# **Понятие системы (в контексте определений ГОСТ Р 57193-2016 или ГОСТ Р ИСО/МЭК 12207-2010). Структура системы управления (кибернетический подход)*.* Признаки систем.**

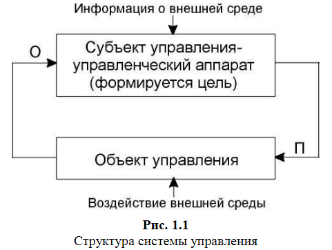
Система - комбинация взаимодействующих элементов, организованных для достижения одной или нескольких поставленных целей.

Системный элемент может представлять собой технические и программные средства, данные, людей, процессы, процедуры, средства, материалы и природные объекты или любые их сочетания. Система конкретизируется в процессе рассмотрения ее основных свойств. Можно выделить некоторые наиболее характерные признаки систем:

* целостность: «Система есть абстрактная сущность, обладающая целостностью и определенная в своих границах». Каждый элемент вносит свой вклад в осуществление целей системы. Между элементами системы существуют устойчивые связи, превосходящие по силе связи этих элементов с окружающей средой. Такие связи называются системообразующими;
* эмерджентность - появление у системы свойств, не присущих элементам системы; принципиальная несводимость свойств системы к сумме свойств составляющих ее компонентов. «Возможности системы превосходят сумму возможностей составляющих ее частей; общая производительность или функциональность системы лучше, чем у простой суммы элементов»;
* гомеостаз - «обеспечение устойчивого функционирования системы и достижения общей цели», равновесие. Для системы характерна возможность находиться в некотором устойчивом состоянии. Если в результате внешних воздействий система была выведена из устойчивого состояния, она способна вернуться в такое состояние;
* адаптивность «к изменениям внешней среды и управляемость посредством воздействия на элементы системы». Система способна приспосабливаться к изменениям окружающей среды посредством модификации своей структуры или поведения, чтобы сохранить или улучшить свои свойства;
* обучаемость путем «изменения структуры системы в соответствии с изменением целей системы».

Систему, реализующую функции управления, называют системой управления. Наука, изучающая управляемые системы или системы управления – от систем простых механизмов до сложных социальных систем, - называется кибернетикой.

В соответствии с кибернетическим подходом система управления представляет собой совокупность объекта управления, например предприятия, и субъекта управления - управленческого аппарата (рис. 1.1). Управленческий аппарат объединяет в себе сотрудников, формирующих цели, разрабатывающих планы, вырабатывающих требования к принимаемым решениям, а также контролирующих их выполнение. В задачу же объекта управления входит выполнение планов, выработанных управленческим аппаратом, т. е. реализация той деятельности, для которой создавалась система управления.

Оба компонента системы управления связаны прямой (П) и обратной (О) связями. Прямая связь выражается потоком директивной информации, направляемой от управленческого аппарата к объекту управления, а обратная представляет собой идущий в обратном направлении поток отчетной информации о выполнении принятых решений.

# **Понятие информационной системы и информационных технологий (в контексте определений Федерального закона от 27.07.2006 N 149-ФЗ и ГОСТ 33707-2016). Состав основных компонентов информационных систем (виды обеспечения).**

В Федеральном законе от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» содержится следующее определение ИС: «информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств».

В свою очередь, информационные технологии определяются как процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов.

В соответствии с рассмотренными выше определениями выделим основные функции ИС:

* сбор и хранение больших объемов информации;
* обработка информации в ходе решения задач;
* отображение информации в виде, удобном для изучения и принятия решений.

Основными составляющими любой ИС являются как минимум:

* БД как совокупность взаимосвязанных упорядоченных определенным образом данных;
* программные модули, предназначенные для обработки данных;
* пользовательский интерфейс.

Изображение выглядит как стол

Автоматически созданное описание

# **Понятие «Жизненный цикл информационной системы (ИС)». Модели жизненного цикла ИС (базовые), их сущность, особенности становления и развития.**

Жизненный цикл ИС можно представить как ряд событий, происходящих с системой в процессе ее создания и использования. Модель ЖЦ отражает различные состояния системы, начиная с момента возникновения необходимости в данной ИС 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.

Одной из первых моделей жизненного цикла, достаточно широко применявшейся при разработке ИС и программных комплексов, была каскадная (водопадная) модель. Эта модель предусматривает наличие нескольких стадий, заключающихся в выполнении определенного перечня работ, что облегчает планирование сроков реализации стадий и их стоимости. Каждая из стадий выполняется последовательно, в строго определенном порядке. Переход к очередной стадии осуществляется только по завершении всех необходимых работ на предыдущей стадии. Завершение стадии подробно документируется, что снимает ограничение на неизменность коллектива разработчиков ИС. В каскадной модели жизненного цикла ИС не предусмотрены обратные связи между отдельными этапами. Это обстоятельство является существенным недостатком подобных моделей. В каскадной модели предполагается, что все проблемные ситуации, выявленные на ранних стадиях, могут быть успешно решены в дальнейшем, однако это не всегда осуществимо. Многие ошибки анализа, проектирования и реализации могут быть выявлены только на последующих этапах или только в ходе эксплуатации. Устранение таких ошибок представляет трудоемкую задачу, связанную с дополнительными неоправданными издержками.

Попытки устранить недостатки каскадной модели, прежде всего такие, как отсутствие обратных связей с предшествующими стадиями, привели к ее модификации: была предложена новая модель жизненного цикла ИС - поэтапная с промежуточным контролем.

Такая модель предусматривает в случае необходимости, до ввода ИС эксплуатацию, возможность возвращения к предшествующим стадиям для устранения выявленных на предыдущих стадиях упущений. С одной стороны, это позволяет путем межэтапных корректировок учитывать реально существующее взаимовлияние результатов разработки с целью повышения качества и функциональности ИС. С другой стороны, сроки разработки ИС вследствие многочисленных изменений ранее принятых решений могут существенно затянуться, стоимость проекта значительно возрасти. При этом не всегда можно утверждать, что постоянные уточнения и доработки обеспечат более высокие конечные функциональные характеристики системы, отвечающие реальным потребностям заказчика.

В приведенных выше моделях жизненного цикла заказчик может увидеть реализованные возможности ИС только после сдачи ее в эксплуатацию. Учитывая, что сроки разработки реальных систем могут составлять годы, то в отдельных случаях ИС к моменту завершения проекта уже могут перестать удовлетворять потребителей по причине изменившихся условий. Это является существенным недостатком всех приведенных выше моделей жизненного цикла.

Отмеченный недостаток позволяет устранить спиральная модель жизненного цикла. Отражая эволюционную технологию проектирования «сверху вниз», она представляет жизненный цикл ИС в виде множества итераций, когда последовательно уточняются требования заказчика, наращиваются и детализируются функциональные возможности разрабатываемой ИС. Обеспечивая более высокую эффективность и конечную реализуемость разрабатываемых проектов, спиральная модель обнаружила и недостатки, связанные прежде всего с большим количеством возвратов к ранее пройденным этапам вследствие постоянных уточнений и изменений требований к ИС:

* повышение сложности управления проектом;
* проблемы в согласовании связей по управлению и данными между различными фрагментами проекта;
* трудности контроля и управления сроками разработки;
* затруднения в определении момента окончательного завершения работ этапа для перехода к следующему этапу;
* необходимость постоянного контактирования заказчика с разработчиками с целью системного анализа и оценки новых принимаемых решений.

