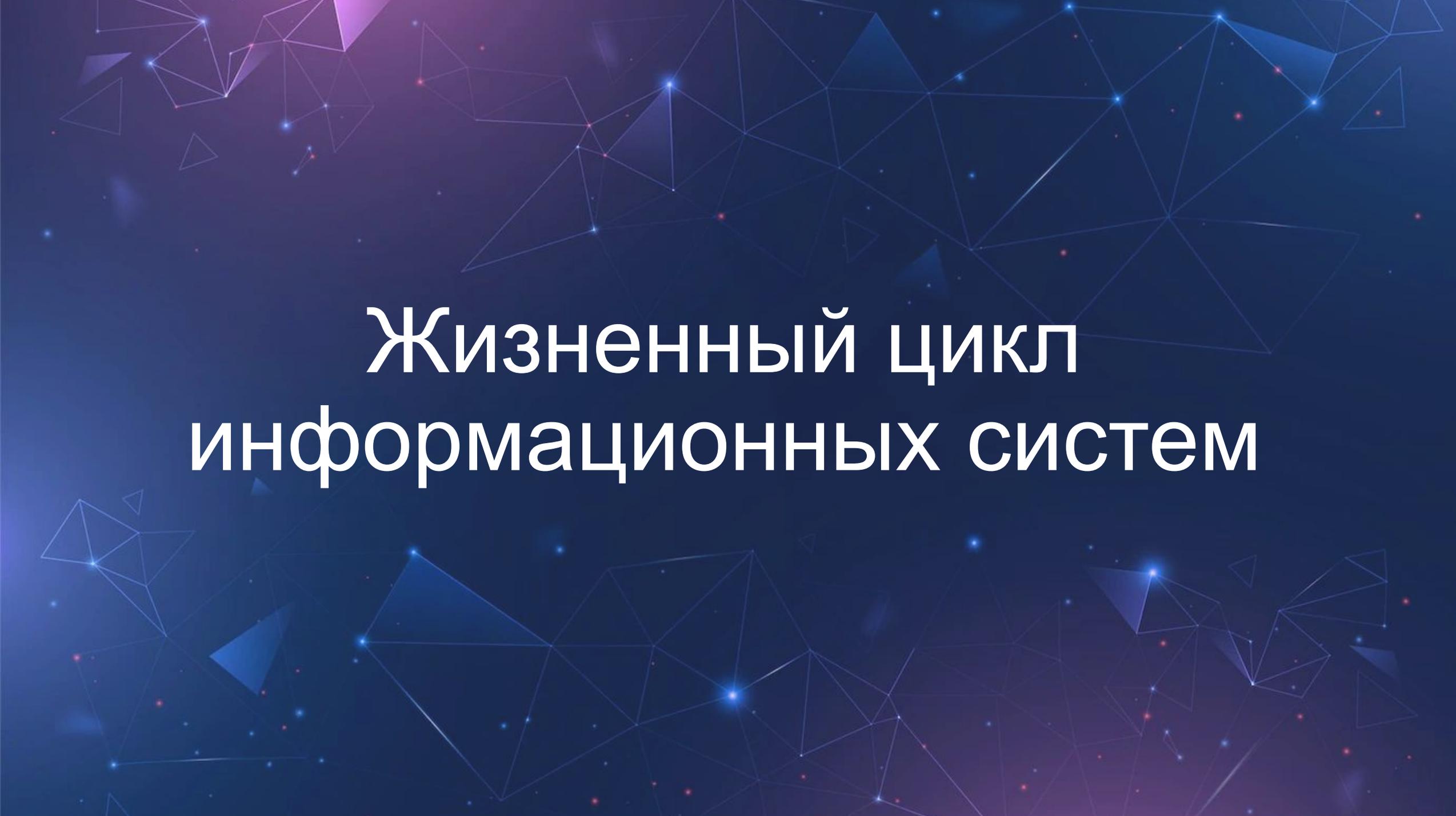
- Фактографические или документальные

По роли, которую ИС играют в профессиональной деятельности

- Системы управления. АСУ, САУ
- Вычислительные информационные системы

По техническим средствам

- Системы принятия решения
- Один компьютер / Локальная сеть / Глобальная сеть

The background features a complex network of thin, light blue lines connecting various points, creating a mesh-like structure. Interspersed among these lines are several small, glowing blue and purple dots, some of which are larger and more prominent. The overall color palette is a gradient of deep blues and purples, with a subtle glow emanating from the central text area.

Жизненный цикл информационных систем

Понятие жизненного цикла (ЖЦ) ИС

Жизненный цикл информационной системы (ЖЦИС) - это период создания и использования ИС, начиная с момента возникновения потребности в ИС и заканчивая моментом полного её выхода из эксплуатации.

- анализ требований
- проектирование
- кодирование (программирование)
- тестирование и отладка
- эксплуатация и сопровождение

Стадии жизненного цикла информационной системы

- Предпроектное обследование
- Проектирование
- Разработка ИС
- Ввод ИС в эксплуатацию
- Эксплуатация ИС

Предпроектное обследование

Сбор материалов для проектирования; формулирование требований, изучение объекта автоматизации, даются предварительные выводы предпроектного варианта ИС.

Анализ материалов и разработка документации; разрабатывается технико-экономическое обоснование с техническим заданием на стадии проектирования ИС.

Проектирование

Предварительное проектирование:

- выбор проектных решений по аспектам разработки ИС; описание реальных компонент ИС;

Детальное проектирование:

- выбор или разработка математических методов или алгоритмов программ;
- корректировка структур БД;

Разработка техно-рабочего проекта ИС (ТРП)

документацией на ее установку.
Разработка методологии реализации функций управления с помощью ИС и описанием регламента действий аппарата управления

Разработка ИС

получение и
установка
технических и
программных
средств

тестирование и
доводка
программного
комплекса

разработка
инструкций по
эксплуатации
программно-
технических средств

Ввод ИС в эксплуатацию

Ввод технических
средств

Ввод программных
средств

обучение и
сертификация
персонала

опытная
эксплуатация

сдача и
подписание актов
приёмки-сдачи
работ

Эксплуатация ИС

повседневная эксплуатация

общее сопровождение всего
проекта

Модель жизненного цикла ИС

Структура, описывающая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного обеспечения в течение всей жизни ИС, от определения требований до завершения её использования

каскадная
(водопадная)

инкрементная
модель

спиральная
модель

Каскадная модель

Классическая модель однократного прохода, которая описывает линейную последовательность этапов создания ИС

- Анализ
- Проектирование
- Реализация
- Тестирование
- Внедрение

Достоинства и недостатки каскадной модели

Достоинства

- Явное описание всех этапов работы и определение последовательности их реализации. Это позволяет планировать сроки завершения работ и соответствующие затраты.

Недостатки

- Реальный процесс создания ИС в действительности практически никогда не укладывается в жёсткую каскадную схему. Постоянно возникает потребность в возврате к предыдущим этапам для уточнения требований и исходных данных.

Инкрементная модель

Метод, в котором проект проектируется, реализуется и тестируется инкрементно (то есть каждый раз с небольшими добавлениями) до самого окончания разработки.

Версия 1



Версия 2



Версия 3



Достоинства и недостатки инкрементной модели

Достоинства

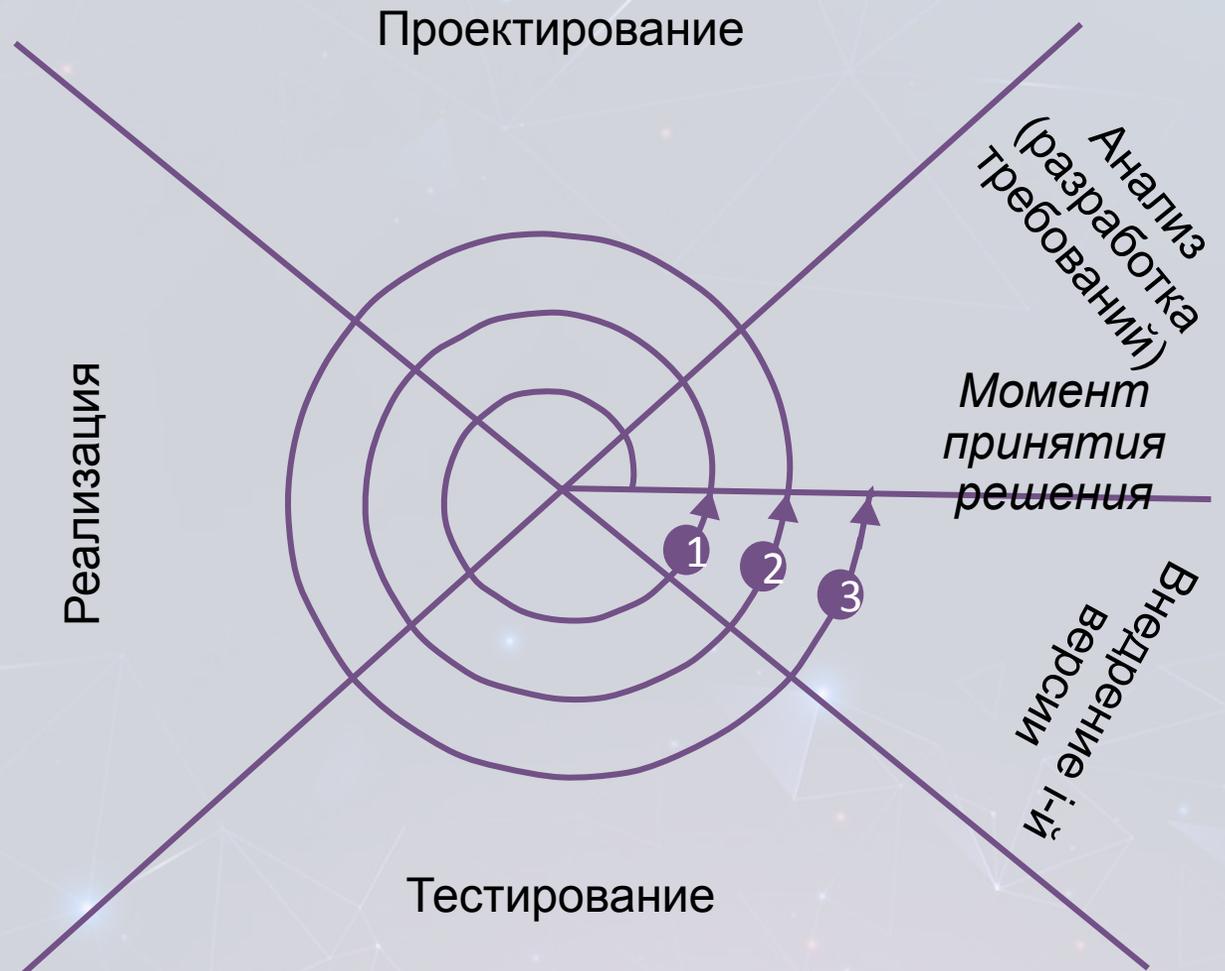
- Жизненный цикл позволяет заказчику контролировать процесс разработки системы, начиная с её самой ранней версии – прототипа.

Недостатки

- Как и для классической каскадной модели ЖЦ, перед началом разработки необходимо сформулировать полный набор требований к информационной системе для каждой версии, включая прототип и промежуточные версии.

Спиральная модель

Модель процесса разработки программного обеспечения, ориентированная на риск. Основываясь на уникальных моделях рисков данного проекта, спиральная модель направляет команду на принятие элементов одной или нескольких моделей процесса, таких



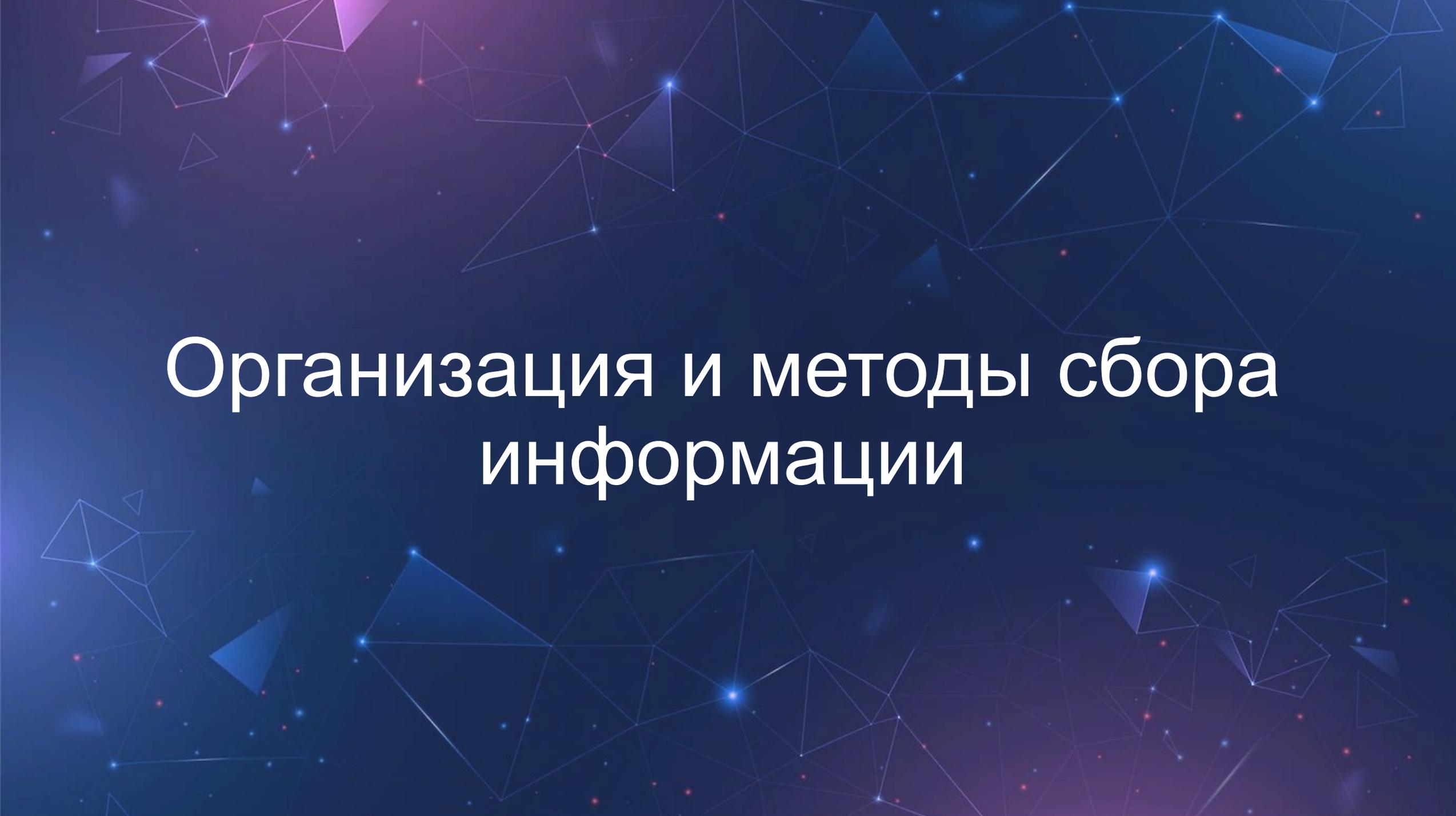
Достоинства и недостатки спиральной модели

Достоинства

- реализации доводится обоснованный окончательный вариант ИС, который удовлетворяет действительным требованиям заказчика
- ускорение разработки ИС обусловленное более

Недостатки

- сложность планирования работ и оценки затрат, сроков и рисков выполнения проекта.