Несмотря на отмеченные недостатки, достоинства спиральной модели превалировали, поэтому на протяжении многих лет использовалась именно эта модель, подвергаясь уточнению и модификации.

# **Семейство стандартов структурного моделирования IDEF: первое поколение (функциональный, информационный, динамический методы моделирования).**

Методы IDEF (Integrated DEFinition) — это семейство нотаций и методов моделирования, первоначально разработанных США как часть методологии описания рабочих процессов и информационных систем, в настоящее время находятся в свободном доступе. IDEF широко применяется в течение многих лет и реализован во многих средствах моделирования. В основу была положена технология структурированного анализа и проектирования SADT.

Методология IDEF это наиболее глубоко проработанная и обширная методология, которая позволяет описывать не только бизнес-процессы, но и функциональные блоки (например, маркетинг и финансы), различные объекты в компании и действия над ними, а также состояние и динамику развития бизнес-единиц компании в целом.

Первое поколение методов IDEF:

* IDEF0 — методология функционального моделирования (Function Modeling);
* IDEF1 — методология моделирования информационных потоков (Information Modeling);
* IDEF2 — методология динамического моделирования развития систем (Simulation Modeling);

# **Содержание стадий жизненного цикла информационных систем (с учетом ГОСТ Р 57193-2016 "Системная и программная инженерия. Процессы жизненного цикла систем").**

Каждая система имеет жизненный цикл. Система развивается через свой жизненный цикл как результат действий, выполняемых и управляемых специалистами организации, используя для этих действий процессы. Жизненные циклы изменяются согласно природе, целям, использованию системы и преобладающим обстоятельствам. Каждая стадия имеет свои цели и вклад в жизненный цикл и рассматривается при планировании и осуществлении всего жизненного цикла системы. Стадии представляют собой главные периоды жизненного цикла, связанные с системой, и касаются состояния в описании системы или непосредственно состояния самой системы. Стадии описывают главное развитие и контрольные точки достижения системы в ее жизненном цикле. Они дают развитие прохождениям решений по жизненному циклу. Эти прохождения решений используются организациями с тем, чтобы понимать и управлять изначальной неопределенностью и рисками, связанными с затратами, сроками и функциональными возможностями при создании или использовании системы. Тем самым стадии предоставляют организациям такую структуру, в пределах которой у руководства организации существует представление наивысшего уровня и возможности управления проектом и техническими процессами.

В [19] типичные стадии жизненного цикла системы включают замысел, разработку, производство, применение, поддержку и выведение из эксплуатации.

Организации используют стадии различным образом так, чтобы удовлетворить стратегиям разнообразного бизнеса и снизить риски. Использование стадий одновременно и в различных последовательностях может привести к формам жизненного цикла с отчетливо различными характеристиками.

Выделяют следующие **стадии ЖЦ ИС: анализ, проектирование, реализация проекта, внедрение** (ввод в эксплуатацию ИС), **сопровождение** (эксплуатация, наращивание возможностей – модернизация), **вывод из эксплуатации** (замена).

анализ – определение того, что должна делать система, для решения каких информационных проблем (потребностей) она предназначения, т.е. анализ − это стадия формирования требований на разрабатываемую систему (концептуальная разработка);

проектирование (логическая разработка) – определение того, как система будет делать то, что она должна делать; проектирование – это, прежде всего, преобразование требований к ИС в формальные (детальные) спецификации подсистем, функциональных компонентов и способов их взаимодействия в системе, иначе эту стадию определяют как стадию «формирования проектных решений»;

реализация (физическая разработка) – создание функциональных компонентов и подсистем по отдельности, соединение подсистем в единое целое и тестирование – проверка функционального и параметрического соответствия системы показателям, определенным на этапе анализа;

внедрение – установка и ввод системы в действие (внедрение разработки в бизнес-систему);

эксплуатация и сопровождение – обеспечение штатного процесса эксплуатации системы на предприятии заказчика;

ликвидация – вывод системы из эксплуатации в целях построения более эффективной ИС.

# **Сущность методологии структурного анализа и проектирования SADT. Принципы функционального моделирования.**

Методология SADT (Structured Analysis and Design Technique технология структурного анализа и проектирования) была разработана специально для описания искусственных систем средней сложности. Она успешно применялась и применяется для описания различных технологических процессов, процессов управления, планирования и проектирования информационных систем. В основе данной методологии лежит концепция графического описания систем и системного моделирования. Она позволяет рассматривать систему с помощью описания ее функций или с помощью описания ее объектов (данных). Поэтому различают два типа моделей в методологии SADT, а именно:

* функциональные SADT — модели, ориентированные на функции описываемой системы;
* SADT — модели данных, ориентированные на объекты описываемой системы.

Функциональная модель организации представляет собой систему функций (бизнес-процессов), осуществляемых в рамках данной организации. Связи между функциями (процессами) отражаются посредством движения объектов между ними, т.е. с помощью моделей данных. Целью построения модели SADT является получение ответов на вопросы, которые ставятся перед лицом, осуществляющим моделирование. При использовании в моделировании бизнес-процессов методологии SADT необходимо соблюдать следующие принципы построения моделей:

* + модель может иметь только единственный субъект;
  + у модели может быть только одна точка зрения;
  1. модель бизнес-процесса — это набор взаимосвязанных диаграмм.

В рамках функционального моделирования SADT субъектом является сама описываемая система, которая, как правило, взаимодействует с окружающей средой. Поэтому важно с самого начала анализа определить границы системы, т.е. установить, что именно является субъектом моделирования. Таким образом, определение границ моделируемой системы позволяет избежать включения в анализ посторонних субъектов и сосредоточить внимание именно на цели моделирования.

Другим важным требованием данной методологии является определение позиции, с которой рассматривается данная система. Иначе, нужно сформулировать, с какой точки зрения создается модель. Часто бывает так, что из множества точек зрения только одна может дать описание, удовлетворяющее поставленным ранее целям и вопросам. Поэтому при определении точки зрения не следует забывать о цели моделирования.

После определения цели моделирования, границы моделируемой системы и точки зрения можно переходить к непосредственному построению взаимосвязанных графических описаний с верхнего уровня и до простейших операций, осуществляемых в ходе функционирования исследуемой системы.

В рамках функционального моделирования следует придерживаться принципа объективности при определении исполнителей процессов, т.е. нужно избегать привязки функций к существующей организационной структуре.

# **Каноническое проектирование информационных систем. Содержание стадий и этапов жизненного цикла информационных систем (с учетом ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания).**

Организация канонического проектирования ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС. Стадии и этапы работы описаны в стандарте ГОСТ 34. 601-90.

В зависимости от сложности объекта автоматизации и набора задач, требующих решения при создании конкретной ИС, стадии и этапы работ могут иметь различную трудоемкость. В соответствии с требованиями стандарта допускается объединять последовательные этапы и даже исключать некоторые из них на любой стадии проекта. Допускается также начинать выполнение работ следующей стадии до окончания предыдущей.

Стадии и этапы создания ИС, выполняемые организациями - участниками проекта, фиксируются в договорах и технических заданиях на выполнение работ.

Стадия 1. Формирование требований к ИС:

* + обследование объекта автоматизации и обоснование необходимости создания ИС;
  + выявление требований пользователей к ИС;
  + оформление отчета о выполненной работе и технического задания на разработку проекта.