Организация и методы сбора информации

Методология сбора информации

Качественные методы сбора информации включают сбор, анализ и интерпретацию данных путем наблюдения за тем, что люди делают и говорят

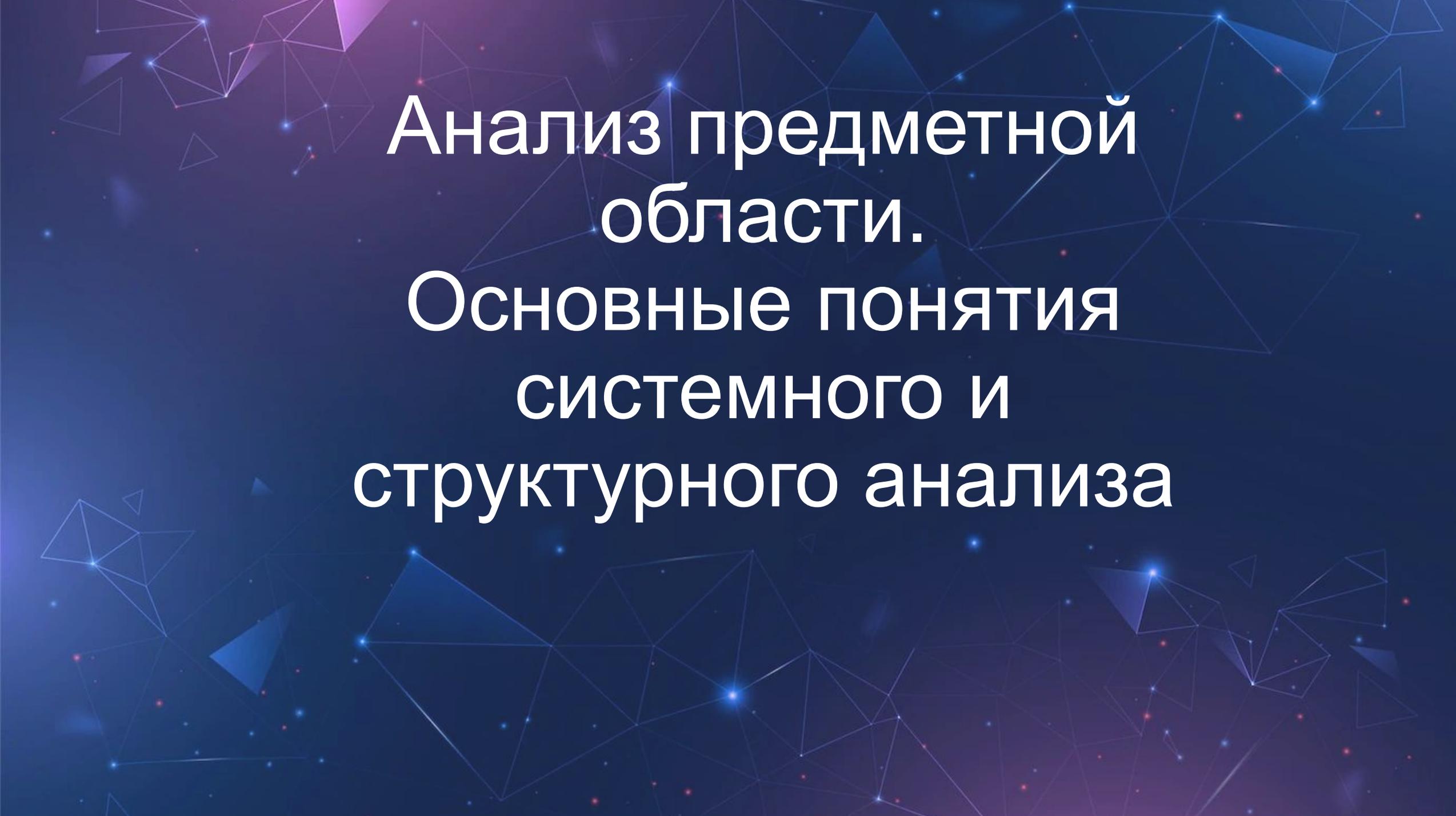
- 1
 - Глубинное интервью

- 2
 - Индивидуальное интервью

- 3
 - Проекционные методы

- 4
 - Наблюдение

- 5
 - Имитационное моделирование



Анализ предметной
области.
Основные понятия
системного и
структурного анализа

Анализ предметной области

Процесс выявления, сбора знаний о предметной области с целью их повторного использования при создании новых систем.

Целью этапа анализа является преобразование общих, расплывчатых знаний об исходной предметной области (требований заказчика) в точные определения и спецификации для разработчиков, а также генерация функционального описания системы.

На этом этапе определяются и специфицируются:

внешние и
внутренние
условия работы
системы

функциональная
структура системы

распределение
функций между
человеком и
системой,
интерфейсы

требования к
техническому,
информационным
и программным
компонентам
системы

требования к
качеству и
безопасности

состав
технической и
пользовательской
документации

условия
внедрения и
эксплуатации

Анализ требований

Обследование предприятия

- исследования системы управления предприятием
- обследования функциональной и

Определение требований:

- формулируются цель и задачи проекта
- происходит сбор и определение всех возможных требований

Анализ

- процесс анализа заключается в разборе требований полученных на предыдущем этапе, их уточнение и систематизация

Системный и структурный анализ

Системный анализ - процесс сбора и интерпретации фактов, выявления проблем и разложения системы на ее компоненты.

Системный анализ проводится с целью изучения системы или ее частей для определения ее целей.

Системный и структурный анализ

Структурный анализ – это метод разработки, который позволяет аналитику логически понимать систему и ее действия.

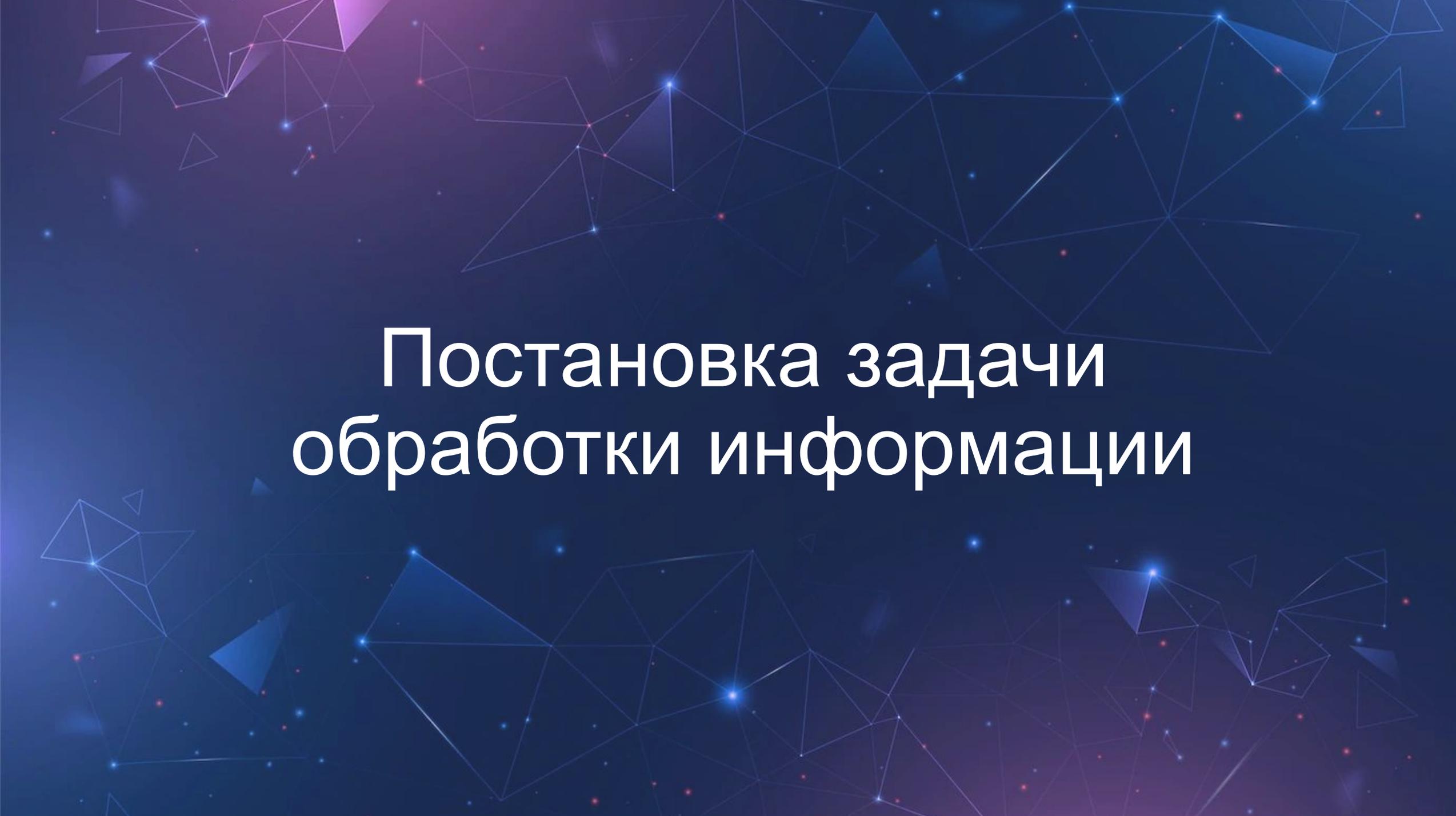
Он имеет следующие атрибуты:

графический,
который указывает
на представление
приложения

разделяет процессы
так, что дает четкую
картину потока
системы

логично, а не
физически, т. е.
элементы системы
не зависят от
поставщика или
оборудования

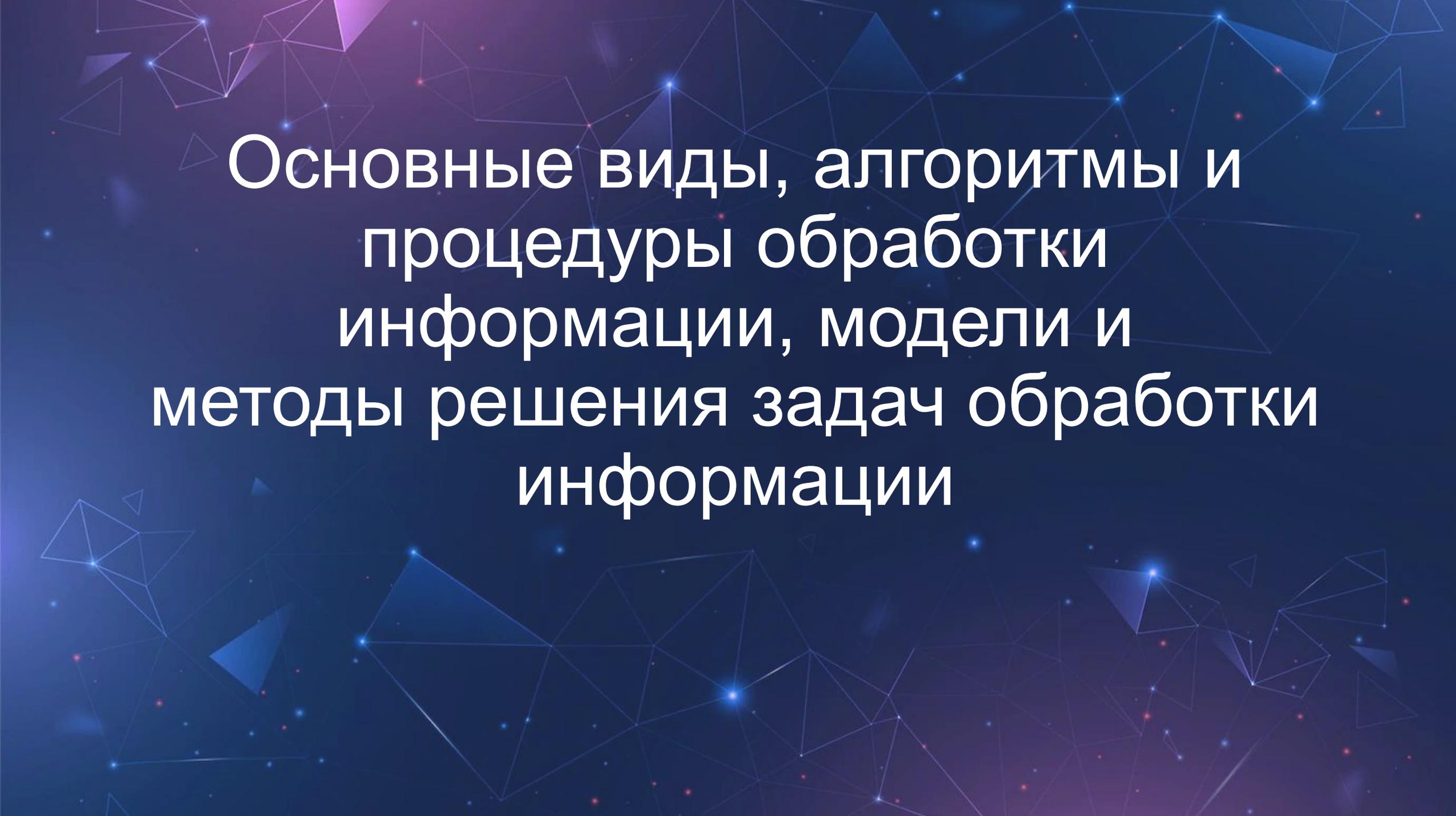
подход, который
работает от обзоров
высокого уровня до
деталей более
низкого уровня



Постановка задачи обработки информации

Обработка информации состоит в получении одних «информационных объектов» из других «информационных объектов» путем выполнения некоторых алгоритмов и является одной из основных операций, осуществляемых над информацией, и главным средством увеличения ее объема и разнообразия.

- Обработка информации
 - Числовая
 - переменные, векторы, матрицы, многомерные массивы, константы и т.д.
 - Нечисловая
 - файлы, записи, поля, иерархии, сети, отношения и т.д.



Основные виды, алгоритмы и
процедуры обработки
информации, модели и
методы решения задач обработки
информации

Виды обработки информации

последовательная обработка, применяемая в традиционной фоннеймановской архитектуре ЭВМ, располагающей одним процессором

параллельная обработка, применяемая при наличии нескольких процессоров в ЭВМ

конвейерная обработка, связанная с использованием в архитектуре ЭВМ одних и тех же ресурсов для решения разных задач

Основные процедуры обработки данных

- Создание данных
- Модификация данных
- Контроль, безопасность и целостность
- Поиск информации, хранимой в памяти компьютера
- Поддержка принятия решения

Процесс принятия решения

Принятие
решений в

условиях

определенности

Принятие

решений в

условиях риска

решений в

условиях

Принятие

неопределенно

сти

условиях

многочисленности

задач

искусственного

интеллекта

Экспертная

система

система

Для решения задач в экспертных системах используют

метод логического вывода, основанный на технике доказательств, называемой резолюцией и использующей опровержение отрицания (доказательство «от противного»);

метод структурной индукции, основанный на построении дерева принятия решений для определения объектов из большого числа данных на входе;

метод эвристических правил, основанных на использовании опыта экспертов, а не на абстрактных правилах формальной логики;

метод машинной аналогии, основанный на представлении информации о сравниваемых объектах в удобном виде, например, в виде структур данных, называемых фреймами

Процесс выработки решения на основе первичных данных

- выработка допустимых вариантов решений путем математической формализации с использованием разнообразных моделей
- выбор оптимального решения на основе субъективных факторов

Обязательные компоненты для поддержки принятия решений

обобщающий
анализ

прогнозирование

ситуационное
моделирование

Информационные системы поддержки принятия решений

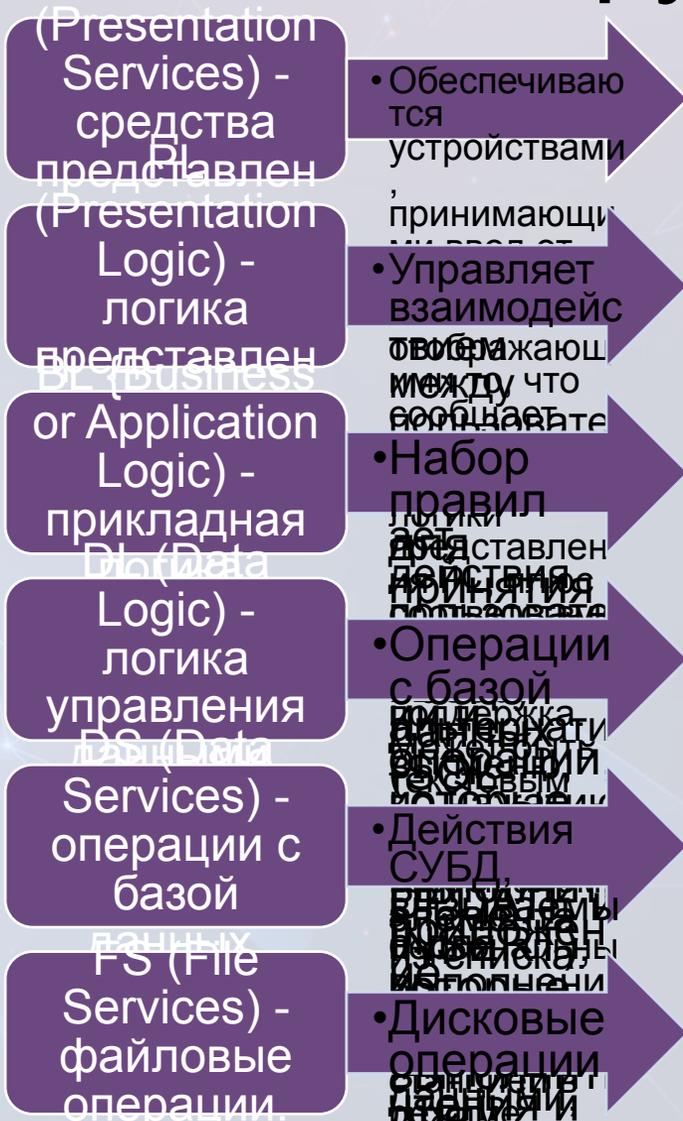
Системы поддержки принятия решений DSS (Decision Support System) осуществляют отбор и анализ данных по различным характеристикам и включают средства:

- доступа к базам данных;
- извлечения данных из разнородных источников;
- моделирования правил и стратегии деловой деятельности;
- деловой графики для представления результатов анализа;
- анализа «если что»;

Системы оперативной аналитической обработки OLAP (OnLine Analysis Processing) для принятия решений используют следующие средства:

- мощную многопроцессорную вычислительную технику в виде специальных OLAP-серверов;
- специальные методы многомерного анализа;
- специальные хранилища данных Data Warehouse.

Типовые функциональные компоненты



Средства разработки информационных приложений

традиционные
системы
программирования

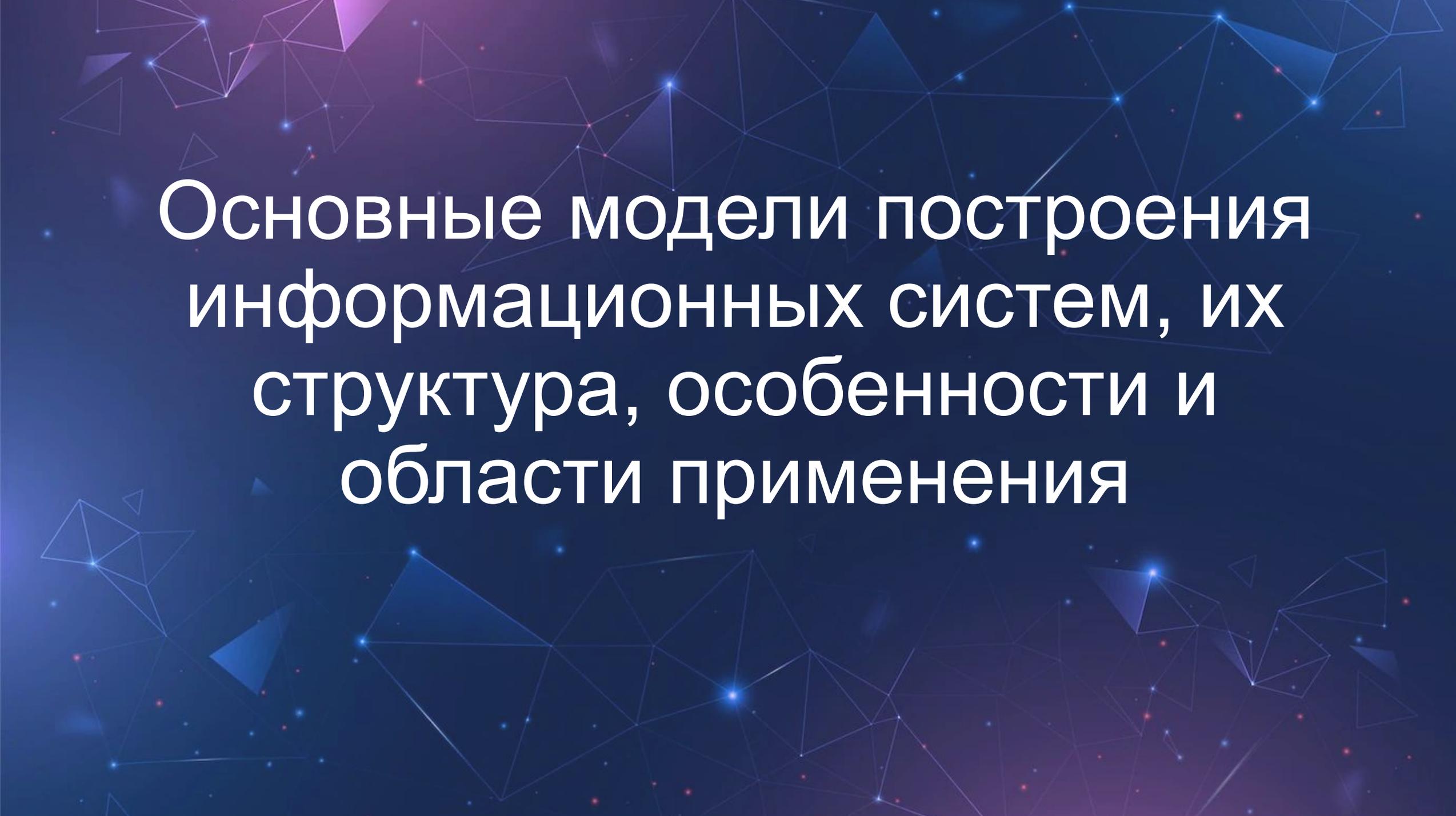
инструменты для
создания файл-
серверных
приложений

средства разработки
приложений «клиент-
сервер»

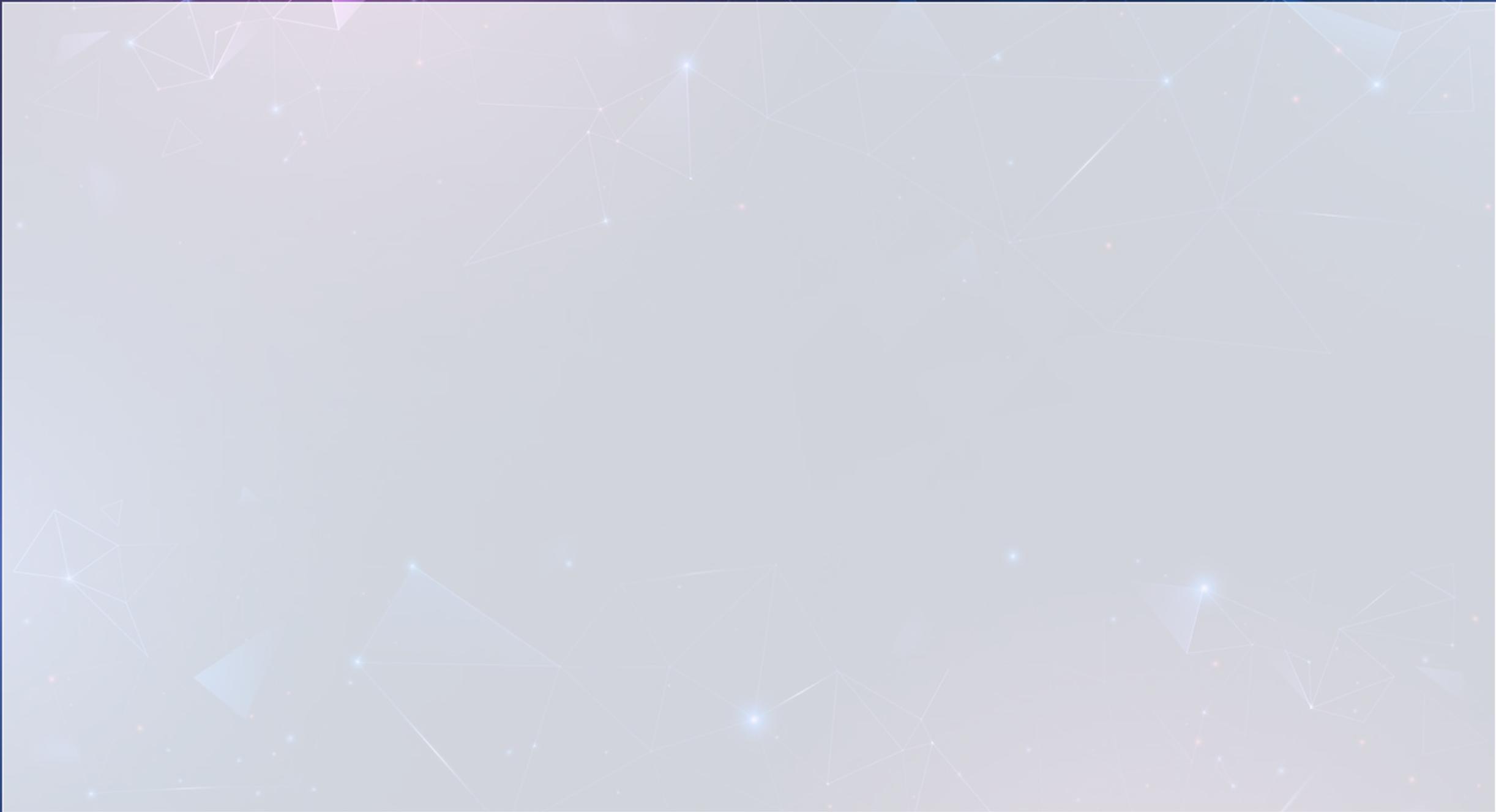
средства
автоматизации
делопроизводства и
документооборота