Стадия 2. Разработка концепции ИС:

* + детальное изучение объекта автоматизации;
  + проведение необходимых научно-исследовательских работ;
  + разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;
  + оформление отчета и утверждение концепции.

Стадия 3. Техническое задание:

* + разработка и утверждение технического задания на создание ИС.

Стадия 4. Эскизный проект:

* разработка предварительных проектных решений по системе и ее частям;
* разработка эскизной документации на ИС и ее части.

Стадия 5. Технический проект:

* разработка проектных решений по системе и ее частям;
* разработка документации на ИС и ее части;
* разработка и оформление документации на поставку комплектующих изделий;
* разработка заданий на проектирование в смежных частях проекта.

Стадия 6. Рабочая документация:

* + разработка рабочей документации на ИС и ее части;
  + разработка и адаптация программ.

Стадия 7. Ввод в действие:

* + подготовка объекта автоматизации;
  + подготовка персонала;
  + комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
  + строительно-монтажные работы;
  + пусконаладочные работы;
  + проведение предварительных испытаний;
  + проведение опытной эксплуатации;
  + проведение приемочных испытаний.

Стадия 8. Сопровождение ИС:

* + выполнение работ в соответствии с гарантийными обязательствами;
  1. послегарантийное обслуживание.

# **Сущность методологии UP (unified process - унифицированного процесса) для разработки программного обеспечения. Фазы жизненного цикла проекта UP.**

**Unified Process** (англ. *унифицированный процесс*) — методология для построения процессов разработки программного обеспечения, позволяющий команде разработки преобразовывать требования заказчика в работоспособный продукт. В зависимости от требований и доступных ресурсов, процесс разработки может быть адаптирован путём включения или исключения определённых проектных активностей. Unified Process активно использует [унифицированный язык моделирования](https://ru.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B8%D1%84%D0%B8%D1%86%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (UML). В ядре UML лежит модель, которая позволяет команде разработки в упрощённом виде понять многообразие сложных процессов, необходимых для разработки программного обеспечения. Различные модели, используемые в Unified Process, позволяют визуализировать систему, описать её структуру и поведение, задокументировать принимаемые в процессе разработки решения. Жизненный цикл проекта UP ориентирован на предоставление заинтересованным лицам и членам проектной группы вех для ознакомления и принятия решений на протяжении всего проекта. Такая возможность позволяет эффективно контролировать ход разработки и своевременно принимать решения о качестве результатов. Жизненный цикл определяется планом проекта, а конечным результатом является окончательное приложение. UP делит жизненный цикл проекта на четыре фазы: начальную, разработки, реализации, передачи.

Изображение выглядит как текст, газета

Автоматически созданное описание

# **Сущность BPMN 2.0 - нотации и модели бизнес-процессов. Разработчик BPMN 2.0. Объединение концепций и объектов в BPMN 2.0.**

BPMN (Business Process Management Notation) – это язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса. Говоря проще, такая нотация представляет собой описание графических элементов, используемых для построения схемы протекания бизнес-процесса. Как минимум, такая схема нужна, чтобы выстроить в соответствии с ней бизнес-процесс и понятно регламентировать его для всех участников. Немаловажным является то, что моделирование BPMN позволяет впоследствии провести автоматизацию бизнес-процессов в соответствии с имеющейся схемой. Разработана [Business Process Management Initiative](https://ru.wikipedia.org/wiki/Object_Management_Group) (BPMI.org) и поддерживается [Object Management Group](https://ru.wikipedia.org/wiki/Object_Management_Group), после слияния обеих организаций в 2005 году. Последняя версия BPMN — 2.0 (2.0.2), предыдущая версия — 1.2. The Business Process Model and Notation — это рабочее название BPMN версии 2.0. Концепция BPMN 2.0 подразумевает создание единой спецификации, описывающей нотацию, метамодель и формат обмена моделей, но с новым именем, которое сохранило бы бренд «BPMN». Финальный вариант нотации был опубликован в январе 2011 года.

Объединение используется для контроля расхождения и схождения последовательного потока. Таким образом, оно будет обозначать ветвление, раздвоение, слияние и соединение маршрутов. Внутренние маркеры будут указывать на тип контроля развития процесса. Изображение ромб. Значки внутри ромба обозначают тип контроля потока. Типы контроля включают в себя:

· Исключающее ИЛИ – исключающее решение и слияние. Может быть как основанным на данных, так и основанным на событиях Основанный на данных может изображаться с маркером X или без него XOR

· ИЛИ – включающее решение и слияние

· Сложный - сложные условия и ситуации

· И – раздвоение и соединение +

Объединения, основанные на событиях: Данное ветвление представляет собой точку ветвления, где альтернативные маршруты основываются на событии, имеющим место в данной точке процесса. Данное событие - обычно это получение сообщения - определяет маршрут. Могут использоваться другие типы событий, например, таймер. Можно выбрать только один из альтернативных маршрутов.

Включающие объединения: Данное ветвление представляет собой точку ветвления, где альтернативные маршруты основываются на условных выражениях, содержащихся в пределах исходящего последовательного потока. В некотором смысле это группировка связанных независимых двойных (Да/Нет) ветвлений. Так как каждый маршрут независим, могут быть задействованы все комбинации маршрутов, от нуля до бесконечности. Однако ветвление должно быть построено так, чтобы был задействован хотя бы один маршрут. С целью проверки наличия хотя бы одного маршрута, может использоваться условие по умолчанию.

# **Сущность методологий проектирования информационных систем (предпосылки возникновения методологий, цель, задачи типовой методологии).**

При разработке ИС применяется множество различных методологий и процессов. Под методологией понимают набор некоторых принципов, методик и практик, применяя которые, можно достигать определенных целей и результатов. Проектирование ИС является одним из основных элементов разработки ИТ-архитектуры предприятия.

На первом этапе основным подходом в проектировании ИС был метод «снизу вверх», когда система создавалась как набор приложений, наиболее важных в данный момент для поддержки деятельности предприятия. Основной целью этих проектов было не создание тиражируемых продуктов, а обслуживание текущих потребностей конкретного учреждения. Следующий этап связан с осознанием того факта, что существует потребность в стандартных программных средствах автоматизации деятельности различных учреждений и предприятий. Системы начали проектироваться «сверху вниз», на основе обобщенного описания бизнес-задач, в предположении, что одна программа должна удовлетворять потребности многих пользователей.

Заложенные «сверху» жесткие рамки не дают возможности гибко адаптировать систему к специфике деятельности конкретного предприятия. Решение этих задач требует серьезных доработок системы. Таким образом, материальные и временные затраты на внедрение системы и ее доводку под требования заказчика обычно значительно превышают запланированные показатели. В то же время заказчики ИС стали выдвигать все больше требований, направленных на обеспечение возможности комплексного использования корпоративных данных в управлении и планировании своей деятельности и согласованности ИТ-решений с быстро меняющимися потребностями бизнеса.

Таким образом, возникла насущная необходимость формирования новой методологии построения ИС. Цель такой методологии заключается в регламентации процесса проектирования ИС и обеспечении управления этим процессом с тем, чтобы гарантировать выполнение требований как к самой ИС, так и к характеристикам процесса разработки.

Основными задачами, решению которых должна способствовать методология проектирования корпоративных ИС, являются следующие:

* обеспечивать создание корпоративных ИС, отвечающих целям и задачам организации, а также предъявляемым требованиям по автоматизации деловых процессов заказчика;
* гарантировать создание системы с заданным качеством в заданные сроки и в рамках установленного бюджета проекта;
* поддерживать удобную дисциплину сопровождения, модификации и наращивания системы;
* обеспечивать преемственность разработки, т.е. использование в разрабатываемой ИС существующей информационной инфраструктуры организации (задела в области ИТ).