средства разработки
Интернет/Инtranет-
приложений

средства
автоматизации
проектирования
приложений

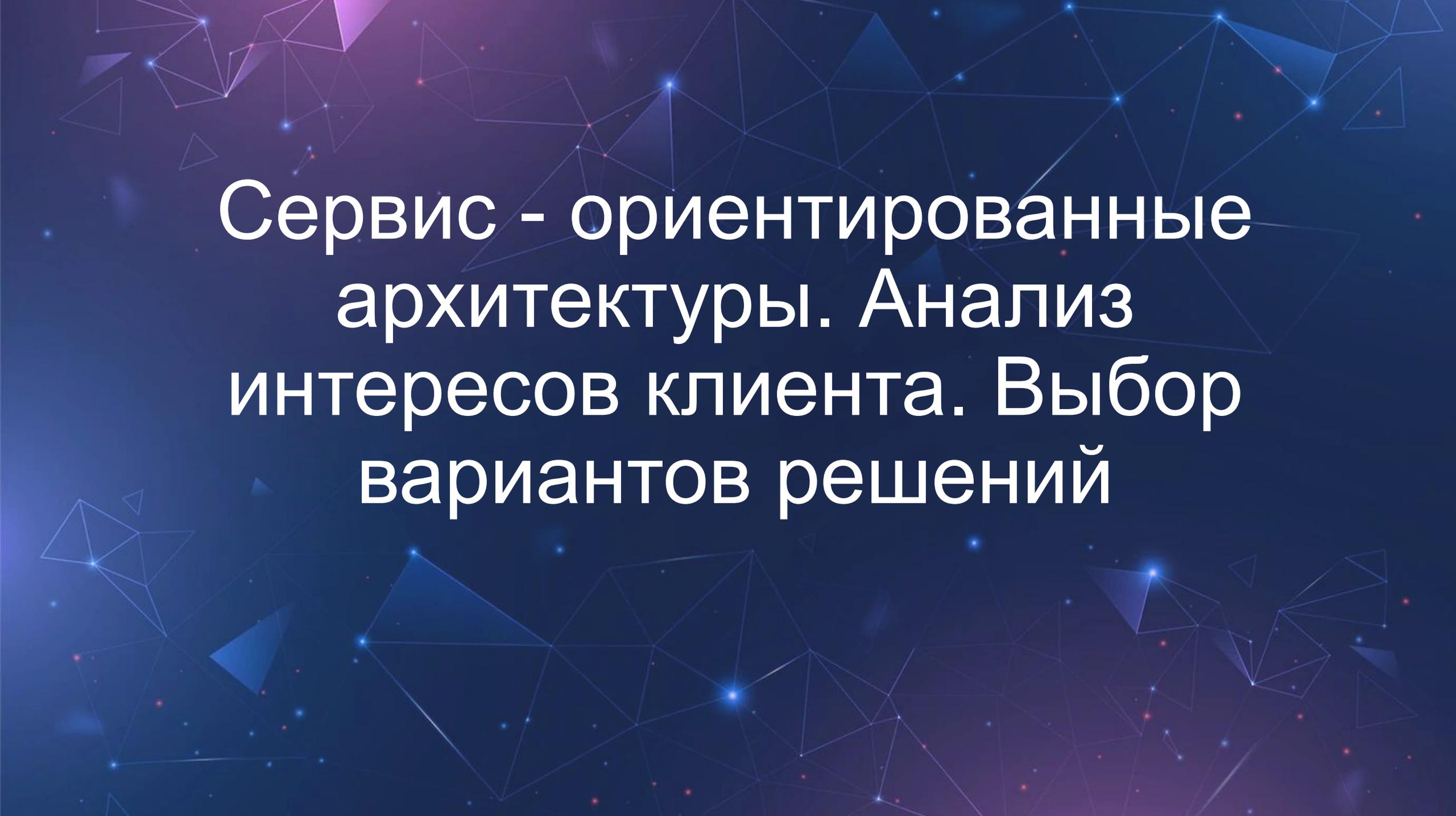


Основные модели построения информационных систем, их структура, особенности и области применения



Средства разработки информационных приложений

Модель «Черного ящика»



Сервис - ориентированные
архитектуры. Анализ
интересов клиента. Выбор
вариантов решений

Сервис-ориентированная архитектура (SOA)

SOA - модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам

SOA можно свести к нескольким идеям, причём архитектура не диктует способы их реализации:

Сочетаемость приложений, ориентированных на пользователей.

Множественное использование бизнес-сервисов.

Независимость от набора технологий.

Автономность (независимые эволюция, масштабируемость и развёртываемость).

Паттерны, относящиеся к SOA

Общая архитектура

брокера

объектных

запросов

(CORBA)

Веб-сервисы

Очередь

сообщений

Сервисная шина

предприятия

(ESB)

Микросервисы

Общая архитектура брокера объектных запросов (CORBA)

Стандарт CORBA был реализован несколькими вендорами.

Не зависящие от платформы вызовы удалённых процедур. Он обеспечивает:

от
платформы
вызовы
удалённых

процедур

Транзакции (в
том числе
удалённые)

Безопасность

События

Независимост
ь от выбора
языка
программиров
ания

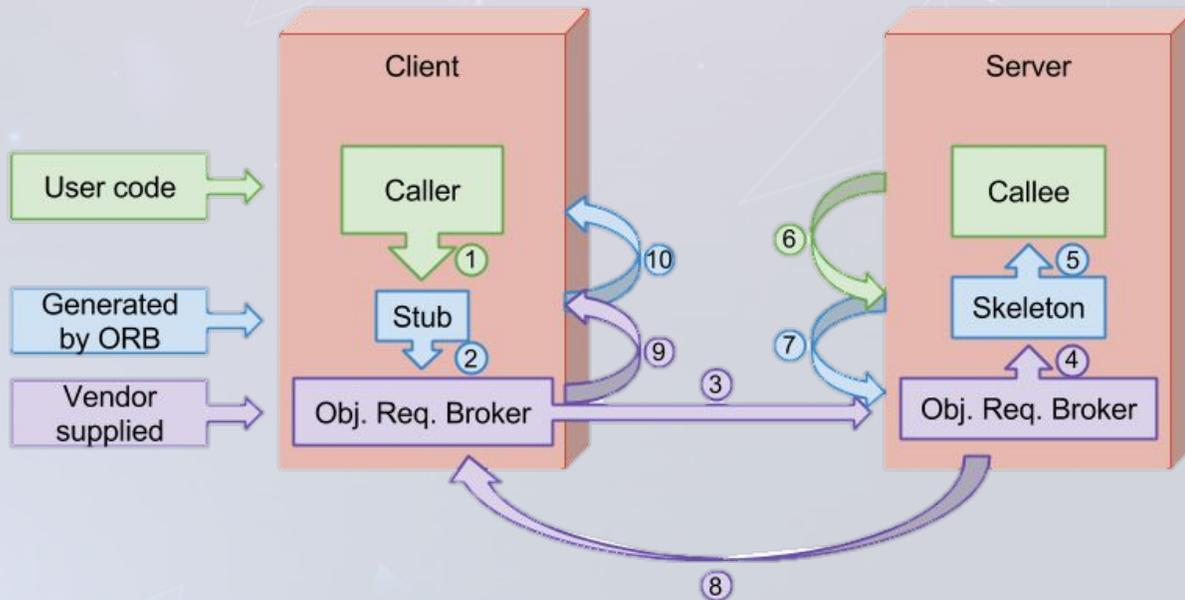
Независимост
ь от выбора
ОС

Независимост
ь от выбора
оборудования

Независимост
ь от
особенностей
передачи
данных/связи

Набор данных
через язык
описания
интерфейсов

Принцип работы



Вызывающая программа (caller) вызывает локальную процедуру, реализованную заглушкой.

1. Заглушка проверяет вызов, создаёт сообщение-запрос и передаёт его в ORB.
2. Клиентский ORB шлёт сообщение по сети на сервер и блокирует текущий поток выполнения.
3. Серверный ORB получает сообщение-запрос и создаёт экземпляр скелета.
4. Скелет исполняет процедуру в вызываемом объекте.
5. Вызываемый объект проводит вычисления и возвращает результат.
6. Скелет упаковывает выходные аргументы в сообщение-ответ и передаёт его в ORB.

7. ORB шлёт сообщение по сети клиенту.
8. Клиентский ORB получает сообщение, распаковывает и передаёт информацию заглушке.
9. Заглушка передаёт выходные аргументы вызывающему методу,

Достоинства и недостатки

Достоинства

- Независимость от выбранных технологий (не считая реализации ORB)
- Независимость от особенностей передачи данных/связи

Недостатки

- Независимость от местоположения: клиентский код не имеет понятия, является ли вызов локальным или удалённым
- Сложная, раздутая и неоднозначная спецификация: её собрали из нескольких версий спецификаций разных вендоров, поэтому (на тот момент) она была раздутой, неоднозначной и трудной в реализации.

Веб-сервисы

Нужен был надёжный канал связи, поэтому:

HTTP стал по умолчанию

Нужно было уменьшить количество удалённых обращений, поэтому:

Для обмена сообщениями удалёнными

удалённые использовали в качестве

Нужно было упростить спецификацию обмена сообщениями, поэтому:

Вместо многоступенчатых удалённых

удалённые использовали в качестве

удалённые использовали в качестве

удалённые использовали в качестве

Достоинства и недостатки

Достоинства

- Независимость набора технологий, развёртывания и масштабируемости сервисов
- Стандартный, простой и надёжный канал связи (передача текста по HTTP через порт 80)

• Оптимизированный обмен сообщениями

Недостатки

- Разные веб-сервисы тяжело интегрировать из-за различий в языках передачи сообщений. Например, два веб-сервиса, использующих разные JSON-представления одной и той же концепции
- Синхронный обмен сообщениями может перегрузить системы

Очередь сообщений

Очередь сообщений использует в качестве компонента инфраструктуры программный брокер сообщений (RabbitMQ, Beanstalkd, Kafka и т. д.). Для реализации связи между приложениями можно по-разному настроить очередь:

Достоинства и недостатки

Достоинства

- Независимость набора технологий, развёртывания и масштабируемости сервисов.
- Стандартный, простой и надёжный канал связи (передача текста по HTTP через порт 80).

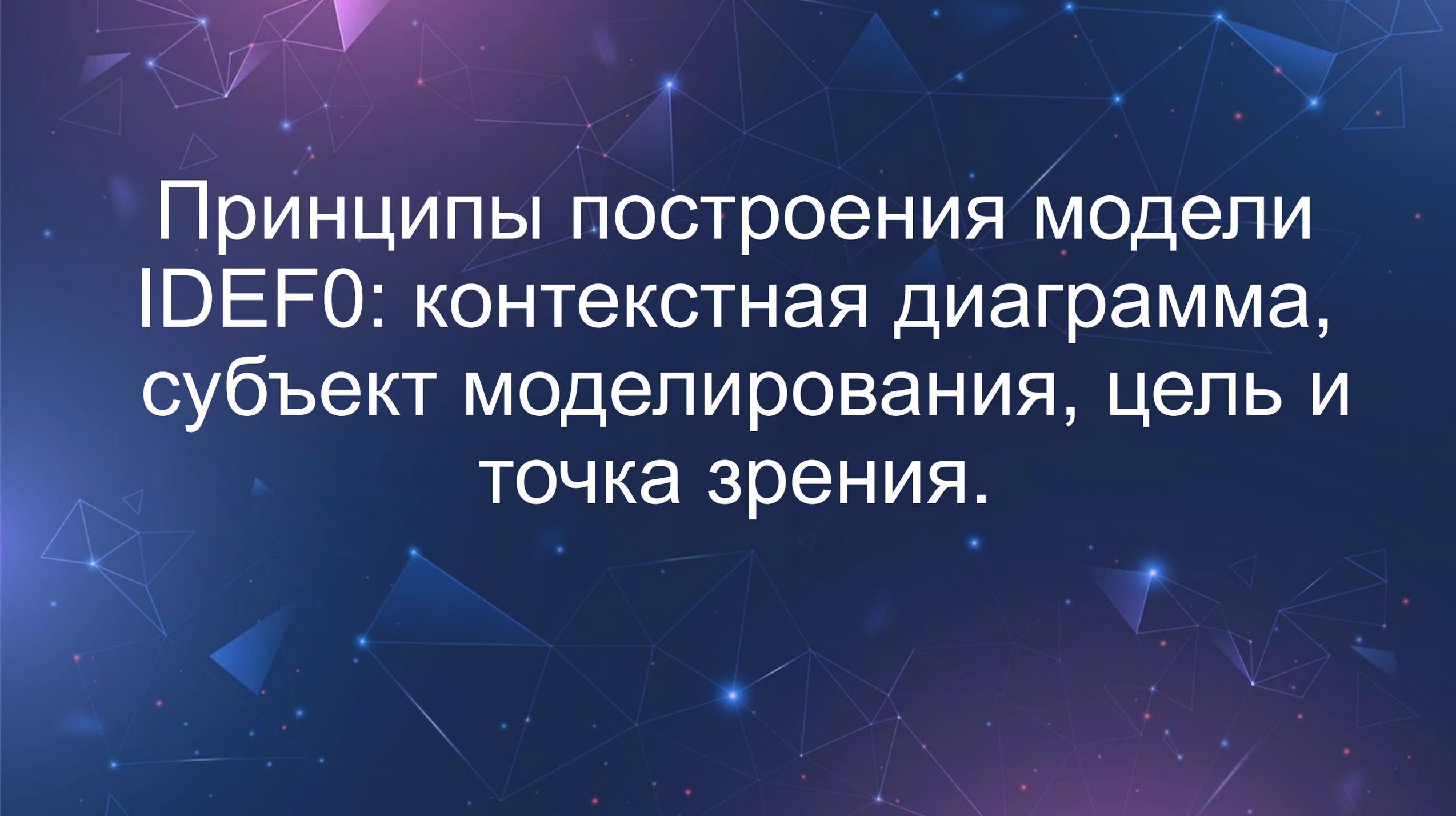
• Оптимизированный обмен сообщениями

Недостатки

- Асинхронность обмена сообщениями помогает упростить интеграцию сервисов.
- Разные веб-сервисы тяжело интегрировать из-за различий в языках передачи сообщений. Например, два веб-сервиса, использующих разные JSON-представления одной и той же концепции.

Очередь сообщений

Очередь сообщений использует в качестве компонента инфраструктуры программный брокер сообщений (RabbitMQ, Beanstalkd, Kafka и т. д.). Для реализации связи между приложениями можно по-разному настроить очередь:



Принципы построения модели
IDEF0: контекстная диаграмма,
субъект моделирования, цель и
точка зрения.

Модель IDEF0

Наиболее удобным языком моделирования бизнес-процессов является IDEF0, предложенный Дугласом Россом (SoftTech, Inc.) и называвшийся первоначально SADT - Structured Analysis and Design Technique.

В IDEF0 система представляется как совокупность взаимодействующих функциональных блоков.



Модель IDEF0

Общая методология IDEF состоит из трех частных методологий моделирования, основанных на графическом

представлении систем:

а также потоки информации и IDEF1 применяется для материальных объектов, построения информационной модели, отображающей

структуру и содержание информационных потоков,

необходимых для поддержки функций системы

меняющихся во времени поведения функций, информации и ресурсов системы

Свойства IDEF0

Графический язык

Язык обеспечивает
точное и лаконичное
описание
моделируемых
объектов

Язык облегчает
взаимодействие и
взаимопонимание
системных
аналитиков,
разработчиков и
персонала
изучаемого объекта

Язык легок и прост в
изучении и освоении.

Язык может
генерироваться
рядом
инструментальных
средств машинной
графики

Основные определения (понятия) методологии и языка IDEF0

- **Блок:** прямоугольник, содержащий имя и номер и используемый для описания функции.
- **Ветвление:** разделение стрелки на два или большее число сегментов. Может означать «развязывание пучка».
- **Внутренняя стрелка:** входная, управляющая или выходная стрелка, концы которой связывают источник и потребителя, являющиеся блоками одной диаграммы. Отличается от граничной стрелки.
- **Входная стрелка:** класс стрелок, которые отображают вход IDEF0-блока, то есть данные или материальные объекты, которые преобразуются функцией в выход. Входные стрелки связываются с левой стороной блока IDEF0.
- **Глоссарий:** список определений для ключевых слов, фраз и аббревиатур, связанных с узлами, блоками, стрелками или с моделью IDEF0 в целом.
- **Граничная стрелка:** стрелка, один из концов которой связан с источником или потребителем, а другой не присоединен ни к какому блоку на диаграмме. Отображает связь диаграммы с другими блоками системы и отличается от внутренней стрелки.
- **Декомпозиция:** разделение моделируемой функции на функции - компоненты.
- **Дерево узлов:** представление отношений между родительскими и дочерними узлами модели IDEF0 в форме древовидного графа. Имеет то же значение и содержание, что и перечень узлов.
- **Диаграмма А-0:** специальный вид (контекстной) диаграммы IDEF0, состоящей из одного блока,

Основные определения (понятия) методологии и языка IDEF0

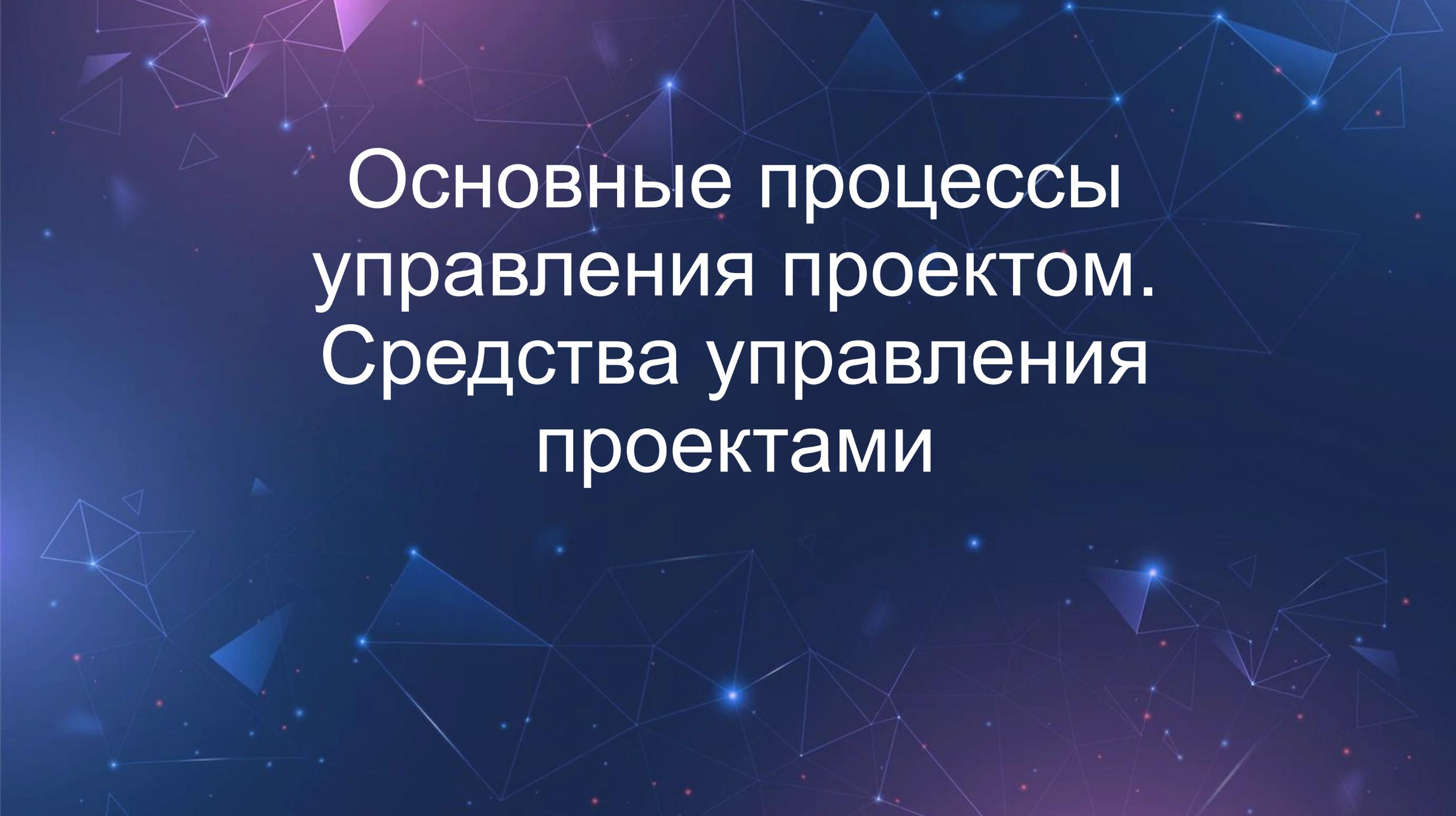
- **Диаграмма:** часть модели, описывающая декомпозицию блока.
- **Диаграмма-иллюстрация (FEO):** графическое описание, используемое, для сообщения специфических фактов о диаграмме IDEF0. При построении диаграмм FEO можно не придерживаться правила IDEF0.
- **Дочерний блок:** блок на дочерней (порожденной) диаграмме.
- **Дочерняя диаграмма:** диаграмма, детализирующая родительский (порождающий) блок.
- **Имя блока:** глагол или глагольный оборот, помещенный внутри блока и описывающий моделируемую функцию.
- **Интерфейс:** разделяющая граница,
- **Код ICOM:** аббревиатура (Input - Вход, Control - Управление, Output - Выход, Mechanism – Механизм), код, обеспечивающий соответствие граничных стрелок дочерней диаграммы со стрелками родительского блока; используется для ссылок.
- **Контекст:** окружающая среда, в которой действует функция (или комплект функций на диаграмме).
- **Контекстная диаграмма:** диаграмма, имеющая узловой номер A-n ($n \geq 0$), которая представляет контекст модели. Диаграмма A-0, состоящая из одного блока, является необходимой (обязательной) контекстной диаграммой; диаграммы с узловыми номерами A-1, A-2, ... - дополнительные контекстные диаграммы.
- **Метка стрелки:** существительное или оборот существительного, связанные со стрелкой или сегментом стрелки и определяющие их значение.

Основные определения (понятия) методологии и языка IDEF0

- **Номер блока:** число (0 - 6), помещаемое в правом нижнем углу блока и однозначно идентифицирующее блок на диаграмме.
- **Перечень узлов:** список, часто ступенчатый, показывающий узлы модели IDEF0 в упорядоченном виде. Имеет то же значение и содержание, что и дерево узлов.
- **Примечание к модели:** текстовый комментарий, являющийся частью диаграммы IDEF0 и используемый для записи факта, не нашедшего графического изображения.
- **Родительская диаграмма:** диаграмма, которая содержит родительский блок.
- **Родительский блок:** блок, который подробно описывается дочерней диаграммой.
- **Сегмент стрелки:** сегмент линии, который начинается или заканчивается на стороне блока, в точке ветвления или слияния, или на границе (несвязанный конец стрелки).
- **Семантика:** значение синтаксических компонентов языка.
- **Синтаксис:** Структурные компоненты или характеристики языка и правила, которые определяют отношения между ними.
- **Слияние:** объединение двух или большего числа сегментов стрелок в один сегмент. Может означать «развязывание пучка»
- **С-номер:** номер, создаваемый в хронологическом порядке и используемый для идентификации диаграммы и прослеживания ее истории; может быть использован в качестве

Основные определения (понятия) методологии и языка IDEF0

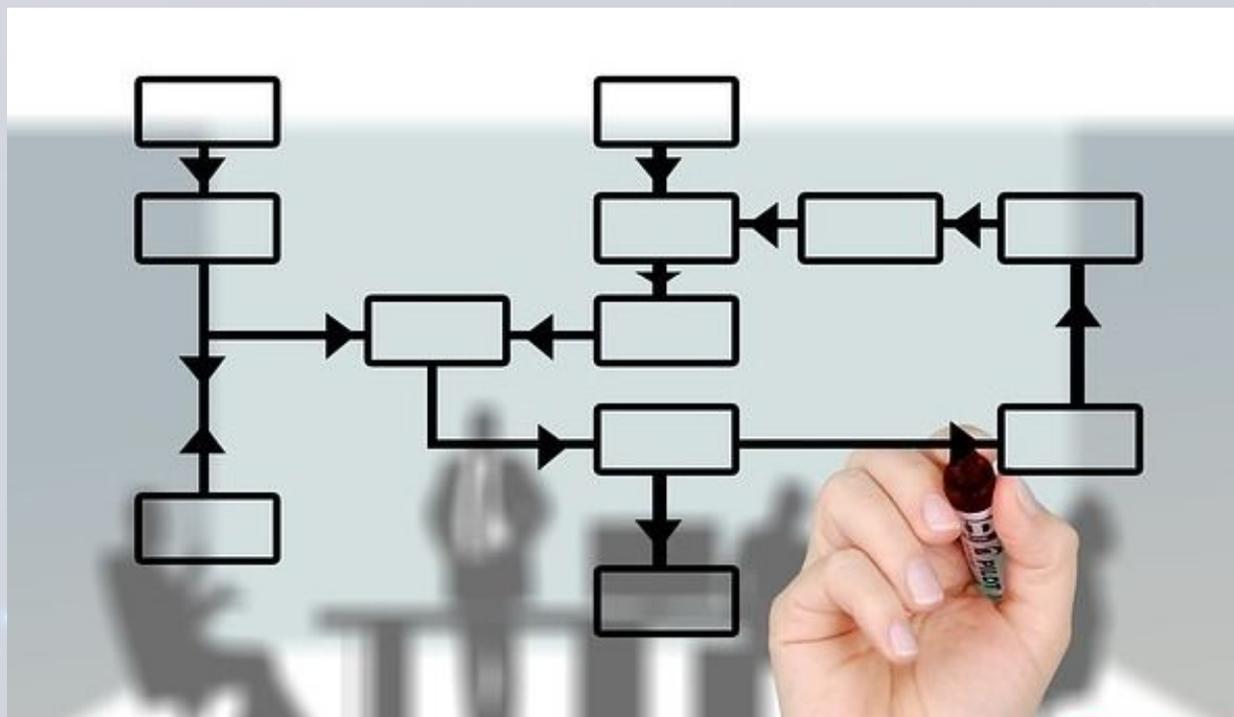
- **Стрелка:** направленная линия, состоящая из одного или нескольких сегментов, которая моделирует открытый канал или канал, передающий данные или материальные объекты от источника (начальная точка стрелки), к потребителю (конечная точка с «наконечником»). Имеется 4 класса стрелок: входная стрелка, выходная стрелка, управляющая стрелка, стрелка механизма (включает стрелку вызова).
- **Сегмент стрелки:** сегмент линии, который начинается или заканчивается на стороне блока, в точке ветвления или слияния, или на границе (несвязанный конец стрелки).
- **Семантика:** значение синтаксических компонентов языка.
- **Синтаксис:** Структурные компоненты или характеристики языка и правила, которые определяют отношения между ними.
- **Слияние:** объединение двух или большего числа сегментов стрелок в один сегмент. Может означать «развязывание пучка»
- **С-номер:** номер, создаваемый в хронологическом порядке и используемый для идентификации диаграммы и прослеживания ее истории; может быть использован в качестве



Основные процессы управления проектом. Средства управления проектами

Основные процессы управления проектом

Управление проектами — интегрированный процесс. Действия (или их отсутствие) в одном направлении обычно влияют и на остальные направления.