Внедрение методологии должно приводить к снижению сложности процесса создания ИС за счет полного и точного описания этого процесса, а также применения современных методов и технологий создания ИС на всем жизненном цикле ИС - от замысла до реализации.

Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла (ЖЦ) ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов.

# **Стадия разработки и утверждения технического задания как основа для создания информационной системы (согласно ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»). Краткая суть предшествующих и последующих стадий.**

Техническое задание (ТЗ) — это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления. При разработке технического задания необходимо решить следующие задачи:

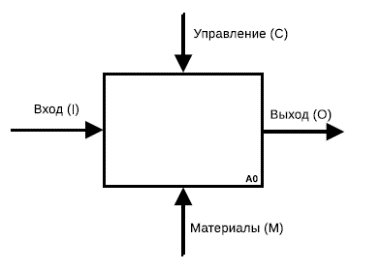
* 1. установить общую цель создания ИС, определить состав подсистем и функциональных задач;
  2. разработать и обосновать требования, предъявляемые к подсистеме-мам;
  3. разработать и обосновать требования, предъявляемые к информационной базе, математическому и программному обеспечению, комплексу технических средств (включая средства связи и передачи данных);
  4. установить общие требования к проектируемой системе;
  5. определить перечень работ по созданию системы и исполнителей;
  6. определить этапы создания системы и сроки их выполнения;
  7. провести предварительный расчет затрат на создание системы и оценить уровень экономической эффективности ее внедрения.

Стадия Техническое задание включает в себя три этапа. На этапе обоснования необходимости разработки программы формулируется постановка задачи, осуществляется сбор необходимых исходных данных, выбираются и обосновываются критерии эффективности и качества программы, обосновывается необходимость проведения этапа научно-исследовательской работы (НИР). На этапе НИР проводится информационное обследование, определяется состав входных и выходных данных задачи, осуществляется предварительный выбор метода решения задачи, устанавливаются возможности применения ранее разработанных программ. На этапе разработки и утверждения ТЗ определяются требования к программе, разрабатывается технико-экономическое обоснование ее разработки, определяются необходимые стадии, этапы и сроки разработки программы и подготовки документации на нее, выбирается язык программирования, определяется необходимость проведения НИР на последующих этапах, согласовывается и утверждается ТЗ.

Техническое задание (ТЗ) — это документ, определяющий задание на проектирование, создание и внедрение ИС, т. е. все последующие стадии ЖЦ. Кроме того, ТЗ определяет и устанавливает основные контуры будущей ИС. Техническое задание играет ключевую роль в формировании профиля проекта ИТ, поскольку регламентирует применение доступных нормативных моделей в создании информационной технологии решения информационной задачи, а также потребность в разработке собственных моделей и их последующей реализации на стадии технического и рабочего проектов.

Предшествующая стадия – обследование. Результаты обследования представляют объективную основу для формирования технического задания на ИС. Обследование — это изучение и диагностический анализ организационной структуры предприятия, его деятельности, организации управления и учета, существующей системы обработки информации. Последующая стадия – эскизный проект. Содержание эскизного проекта задается в ТЗ на систему. Эскизный проект предусматривает разработку предварительных проектных решений по системе и ее частям.

# **Методология функционального моделирования IDEF0 (развитие методологии SADT). Синтаксис и семантика нотации IDEF0. Контекстная диаграмма и декомпозиции. Модели AS-IS и TO-BE.**

IDEF0 (Integration Definition for Function Modeling) – методология функционального моделирования для описания функций предприятия, предлагающая язык функционального моделирования для анализа, разработки, реинжиниринга и интеграции информационных систем бизнес-процессов; или анализа инженерии разработки ПО. IDEF0 основан на SADT и в своей исходной форме включает одновременно: определение языка графического моделирования (синтаксис и семантику) и описание полной методологии разработки моделей.

Нотация IDEF0 содержит только две сущности — блоки и стрелки.

Функциональные блоки (Activity Box) задают действия. Функциональный блок графически изображается в виде прямоугольника. Он задаёт некоторую конкретную функцию в рамках рассматриваемой системы.

1. I (Input), вход — то, что потребляется в ходе выполнения процесса;
2. C (Control), управление — ограничения и инструкции, влияющие на выполнение процесса;
3. O (Output), выход — то, что является результатом выполнения процесса;
4. M (Mechanism), исполняющий механизм — то, что используется для выполнения процесса, но остаётся неизменным.

IDEF0-модели состоят из трех типов документов: графических диаграмм, текста, глоссария. Эти документы имеют перекрестные ссылки друг на друга.

Контекстная диаграмма представляет всю систему как один блок и показывает контекст системы, т. е. связь системы с внешним миром. Модель может иметь только одну контекстную диаграмму. Диаграммы декомпозиции получаются в результате разбиения контекстной диаграммы на отдельные активности. Такой процесс называется функциональной декомпозицией, а диаграммы, получившиеся в результате декомпозиции, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы производится декомпозиция каждой получившейся диаграммы и т. д. Декомпозиция продолжается до достижения нужного уровня подробности описания.

AS-IS – модель «как есть», модель существующего состояния организации. Данная модель позволяет систематизировать протекающие в данный момент процессы, а также используемые информационные объекты. На основе этого выявляются узкие места в организации и взаимодействии бизнес-процессов, определяется необходимость тех или иных изменения в существующей структуре. На этапе построения модели AS-IS важным считается строить максимально приближенную к действительности модель, основанную на реальных потоках процессов, а не на их идеализированном представлении.

Таким образом, учитывая анализ модели «КАК ЕСТЬ», была построена модель «КАК ДОЛЖНО БЫТЬ». Модель «TO-BE» отображает те полезные функции, которые позволят успешно внедрить и использовать данное программное обеспечение для организации процесса дистанционного обучения пользователей.

# **Требования к видам обеспечения информационной системы (согласно ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»). Краткая суть основных видов обеспечения.**

В подразделе «Требования к видам обеспечения АС» приводят требования к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечения АС.

Для математического обеспечения АС приводят требования к составу, области применения (ограничениям) и способам использования в АС математических методов и моделей, типовых алгоритмов и алгоритмов, подлежащих разработке.

Для информационного обеспечения АС приводят следующие требования:

* к составу, структуре и способам организации данных в АС;
* к информационному обмену между компонентами АС и со смежными АС;
* к информационной совместимости со смежными АС;
* по использованию действующих и по разработке новых классификаторов, справочников, форм документов;
* по применению систем управления базами данных;
* к представлению данных в АС;
* к контролю, хранению, обновлению и восстановлению данных.

Для лингвистического обеспечения АС приводят следующие требования:

* к языкам, используемым в АС, и возможности расширения набора языков (при необходимости);
* к способам организации диалога;
* к разработке и использованию словарей, тезаурусов;
* к описанию синтаксиса формализованного языка.

Для программного обеспечения АС приводят следующую информацию:

* требования к составу и видам программного обеспечения;
* требования к выбору используемого программного обеспечения;
* требования к разрабатываемому программному обеспечению;
* перечень допустимых покупных программных средств (при наличии).

Для технического обеспечения АС приводят следующие требования:

* к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию в АС;
* к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения АС.

В требованиях к метрологическому обеспечению АС приводят следующую информацию:

* количественные значения показателей метрологического обеспечения;
* требования к методам (методикам) измерений и измерительного контроля параметров и их характеристик;
* требования к средствам измерений и измерительного контроля;
* требования к метрологическому обеспечению испытаний АС;
* требования к программе метрологического обеспечения АС;
* требования к метрологической совместимости технических средств АС;
* требования проведения метрологической экспертизы технической документации (при необходимости).

Для организационного обеспечения АС приводят следующие требования:

* к структуре и функциям подразделений, участвующих в функционировании АС или обеспечивающих эксплуатацию;
* к организации функционирования АС и порядку взаимодействия персонала и пользователей АС;
* к организации функционирования АС при сбоях, отказах и авариях;
* к порядку обеспечения нормативными документами, необходимыми для разработки АС.

Для методического обеспечения АС приводят следующую информацию:

* перечень применяемых при разработке и функционировании АС нормативно-технических документов (стандартов, нормативов, методик, профилей и т. п.);
* порядок и правила обеспечения разработчиков А С нормативно-технической документацией.

***Техническое обеспечение***- комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы

***Математическое и программное обеспечение*** - совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

***Информационное обеспечение***- совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.

***К лингвистическому обеспечению ИС***относится естественные и искусственные языки, а также средства их лингвистической поддержки: словари лексики естественных языков, тезаурусы (специальные словари основных понятий языка, обозначаемых отдельными словами или словосочетаниями, с определенными семантическими отношениями между ними) предметной области, переводные словари и др.

***Организационное обеспечение*** - совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

# **Сущность типового проектирования информационных систем. Достоинства и недостатки типового проектирования.**

Для сокращения сроков и затрат на выполнение проектов создания ИС, обеспечения соответствия качества ИТ-решений современному научно-техническому уровню применяются процедуры типового проектирования. Объекты автоматизации для использования достоинств типового проектирования должны обладать сходными характеристиками: похожей структурой, целями и процедурами управления, стандартными формами отчетных документов, возможностями использования сходного программно-технического комплекса. Применительно к задачам создания ИС управления предприятием, в качестве основы для типового проектирования используются пакеты прикладных программ, которые настраиваются под особенности деятельности и бизнес-процессов конкретного предприятия.

Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основополагающим требованием для применения методов типового проектирования является возможность декомпозиции, проектируемой ИС на множество составляющих компонентов (подсистем, комплексов задач, программных модулей и т.д.). Для реализации выделенных компонентов выбираются и настраиваются имеющиеся на рынке типовые проектные решения. Типовое проектное решение (ТПР) — это тиражируемое (пригодное к многократному использованию) проектное решение.

В зависимости от уровня декомпозиции системы различают элементный, подсистемный и объектный методы типового проектирования.

Изображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описание

# **Унифицированный язык моделирования UML – происхождение, сущность (двухаспектный синтаксис статической структуры и динамического поведения). Основные диаграммы UML.**

В 1994 году Гради Буч и Джеймс Рамбо, работавшие в компании Rational Software, объединили свои усилия для создания нового языка объектно-ориентированного моделирования. За основу языка ими были взяты методы моделирования Object-Modeling Technique и Booch. Unified Modeling Language (UML) — унифицированный язык моделирования. Расшифруем: modeling подразумевает создание модели, описывающей объект. Unified (универсальный, единый) — подходит для широкого класса проектируемых программных систем, различных областей приложений, типов организаций, уровней компетентности, размеров проектов. UML описывает объект в едином заданном синтаксисе, поэтому, где бы вы не нарисовали диаграмму, ее правила будут понятны для всех, кто знаком с этим графическим языком — даже в другой стране. UML позволяет системным архитекторам представлять свое видение системы в виде набора стандартных диаграмм.

В языке UML есть 12 типов диаграмм:

* 4 типа диаграмм представляют статическую структуру приложения;
* 5 типов представляют поведенческие аспекты системы;
* 3 представляют физические аспекты функционирования системы (диаграммы реализации).

Некоторые из видов диаграмм специфичны для определенной системы и приложения. Самыми доступными из них являются:

* Диаграмма прецедентов (Use-case diagram);
* Диаграмма классов (Class diagram);
* Диаграмма активностей (Activity diagram);
* Диаграмма последовательности (Sequence diagram);
* Диаграмма развёртывания (Deployment diagram);
* Диаграмма сотрудничества (Collaboration diagram);
* Диаграмма объектов (Object diagram);
* Диаграмма состояний (Statechart diagram)

# **Методология информационного моделирования (IDEF1, IDEF1X и E-R - модель). CASE средство для информационного моделирования на примере ERrwin Data Modeler – цель, результат моделирования.**

IDEF1 (Integration Definition for Information Modeling) – данная нотация применяется для построения модели информационных потоков в компании, позволяющая отображать и анализировать их структуру и взаимосвязи. Метод IDEF1 также позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме.

Основные характеристики:

* Помогает выяснить структуру и содержание существующих потоков информации на любом предприятии.
* Помогает определить какие проблемы, выявленные в результате функционального анализа и анализа потребностей, вызваны недостатком управления соответствующей информацией.
* Выявляет, информационные потоки, требующие дополнительного управления для эффективной реализации модели.

Основная цель:

* Исследование движения потоков информации и принципов управления ими на начальном этапе процесса проектирования.
* Определяет информацию и структуру ее потоков, имеющих отношение к деятельности предприятия.
* Выясняет взаимосвязи между существующими информационными потоками в рамках предприятия.

Преимущества:

* Результаты могут быть использованы для стратегического и тактического планирования деятельности предприятия и ее улучшения.
* Обеспечение последовательного и строго структурированного процесса анализа информационных потоков в рамках деятельности предприятия.
* Позволяет эффективно выявлять и корректировать неполноту и неточности существующей структуры информации, на всем протяжении этапа моделирования.

Недостатки:

* Диаграммы может визуально показаться непривлекательной.
* Диаграммы с множеством прямоугольников и стрелок может плохо читаться.

IDEF1X (IDEF1 Extended) – это расширение нотации IDEF1 предназначенное для описания данных (информации). В ее основе лежит язык семантического моделирования, основанного на концепции «сущность - связь», позволяющей определять данные и связи между ними. Основными элементами модели IDEF1X являются сущности, атрибуты и отношения.

Основная характеристика: теоретической базой построения информационной модели является теория баз данных типа «сущность-связь». Основные элементы: сущность, атрибут, отношение.

Для чего используется:

* Используется для создания информационной модели предметной области с помощью идентификации ее сущностей и связей между ними.
* Применяется для описания данных в целях последующей автоматизации их обработки с помощью систем управления базами данных.

Преимущества:

* Простота изучения и возможность автоматизации.
* Используются рядом распространенных CASE-средств (в частности, ERwin, Design/IDEF).
* Жесткая и строгая стандартизация моделирования.

Недостатки:

* Разработаны специально для построения реляционных информационных систем, и неприменимы для проектирования, например, объектно-ориентированных систем.
* Невозможность адекватно и полно описать предметную область.

ER-модель (от англ. entity-relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. entity-relationship diagram, ERD, ER-диаграмма).