Процессы проекта

Процессы управления проектами — касающиеся организации и описания работ проекта (которые будут подробно описаны далее).

Процессы, ориентированные на продукт — касающиеся спецификации и производства продукта. Эти процессы определяются жизненным циклом проекта и зависят от области приложения.

Группы процессов

процессы
инициации

процессы
планирования

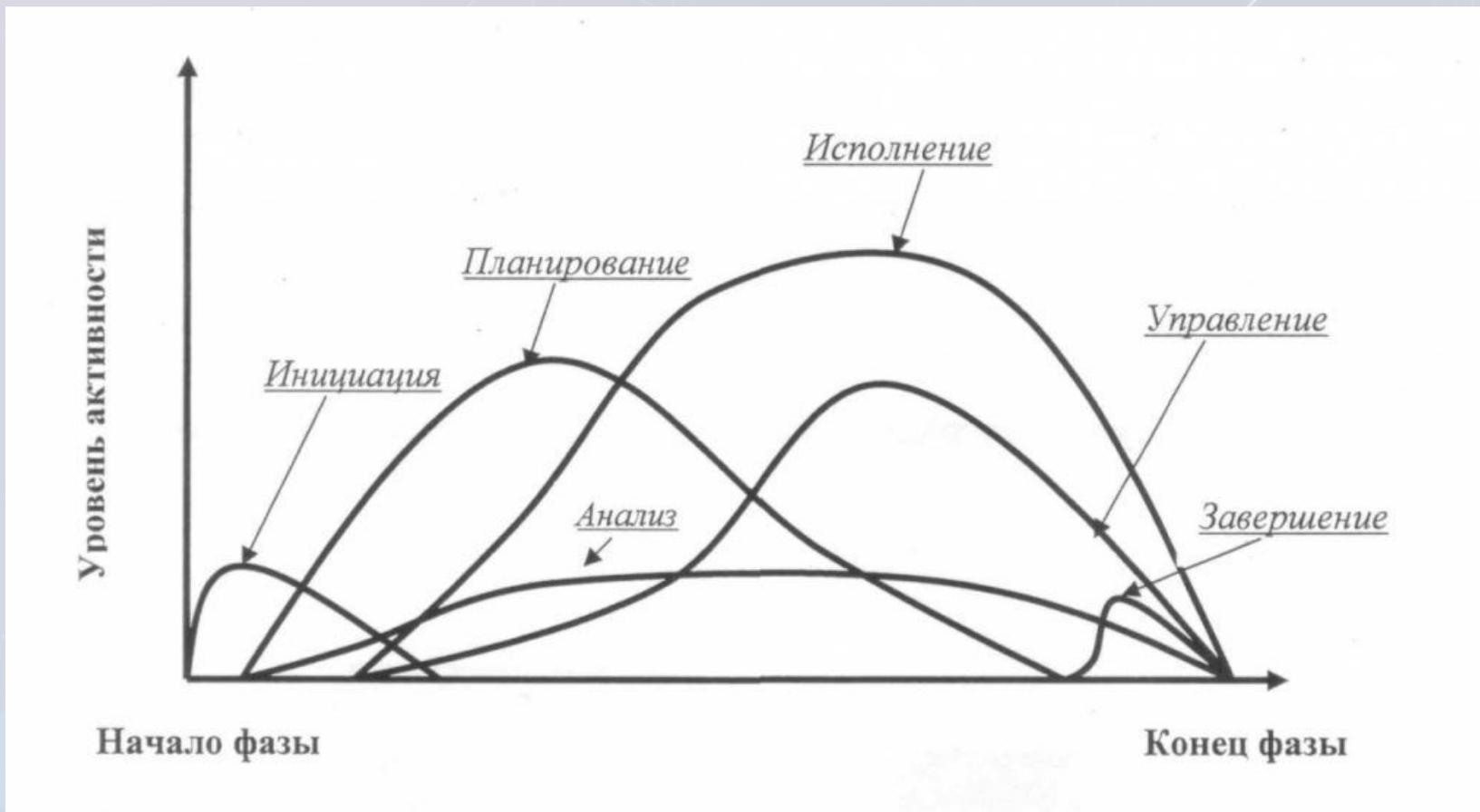
процессы
исполнения

процессы
анализа

процессы
управления

процессы
завершения

Группы процессов



Процессы инициации

Инициация включает единственный подпроцесс — Авторизацию, т.е. решение начать следующую фазу проекта.

Процессы планирования

Следует различать цели проекта и цели продукта проекта, под которым понимается продукция (или услуги), созданная или произведенная в результате исполнения проекта.

Цели продукта — это свойства и функции, которыми должна обладать продукция проекта.

Цели проекта — это работа, которую нужно выполнить для производства продукта с заданными свойствами.

Процессы планирования

Основные процессы планирования

Вспомогательные процессы планирования

Планирование целей

Декомпозиция целей

Определение состава

Определение взаимосвязей операций

Оценка длительностей или объемов работ,

Определение ресурсов проекта

Назначение ресурсов

Оценка стоимостей

Составление расписания выполнения

Оценка бюджета

Разработка плана исполнения проекта

Определение критериев успеха

Планирование качества

Планирование организации

Назначение персонала

Планирование взаимодействия

Идентификация

Оценка риска

Разработка реагирования

Планирование поставок

Подготовка условий

Процессы исполнения и контроля

Вспомогательные процессы
исполнения

Основные процессы исполнения

Исполнения плана
проекта

Учет

Подтверждение
качества

Подготовка
предложений

Выбор поставщиков

Контроль контрактов

Развитие команды
проекта

Процессы анализа

Вспомогательные процессы
анализа

Основные процессы анализа

Анализ
сроков

Анализ
стоимости

Оценка
исполнени
я

Анализ
ресурсов

Анализ
качества

Подтвержд
ение целей

Процессы управления

Управление исполнением проекта — это определение и применение необходимых управляющих воздействий с целью успешной реализации проекта

Основные процессы управления

Вспомогательные процессы

исполнения

Общее
управление
изменения
ми

Управление
ресурсами

Управлен
ие
рисками

Управлен
ие
контракта
ми

Управление
целями

Управление
качеством

Процессы завершения

Завершение проекта сопровождается следующими процессами:

- закрытие контрактов — завершение и закрытие контрактов, включая разрешение всех возникших споров.
- административное завершение — подготовка, сбор и распределение информации, необходимой для формального завершения проекта.

Средства управления проектами

Управление проектами – это управление и организация всего, что нужно для достижения цели – вовремя и в рамках бюджета

Классический
проектный
менеджмент

Agile

Scrum

Lean

Kanban

Six
Sigma

PRINC
E2

Классический проектный менеджмент

- 5 этапов традиционного менеджмента
 - Этап 1. Инициация
 - Этап 2. Планирование
 - Этап 3. Разработка
 - Этап 4. Реализация и тестирование
 - Этап 5. Мониторинг и завершение проекта



Сильные и слабые стороны классического проектного менеджмента

Agile

Инициация и верхнеуровневое планирование проводятся для всего проекта, а последующие этапы: разработка, тестирование и прочие проводятся для каждого мини-проекта отдельно. Это позволяет передавать результаты этих мини-проектов, так называемые, инкременты, быстрее, а приступая к новому подпроекту (итерации) в него можно внести изменения без больших затрат.



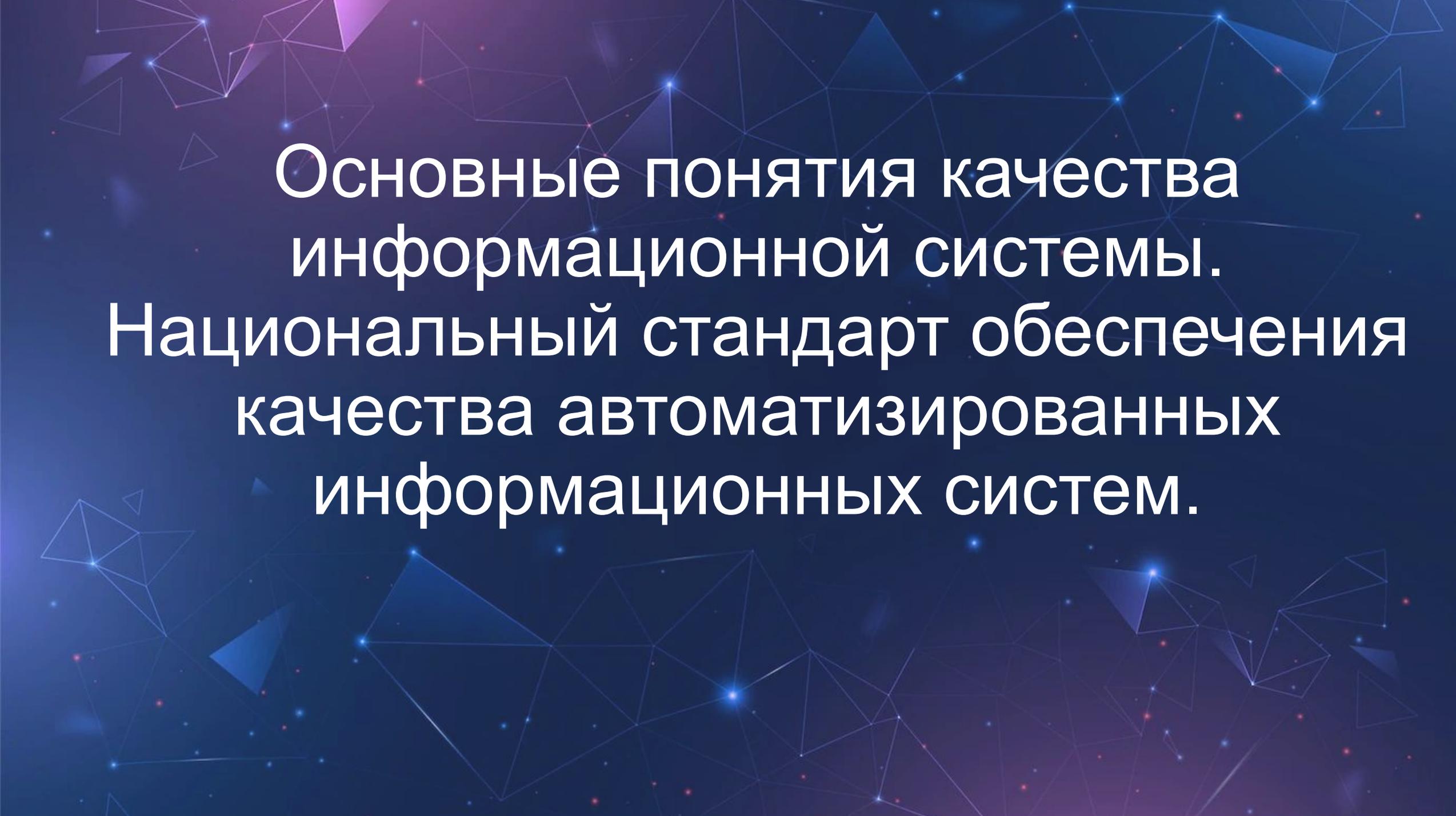
Сильные и слабые стороны Agile

Scrum

- Основная структура процессов Scrum вращается вокруг 5 основных встреч: упорядочивания беклога, планирования Спринта, ежедневных летучек, подведения итогов Спринта и ретроспективы Спринта.
 - Встреча по упорядочиванию беклога
 - Планирование Спринта
 - Ежедневные летучки
 - Подведение итогов Спринта
 - Ретроспектива Спринта



Сильные и слабые стороны Scrum



Основные понятия качества
информационной системы.
Национальный стандарт обеспечения
качества автоматизированных
информационных систем.

Качество информационной системы

Качество информационной системы — это совокупность свойств системы, обуславливающих возможность ее использования для удовлетворения определенных в соответствии с ее назначением потребностей.

- Основные показатели качества информационных систем
 - Надежность — свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения.
 - Достоверность функционирования — свойство системы, обуславливающее безошибочность производимых ею преобразований информации.
 - Безопасность информационной системы — свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации
 - Эффективность — это свойство системы выполнять поставленную цель в заданных условиях использования и с определенным качеством.

•Надежность информационных систем

- Надежность — важнейшая характеристика качества любой системы, поэтому разработана специальная теория — теория надежности.
- Надежность — характеристика временная, она может быть ориентирована либо в прошлое, либо в будущее время и не допускает «точечных» во времени оценок.
- Надежность — комплексное свойство системы; оно включает в себя более простые свойства, такие как безотказность, ремонтпригодность, долговечность и т. д.

Надежность информационных систем

Безотказность — свойство системы сохранять работоспособное состояние в течение некоторого времени или наработки (наработка — продолжительность или объем работы системы).

Ремонтопригодность — свойство системы, заключающееся в приспособленности к предупреждению и обнаружению причин возникновения отказов, повреждений и поддержанию и восстановлению работоспособного состояния путем проведения технического обслуживания и ремонтов.

Долговечность — свойство системы сохранять при установленной системе технического обслуживания и ремонта работоспособное состояние до наступления предельного состояния, то есть такого момента, когда дальнейшее использование системы по назначению недопустимо или нецелесообразно.

Надежность информационных систем

Отказом называют полную или частичную потерю работоспособности системы или ее элемента.

Отказ

Внезапный

Зависимый

Устойчивы

Аппаратны

и

и

Полный и

и

и

постепенн

независим

частичный

самоустра

программн

ый

ый

няющийся

ый

Основные показатели надежности

Показатель надежности — это количественная характеристика одного или нескольких свойств, определяющих надежность системы. В основе большинства показателей надежности лежат оценки наработки системы, то есть продолжительности или объема работы, выполняемой

Показатели
безотказности

Показатели
ремонтотпригодности
и

Показатели
долговечности

Показатели безотказности

Вероятность безотказной работы — вероятность того, что в пределах заданной наработки отказ системы не возникнет.

Вероятность отказа — обратная величина, вероятность того, что в пределах заданной наработки отказ системы возникнет.

Средняя наработка до отказа — математическое ожидание наработки системы до первого отказа (существенно для невосстанавливаемых систем).

Средняя наработка на отказ (T_0) — отношение наработки восстанавливаемой системы к математическому ожиданию числа ее отказов в пределах этой наработки (имеет смысл только для восстанавливаемых систем).

Интенсивность отказов — условная плотность вероятности возникновения отказа невосстанавливаемой системы, определяемая для рассматриваемого момента времени при условии, что до этого момента отказ не возник.

Параметр потока отказов ($X(t)$) — отношение среднего числа отказов для восстанавливаемой системы за произвольно малую ее наработку к значению этой наработки.

Показатели ремонтпригодности

Вероятность восстановления работоспособного состояния — вероятность того, что время восстановления работоспособного состояния не превысит заданного.

Среднее время восстановления работоспособного состояния (T_v) — математическое ожидание времени восстановления работоспособного состояния системы.

Показатели долговечности

Средний ресурс — математическое ожидание наработки системы от начала ее эксплуатации или ее возобновления после ремонта до перехода в предельное состояние.

Срок службы (Тсс) — календарная продолжительность от начала эксплуатации системы или ее возобновления после ремонта до перехода в предельное состояние.

Комплексные показатели надежности

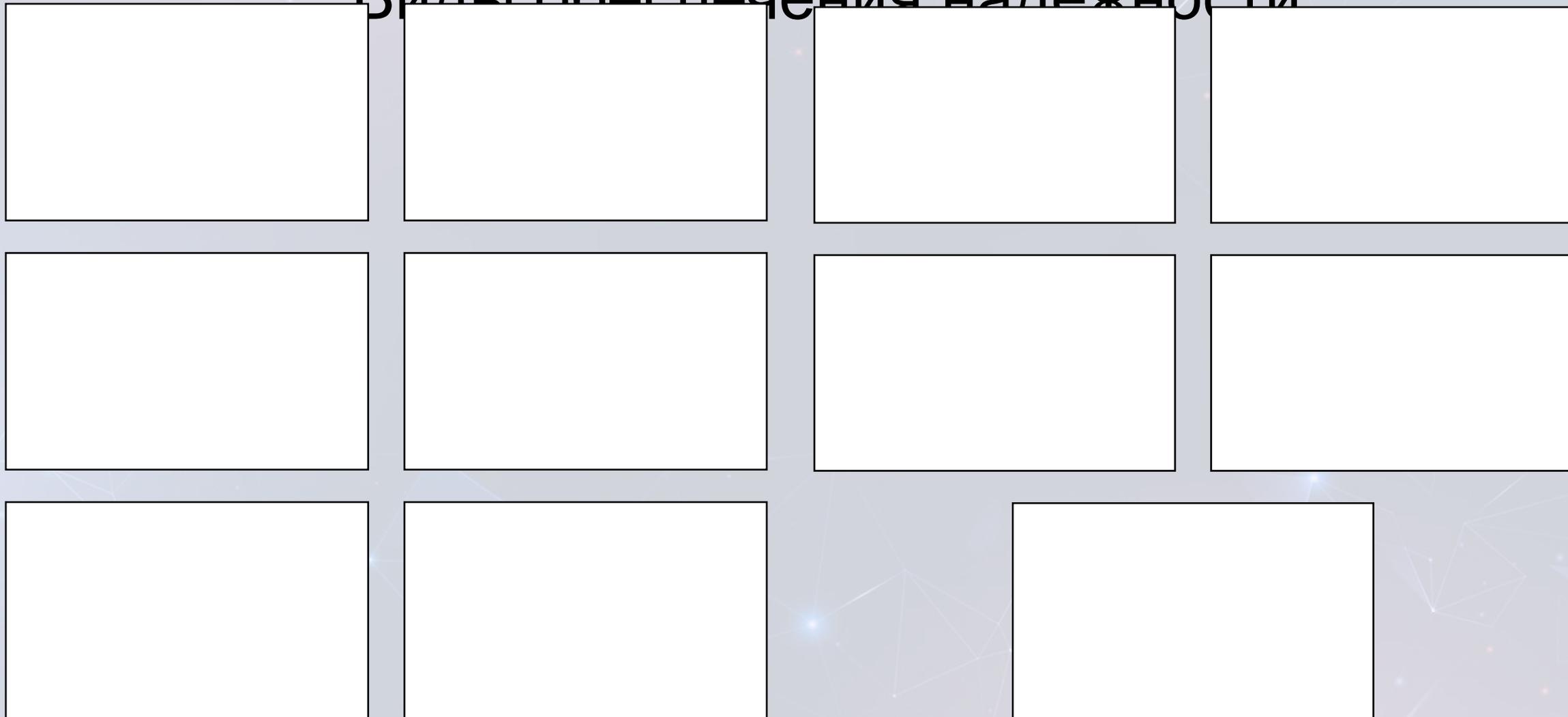
- Коэффициент готовности (K_g) — вероятность того, что система окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых периодов, в течение которых применение системы по назначению не предусматривается
- Коэффициент оперативной готовности — вероятность того, что система окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых периодов, в течение которых применение системы по назначению не предусматривается, и начиная с этого момента будет работать безотказно в течение заданного времени.

Комплексные показатели надежности

- Коэффициент технического использования — отношение математического ожидания интервалов времени пребывания системы в работоспособном состоянии за некоторый период эксплуатации к сумме математических ожиданий интервалов времени пребывания системы в работоспособном состоянии, простоев, обусловленных техническим обслуживанием, и ремонтов за тот же период эксплуатации
- Коэффициент сохранения эффективности — отношение значения показателя эффективности за определенную продолжительность эксплуатации к номинальному значению этого показателя, вычисленному при условии, что отказы в системе в течение того же периода эксплуатации не возникают.

Обеспечение надежности функционирования ИС

Виды обеспечения надежности



Достоверность информационных систем

Достоверность функционирования — это свойство системы, обуславливающее безошибочность производимых ею преобразований информации.

Достоверность информации — это свойство информации отражать реально существующие объекты с необходимой точностью.

$$D = P \{ O[\Delta n] \}$$

где — O реальная точность отображения параметра, $[\Delta n]$ — диапазон необходимой точности отображения параметра.

Показатели достоверности информации

Обеспечение достоверности информации

Контроль — процесс получения и обработки информации с целью оценки соответствия фактического состояния объекта предъявляемым к нему требованиям и выработки соответствующего управляющего решения.

- Классификация методов контроля достоверности по назначению
 - Профилактический контроль и, например, одна из наиболее распространенных его форм — тестовый контроль, предназначены для выявления состояния системы в целом и отдельных ее звеньев до включения системы в рабочий режим.
 - Рабочий контроль, или контроль в рабочем режиме, производится в процессе выполнения системой возложенных на нее функций.
 - Генезисный контроль проводится для выяснения технического состояния системы в прошлые моменты времени с целью определения причин сбоев и отказов системы, сбора статистических данных об ошибках, их характере, величине и последствиях (экономических потерях) этих ошибок для ИС.

Обеспечение достоверности информации

Семантический
контроль оценивает

Предметический

изменения.

Обеспечение достоверности информации

Программный
контроль

целом.

Обеспечение достоверности информации

Безопасность информационных систем

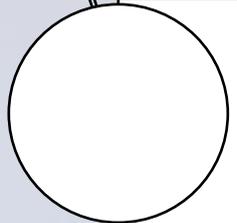
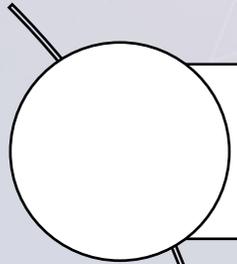
Безопасность информационной системы — свойство, заключающееся в способности системы обеспечить конфиденциальность и целостность информации, то есть защиту информации от несанкционированного доступа, обращения на ее раскрытие, изменение или разрушение.

угроза целостности

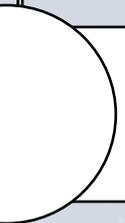
судно-операционная

другую.

Средства обеспечения информационной безопасности

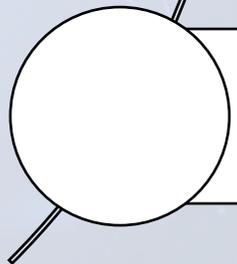


технологические методы, включающие в себя технологии выполнения



информации

аппаратные методы, реализующие физическую защиту систем, от



сетевых компонентов и т. д.

Эффективность информационных систем

Эффективность системы — это свойство системы выполнять поставленную цель в заданных условиях использования и с определенным качеством.

Локальные показатели эффективности

простота и

гибкость

--	--	--	--

Показатели прагматической эффективности

Свойства,
влияющие на
оптимальность
функционирования системы

достоверность преобразования

безопасность информационных систем

точность вычислений и преобразования информации

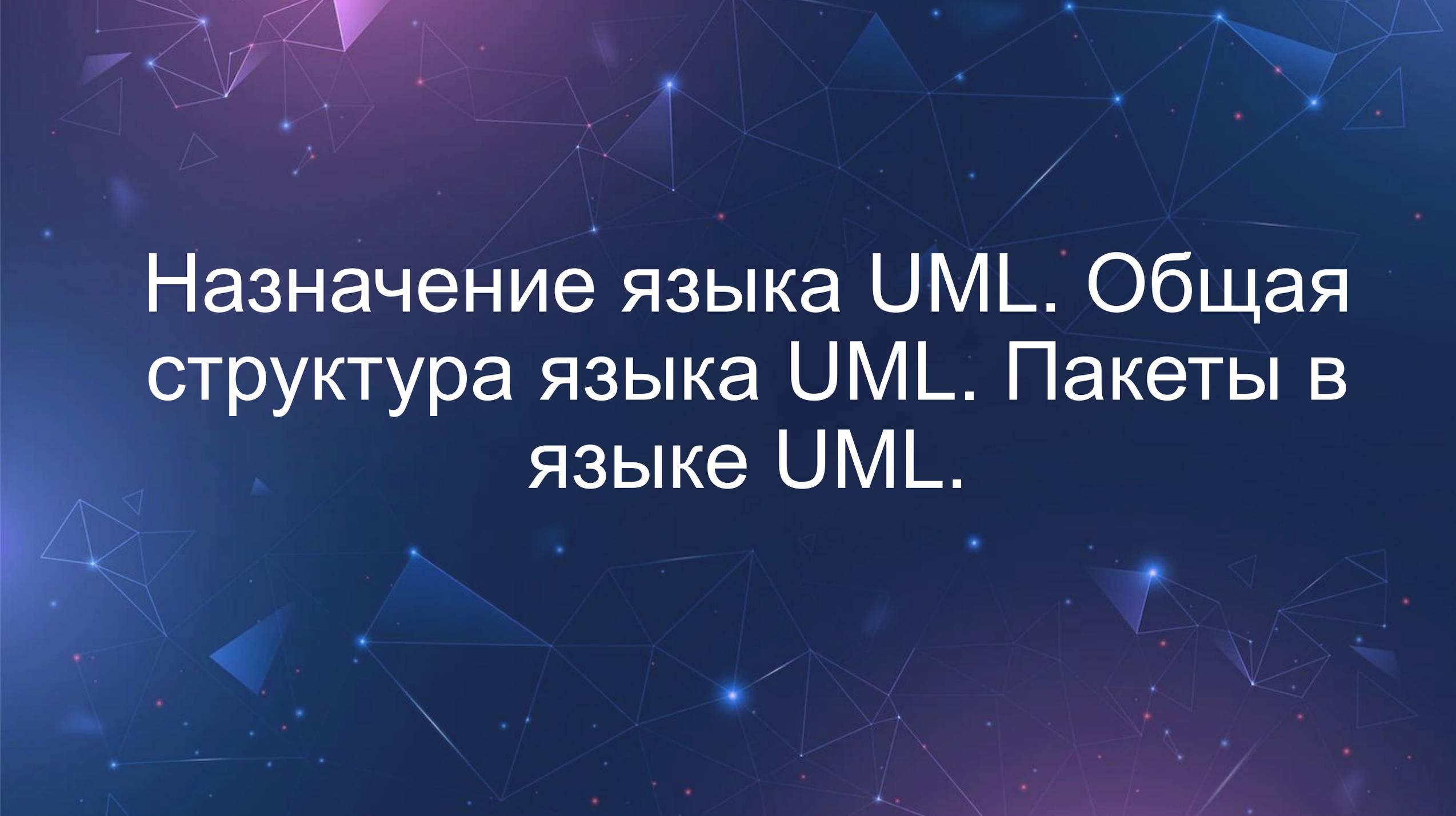
полнота формирования системой результирующей информации

оперативность, показывающая, насколько быстро в системе формируется результирующая информация, не устарела ли она
своевременность

Показатели технико-эксплуатационной эффективности

Показатели технической эффективности должны оценивать техническое совершенство информационной системы как эрготехнической системы при работе ее в различных режимах, оценивать научно-технический уровень организации и функционирования этой системы.