CASE-средство ERwin предназначено для разработчиков, проектировщиков БД, системных аналитиков для построения модели данных в процессе разработки технического проекта информационной системы. ERWin имеет два уровня представления модели - логический и физический. Логический уровень — это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире. Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует, физическая модель зависит от конкретной реализации СУБД. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т. д. Для переключения между логической и физической моделью данных служит список выбора в левой части панели инструментов ERWin.

Для создания моделей данных в ERWin можно использовать две нотации: IDEF1X и IE (Information Engineering). Переключение между нотациями можно сделать в закладке Methodology диалога Preferences (меню Option/Preferences).

ERWin имеет несколько уровней отображения диаграммы: уровень сущностей, уровень атрибутов, уровень определений, уровень первичных ключей и уровень иконок. Переключиться на другие уровни отображения можно при помощи контекстного меню, которое появляется в любом месте диаграммы, не занятом объектами модели (пункт Display Level). ERWin позволяет связать с сущностью большую и малую иконки. При переключении на уровень иконок показывается большая иконка.

# **Процесс (frameworks) как формальное описание бизнес-процессов разработки информационной системы. Отличие процессного подхода от методологий функционального моделирования. Основные элементы процессного подхода. Выделение и классификация процессов.**

Процесс – это некая последовательность действий, которую необходимо выполнить, чтобы получить определенный результат. Необходимо отметить, что в модели бизнеса как процесса результат может и не быть явным в отличии от функциональной модели. Принципиальное отличие процессного моделирования от функционального заключается в том, что при процессном моделировании основное внимание уделяется не тому, что мы хотим получить, а тому, что нужно сделать для получения результата, т. е. не итогам той или иной деятельности, а самой последовательности действий.

# Основные элементы процессного подхода:

Под бизнес-процессом понимают совокупность различных видов деятельности, которые создают результат, имеющий ценность для потребителя. Бизнес-процесс – это цепочка работ (функций), результатом которой является какой-либо продукт или услуга. Каждый бизнес-процесс имеет свои границы и роли. В процессном подходе используются следующие ключевые роли:

Владелец процесса – человек, отвечающий за ход и результаты процесса в целом. Лидер команды — работник, обладающий знаниями о бизнес-процессе и имеющий позитивные личные качества. Коммуникатор – работник, обучающий команду различным методам работы, подготавливающий совместно с лидером совещания и анализирующий их результат. Координатор процесса – работник, отвечающий за согласованную работу всех частей бизнеса и обеспечивающий связь с другими бизнес-процессами. Участники команды – специалисты различных уровней иерархии.

Одним из основных элементов процессного подхода является команда. Существует несколько типов процессных команд:

Ситуационная команда – обычно работает на постоянной основе и выполняет периодически повторяющуюся работу.

Виртуальная команда – создается для разработки нового продукта или услуги.

Ситуационный менеджер – высококвалифицированный специалист, способный самостоятельно выполнить до 90% объема работ.

Важной задачей процессного подхода является формирование процессных команд. Достижение определенной совокупности целей за счет выполнения бизнес-процессов называется деревом целей. Дерево целей имеет, как правило, иерархический вид. Каждая цель имеет свой вес и критерий (количественный или качественный) достижимости. Бизнес-процессы реализуют бизнес-функции предприятия. Под бизнес-функцией понимают вид деятельности предприятия. Бизнес-функции связаны с показателями деятельности предприятия, образующими дерево показателей. На основании показателей строится система показателей оценки эффективности выполнения процессов.

## Выделение и классификация процессов:

При процессном описании должны решаться, как минимум, две задачи:

1.Идентификация всей системы «функциональных областей» и процессов компании и их взаимосвязей.

2.Выделение «ключевых» интегрированных процессов и их описание на потоковом уровне.

Каждая деятельность компании реализуется как процесс, который имеет своего потребителя: внешнего — клиента или внутреннего — сотрудников или подразделения компании, реализующих другие процессы. На стадии системного описания процессов и выявляется значимость каждого процесса — в том числе происходит очищение от малопонятной деятельности. На этом этапе выбираются ключевые процессы для потокового описания, которое необходимо, например, для создания информационной системы предприятия.

Наиболее распространены следующие четыре вида бизнес-процессов:

1.Процессы, создающие наибольшую добавленную стоимость (экономическую стоимость, которая определяется издержками компании, относимыми на продукцию).

2.Процессы, создающие наибольшую ценность для клиентов (маркетинговую стоимость за счет дифференциации продукции).

3.Процессы с наиболее интенсивным межзвенным взаимодействием, создающие транзакционные издержки.

4.Процессы, определенные стандартами ИСО 9000, как обязательные к описанию при постановке системы менеджмента качества.

Целесообразно основываться на следующих классах процессов: 1.основные; 2.процессы управления; 3.процессы обеспечения; 4.сопутствующие; 5.вспомогательные; 6.процессы развития.

# **Язык графического моделирования UML: объектно-ориентированный подход в реализации нотации. Классификаторы (классы), экземпляры классов. Классификаторы в диаграммах прецедентов, классов, последовательностей.**

UML определяет два типа областей действия для членов: *экземпляр* и *классификатор*.

* **Члены классификаторы**. Во многих языках называются static. Область действия — сам класс.
  + Значения полей одинаковы для всех экземпляров в данной единице трансляции
  + Вызов метода не меняет состояние объекта
* **Члены экземпляры**. Область действия — объект.
  + Значения полей могут отличаться в разных объектах
  + Методы могут изменять поля

Чтобы показать принадлежность к классификатору, имя подчёркивается, в противном случае область действия полагается областью действия по умолчанию.

**classifier (классификатор)** - элемент модели, описывающий определенные черты структуры и поведения системы. К классификаторам относятся классы, актанты, композиты, типы данных, интерфейсы, узлы, сигналы, подсистемы и варианты использования.

Объект является отдельным **экземпляром** класса, который создается на этапе выполнения программы. Он может иметь свое собственное имя и конкретные значения атрибутов.

Сообщения, которые посылают объекты друг другу на диаграммах последовательности, как правило, вызывают определенную операцию класса этого объекта. Чтобы соотнести сообщение с операцией, нужно, чтобы каждому объекту был назначен классификатор, то есть класс, объектом которого он является.

# **Бизнес-процессное моделирование в нотации BPMN 2.0: процессные потоковые модели. Синтаксис и семантика нотации BPMN 2.0. Оркестровка и хореография в бизнес-процессных моделях.**

BPMN ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Business Process Model and Notation, нотация и модель бизнес-процессов) — система условных обозначений и их описания в XML для [моделирования бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2). Процессные потоковые модели— это модели, описывающие процесс последовательного во времени преобразования материальных и информационных потоков компании в ходе реализации какой-либо бизнес-функции или функции управления.

На верхнем уровне описывается логика взаимодействия участников процесса, на нижнем — технология работы отдельных специалистов на своих рабочих местах. Процессные потоковые модели отвечают на вопросы кто—что—как—кому. Основной принцип процессного подхода определяет структурирование бизнес–системы в соответствии с деятельностью и бизнес-процессами предприятия, а не в соответствии с его организационно-штатной структурой. Именно бизнес-процессы, обеспечивающие значимый для потребителя результат, представляют ценность и для специалистов, проектирующих ИС.

BPMN-процесс – это любой бизнес-процесс, отражённый с помощью нотации. Процессы состоят из элементов, каждый из которых обозначается на схеме специальным значком. Элементы нотации BPMN – это элементы графической схемы, но также и элементы самого бизнес-процесса. В BPMN 2.0 элементы представлены в виде специальных значков.

Событие (Event) – это то, что происходит в течение бизнес-процесса или его Xореографии. Событие оказывает влияние на ход бизнес-процесса и чаще всего имеет причину (триггер) или воздействие (результат). Изображается в виде круга со свободным центром, предназначенным для дифференцировки внутренними маркерами различных триггеров или их результатов. Согласно влиянию Событий на ход бизнес-процесса, выделяют три типа: Стартовое событие (Start), Промежуточное событие (Intermediate) и Конечное событие (End).

Действие (Activity) – общий термин, обозначающий работу, выполняемую исполнителем в ходе бизнес-процесса. Действия могут быть либо элементарными, либо неэлементарными (составными). Выделяют следующие виды действий: Подпроцесс (Sub-Process) и Задача (Task). И Задача, и Подпроцесс изображаются в виде прямоугольников с закругленными углами. Все Действия могут являться элементами как стандартных Процессов, так и Хореографий.

Шлюзы используются для контроля расхождений и схождений Потока операций как в Процессе, так и в Хореографии. Таким образом, данный термин подразумевает ветвление, раздвоение, слияние и соединение маршрутов. Внутренние маркеры указывают тип контроля развития бизнес-процесса.

Поток операций (Sequence Flow) служит для отображения того порядка, в котором организованы действия Процесса или условия Хореографии.

Поток сообщений (Message Flow) служит для отображения обмена сообщениями между двумя участниками, готовыми эти сообщения отсылать и принимать. На диаграмме взаимодействия BPMN два отдельно взятых Пула представляют собой двух участников Процесса (бизнес-сущности или бизнес-роли).

Ассоциация (Association) служит для установления связи между информацией или Артефактами и элементами потока. Текстовые объекты, а также графические объекты, не относящиеся к элементам потока, могут соотноситься с элементами потока. При необходимости Ассоциация может указывать направление потока.

Пул (Pool) представляет собой Участника Взаимодействия. Пул также может выступать в качестве Зоны ответственности или графического контейнера, отвечающего за разделение определенного набора действий, относящихся к другим Пулам, что обычно встречается в ситуациях типа «бизнес для бизнеса» (B2B). Внутри Пула МОЖЕТ находиться дополнительная информация по выполняемому Процессу. В случае, если такой информации в Пуле не содержится, то он МОЖЕТ представлять собой «черный ящик».

Дорожка (Lane) используется для отображения распределения ролей и может быть как вертикальной, так и горизонтальной. Служит для упорядочивания и категоризации Действий.

Входящий поток операций может быть присоединен к любой точкой Элемента потока (слева, справа, сверху, снизу). Подобно ему, Исходящий поток операций может брать начало из любой точки Элемента потока (слева, справа, сверху, снизу). Поток сообщений обладает теми же свойствами, что и Поток операций.

Оркестровка BPMN – это последовательность выполнения действий в рамках одного управляющего центра (пула). Оркестровка подразумевает, что процесс завершится только после выполнения всех его потоков управления. Поток управления BPMN представляет собой траекторию, по которой в данный момент времени исполняется отдельная операция процесса. Чтобы правильно показать передачу информации между взаимодействующими участниками процесса требуется более подробное **моделирование диаграммы BPMN**. Для этого каждая задача назначается субъекту отдельного пула и взаимодействие между ними происходит с помощью потоков сообщений, как показано на рисунке ниже.

Диаграмма «Хореография» описывает последовательность взаимодействий участников при выполнении **бизнес-процессов BPMN**. Такие диаграммы строят для того, чтобы еще более наглядно показать взаимодействия между **бизнес-процессами BPMN** и их участниками. Задачи и подпроцессы хореографии BPMN предназначены для обозначения взаимодействий между участниками бизнес-процессов (или между бизнес-процессами). Они изображаются в виде графических объектов, состоящих из нескольких элементов. Маркеры задач и подпроцессов хореографии BPMN служат для обозначения циклов и множественных экземпляров задач и подпроцессов. Шлюзы BPMN используются на диаграммах хореографии с той же целью, что и на диаграммах бизнес-процессов – для создания параллельных или альтернативных потоков последовательности. Однако, это использование имеет свои особенности.

# **Бизнес-процессное моделирование в нотации BPMN 2.0: синтаксическая, структурная и поведенческая (семантическая) устойчивости. Поведенческие аномалии. Имитация (симуляция) бизнес-процессов.**

BPMN ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Business Process Model and Notation, нотация и модель бизнес-процессов) — система условных обозначений и их описания в XML для [моделирования бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2). Ключевым элементом BPMN является выбор форм и иконок, соответствующих описанным в данной спецификации элементам. Намерением разработчиков спецификации явилось создание стандартного визуального языка, легко распознаваемого и понятного всем разработчикам моделей процессов. Создание и отображение Диаграмм Процессов BPMN подразумевает ОБЯЗАТЕЛЬНОЕ использование графических элементов, форм и маркеров, описанных в данном документе.

При моделировании диаграммы BPMN допускаются следующие расширения:

* Для определенных графических элементов МОГУТ БЫТЬ добавлены новые маркеры (индикаторы). Такие маркеры или индикаторы могут использоваться для выделения конкретного атрибута элемента BPMN либо для отображения нового подтипа соответствующего понятия.
* На Диаграмму в качестве вариации Артефакта МОЖЕТ БЫТЬ добавлена новая форма, однако, эта новая форма НЕ ДОЛЖНА противоречить форме, определенной для какого-либо другого элемента BPMN или маркера.
* Графические элементы МОГУТ БЫТЬ выделены цветом. Использование какого-то цвета МОЖЕТ соответствовать определенной семантике, используемой для пополнения информации, которую, согласно данному документу, передают графические элементы.
* Стили линий, используемые для отображения графических элементов, МОГУТ БЫТЬ изменены, однако, они НЕ ДОЛЖНЫ противоречить каким-либо другим стилям линий, УКАЗАННЫМ в данной спецификации.
* Расширение НЕ ДОЛЖНО изменять установленной формы любого графического элемента или маркера (например, квадрат не может быть заменен треугольником, закругленные края не могут стать острыми и т.д.)

Создание и отображение диаграмм BPMN ДОЛЖНО БЫТЬ осуществлено в соответствии с техническими требованиям и ограничениям, при этом следует обращать внимание на соединения и другие взаимодействия между графическими элементами на схеме. Там, где разрешенные или требуемые соединения являются условными и основываются на атрибутах соответствующего элемента, ДОЛЖНО БЫТЬ гарантировано сообщение между соединениями и значениями этих атрибутов.

В данной спецификации приводится несколько понятий семантики, используемых для определения Процесса; при этом они ассоциируются с графическими элементами, маркерами и соединениями. Поскольку при применении диаграмма BPMN интерпретируется как семантическая спецификация данного Процесса, такая интерпретация ДОЛЖНА должна согласовываться с описанной в данном документе семантической интерпретацией.

Аномалии:

### Всё завершения в одно завершающее событие (Стилевая ошибка BPMN)

В случае с BPMN одно завершающее событие мешает собрать правильную статистику о самых частых окончаниях процессов. Отсутствие правильной статистики обходится дороже, чем сэкономленные байты.

### **Одна развилка для сборки и разведения токенов (стилевая ошибка BPMN)**

Нотация настоятельно рекомендует не использовать одну развилку и для сведения и для разведения потоков управления:

### **Сверху вниз (стилевая ошибка BPMN)**

В BPMN схемы моделируются слева направо. Некоторые редакторы не дают размещать пулы или дорожки вертикально.