Научно-технический уровень ИС характеризуется системой показателей, отражающих степень соответствия ее технико-эксплуатационных характеристик современным достижениям науки и техники, научно-технического прогресса.



Назначение языка UML. Общая структура языка UML. Пакеты в языке UML.

Назначение языка UML

Язык UML представляет собой общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов



Назначение языка UML

Предоставить в распоряжение пользователей легко воспринимаемый и выразительный язык визуального моделирования, специально предназначенный для разработки и документирования моделей сложных систем самого различного целевого назначения

Снабдить исходные понятия языка UML возможностью расширения и специализации для более точного представления моделей систем в конкретной предметной области

Описание языка UML должно поддерживать такую спецификацию моделей, которая не зависит от конкретных языков программирования и инструментальных средств проектирования программных систем

Описание языка UML должно включать в себя семантический базис для понимания общих особенностей ООАП

Общая структура языка UML

Семантика языка UML. Представляет собой некоторую метамодель, которая определяет абстрактный синтаксис и семантику понятий объектного моделирования на языке UML.

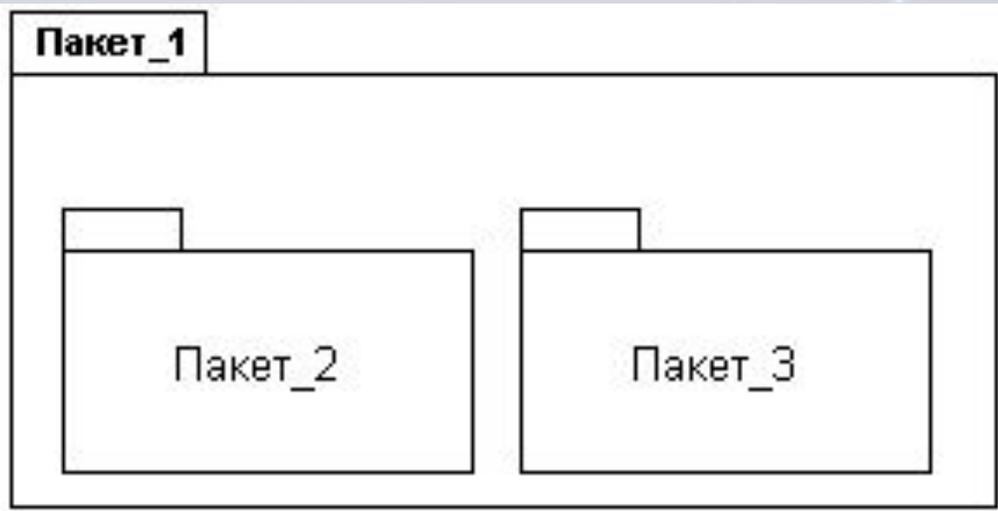
Нотация языка UML. Представляет собой графическую нотацию для визуального представления семантики языка UML.

Пакеты в языке UML

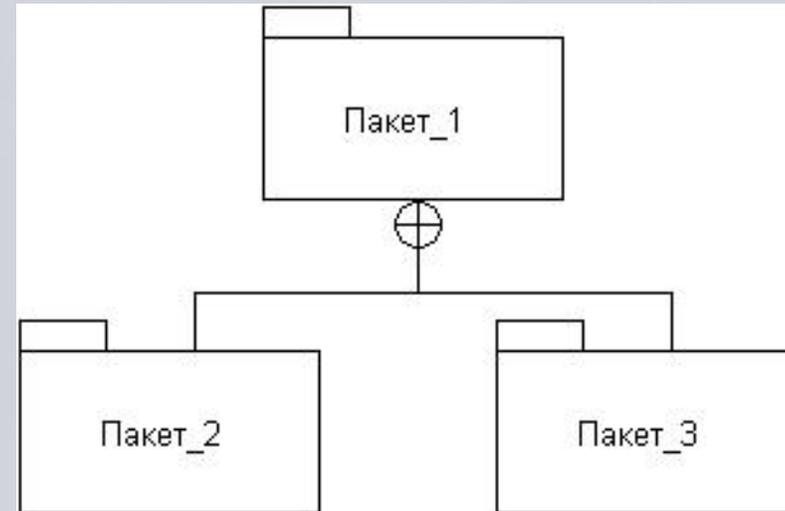
Пакет – основной способ организации элементов модели в языке UML. Каждый пакет владеет всеми своими элементами, т. е. теми элементами, которые включены в него. Про соответствующие элементы пакета говорят, что они принадлежат пакету или входят в него.



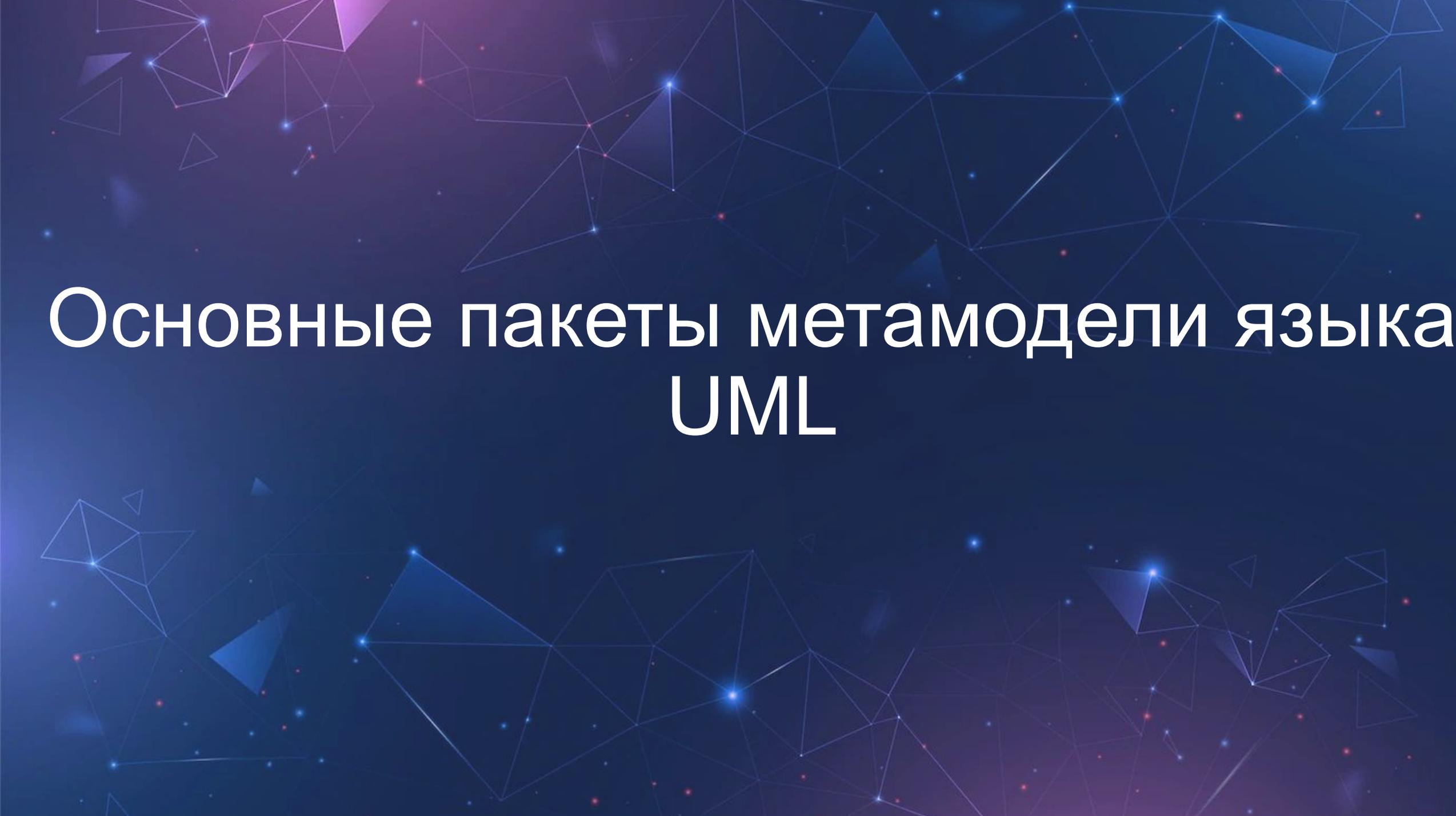
Пакеты в языке UML



Графическое изображение
вложенности пакетов друг в
друга



Графическое изображение
вложенности пакетов друг в
друга с помощью явной
визуализации отношения
включения

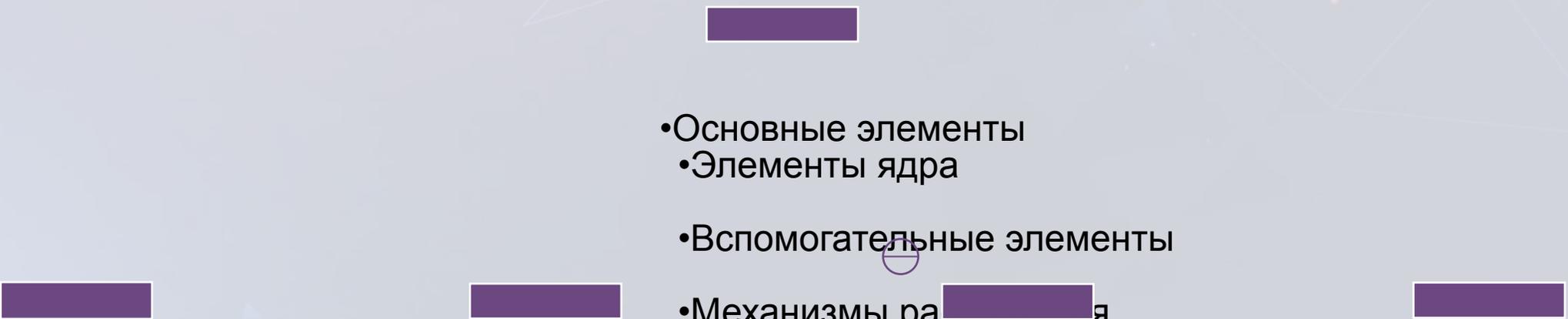


Основные пакеты метамодели языка UML

Основные пакеты метамодели языка UML

- Язык UML
 - Основные элементы
 - Элементы поведения
 - Общие механизмы

Основные пакеты метамодели языка UML

- 
- Основные элементы
 - Элементы ядра
 - Вспомогательные элементы
 - Механизмы расширения
 - Типы данных

Подпакеты пакета Основные элементы языка UML

Пакет Элементы ядра

- В этот пакет входят основные метаклассы языка UML: класс (Class), атрибут (Attribute), ассоциация (Association), ассоциация-класс (AssociationClass), конец ассоциации (AssociationEnd), свойство поведения (BehavioralFeature), классификатор (Classifier), ограничение (Constraint), тип данных (DataType), зависимость (Dependency), элемент (Element), право на элемент (ElementOwnership), свойство (Feature), обобщение (Generalization), элемент отношения обобщения (GeneralizableElement), интерфейс (Interface), метод (Method), элемент модели (ModelElement), пространство имен (Namespace), операция (Operation), параметр (Parameter), структурное свойство (StructuralFeature), правила правильного построения выражений (Well-formedness rules).

Подпакеты пакета Основные элементы языка UML

Пакет Вспомогательные элементы

- В этот пакет входят следующие метаклассы: связывание (Binding), комментарий (Comment), компонент (Component), узел (Node), презентация (Presentation), уточнение (Refinement), цепочка зависимостей (Trace), потребление (Usage), элемент представления (ViewElement), зависимость (Dependency), элемент модели (ModelElement), правила правильного построения выражений (Well-formedness rules).

Подпакеты пакета Основные элементы языка UML

Пакет Механизмы расширения

- Механизмами являются: ограничение (Constraint), стереотип (Stereotype) и помеченное значение (TaggedValue).

Механизмы расширения языка UML предназначены для выполнения следующих задач:

- Уточнения существующих модельных элементов при разработке моделей на языке UML.
- В спецификации самого языка UML для определения стандартных компонентов, которые либо не являются достаточно интересными, либо сложны для непосредственного определения в качестве элементов метамодели UML.
- Определения таких расширений языка UML, которые зависят от специфики моделируемого процесса или от языка реализации программного кода.

Подпакеты пакета Основные элементы языка UML

Пакет Типы данных

- Для определения различных типов данных в языке UML используются как простые конструкции: целое число (Integer), строка (String), имя (Name), Булев (Boolean), время (Time), кратность (Multiplicity), тип видимости (VisibilityKind), диапазон кратности (MultiplicityRange), так и более сложные: выражение (Expression), булевское выражение (BooleanExpression), тип агрегирования (AggregationKind), тип изменения (ChangeableKind), геометрия (Geometry), отображение (Mapping), выражение-процедура (ProcedureExpression), тип псевдосостояния (PseudostateKind), выражение времени (TimeExpression), непрерываемый (Uninterpreted).