Не сходятся токены (формальная ошибка BPMN )

Это одна из неявных особенностей BPMN: по схемам движутся токены и их количество меняется в зависимости от пройденных элементов. Нужно уметь в голове проигрывать такие путешествия токенов.

### Элементы ничем не заканчиваются (стилистическая ошибка BPMN )

BPMN формально разрешает брошенные элементы: Старайтесь чтобы из задачи всегда был выход.

Имитация:

С помощью инструмента моделирования BPMN вы можете визуализировать выполнение процессов и анализировать процессы на основе пошаговых, настраиваемых сценариев с одним или несколькими случаями. Это предоставляет информацию о стоимости, времени цикла, ресурсах и узких местах в ваших процессах.

Пользователи моделирования, у которых есть доступ только для чтения к диаграмме, могут использовать инструмент моделирования BPMN, но не могут управлять сценариями моделирования диаграмм.

Инструмент моделирования работает только с диаграммами BPMN 2.0.

Вы можете получить доступ к симулятору BPMN двумя способами:

* Откройте проводник, выберите модель и нажмите «**Правка» > «Симуляция диаграммы BPMN»** .
* Когда вы открыли диаграмму в редакторе, QuickModel или инструменте сравнения диаграмм, разверните пользовательское меню и щелкните **Моделирование**.

## Функции инструмента моделирования BPMN

Инструмент моделирования имеет следующие функции:

* [Пошаговое моделирование](https://documentation.signavio.com/suite/en-us/Content/process-manager/userguide/step-by-step-simulation.htm) : шаг за шагом проходите процесс по элементам и полностью сосредотачиваетесь на потоке процесса.
* [Моделирование одного случая](https://documentation.signavio.com/suite/en-us/Content/process-manager/userguide/one-case-simulation.htm) : Моделируйте один конкретный случай, чтобы проанализировать затраты и время, затрачиваемое на случай.
* [Моделирование нескольких случаев](https://documentation.signavio.com/suite/en-us/Content/process-manager/userguide/multiple-case-simulation.htm) : Моделирование нескольких запусков процессов с учетом настроенных вероятностей и анализа количественных данных и узких мест.

# **Объектно-ориентированный подход к проектированию информационных систем. Сущность методологии RUP (rational unified process – рационального унифицированного процесса) для разработки программного обеспечения. Стадии (фазы) и рабочие процессы RUP.**

**Rational Unified Process (RUP)** — это процесс разработки программного обеспечения. Его цель состоит в том, чтобы гарантировать высокое качество программного продукта, отвечающего потребностям конечных пользователей, в пределах предсказуемого графика и бюджета выполнения.

Структура жизненного цикла проекта, выполняемого по технологии RUP, графически обычно представляется на координатной плоскости. При этом горизонтальная ось отображает стадии жизненного цикла и характеризует время, а вертикальная — процессы, отражающие виды деятельности, которые, как правило, выполняются в ходе любого проекта.

**1. Стадия Начало ( Inception ).**

В этой стадии, как и в случае UP:

— формируется общее видение проекта и определяются его границы;

— формируется экономическое обоснование проекта;

— определяются основные требования, ограничения и функциональность продукта;

— создается исходная версия модели прецедентов;

— выполняется оценка рисков.

Контрольная веха целей начальной фазы жизненного цикла предполагает согласие заказчика и разработчика в продолжение проекта. Итогом стадии являются ответы на вопросы — достигнуто ли соглашение о задачах проекта и его масштабе; будет ли работа над проектом продолжена.

**2. Стадия Уточнение ( Elaboration ).**

В этой стадии производится анализ предметной области и построение исполняемой архитектуры. Она включает:

— уточнение экономического обоснования с целью получения более точных оценок сроков и стоимости;

— документирование требований, включая детальное описание для основных прецедентов; — проектирование, реализацию и тестирование исполняемой архитектуры;

— мероприятия по снижению основных рисков.

Результатом выполнения фазы являются ответы на вопросы — согласована ли архитектура исполнения, которая будет использоваться для разработки приложения; являются ли созданная на данный момент потребительская ценность и риск приемлемыми.

**3. Стадия Конструирование ( Construction ).**

В данной стадии выполняются мероприятия по созданию продукта, в котором реализована боّльшая часть требуемой функциональности. Стадия завершается вехой функциональной готовности и внешним релизом системы. Результатом стадии являются ответы на вопросы — получили ли мы достаточно близкое к завершению приложение; можно ли переключить внимание коллектива на настройку, окончательную доработку и обеспечение гарантии успешного развертывания.

**4. Стадия Внедрение.**

Эта стадия включает в себя разработку программы бета-тестирования, собственно бета-тестирование, обучение пользователей, а также всестороннюю оценку качества разработанного продукта. Если качество разработанного продукта не соответствует ожиданиям и требованиям пользователей, определенным в стадии Начало, то стадия Внедрение повторяется вновь. Полное выполнение всех целей и получение продукта требуемого качества означает, что достигнута веха готового продукта ( Product Release ) и, следовательно, полный цикл разработки за вершен.

**Рабочие процессы RUP:**

1. Моделирование бизнес-процессов 2. Управление требованиями 3. Анализ и проектирование 4. Реализация 5. Тестирование 6. Развертывание 7. Управление конфигурациями и изменениями 8. Управление проектом 9. Управление средой

# **Сущность методологии RAD для разработки программного обеспечения. Этапы моделирования в технологии RAD.**

**RAD** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *rapid application development* — быстрая разработка приложений) — концепция организации технологического процесса разработки [программных продуктов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82), ориентированная на максимально быстрое получение результата в условиях сильных ограничений по срокам и бюджету и нечётко определённых требований к продукту. Эффект ускорения разработки достигается путём использования соответствующих технических средств и непрерывного, параллельного с ходом разработки, уточнения требований и оценки текущих результатов с привлечением заказчика.

Особенностью этой методологии является реализация следующих подходов:

— короткий по времени, но тщательно проработанный график выполнения работ;

— небольшая команда программистов;

— цикличность разработки: на каждом цикле разработчики по мере того, как приложение приобретает функциональность, запрашивают заказчика и реализуют в продукте новые требования.

Модель ИС, создаваемая по технологии RAD, позволяет заказчикам после каждой итерации в процессе проектирования видеть текущие возможности ИС и представлять, какие необходимы дополнительные вложения, трудозатраты и сроки, чтобы повысить функциональность системы. **Этапы:**

1.Бизнес-моделирования

2. Моделирование данных

3. Моделирование обработки

4. Генерация приложений

5. Тестирование и объединение.

# **Бизнес-процессное моделирование в нотации BPMN 2.0: подпроцессы, граничные и конечные события в подпроцессах. Модификаторы работ (действий, активностей).**

Система условных обозначений и их описания в XML для [моделирования бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2). Подпроцесс представляет собой Действие, заключающее в себе другие Действия, Шлюзы, События и Потоки операций. Графически Подпроцесс изображается в качестве элемента Потока операций Процесса в [спецификации BPMN 2.0](https://www.elma-bpm.ru/bpmn2/). Подпроцесс также может изображаться «открытым» в случае, если необходимо показать другой Процесс внутри данного Подпроцесса. Подпроцесс определяет контекстные рамки, необходимые для обеспечения видимости атрибутов, рамки транзакции, необходимые для управления исключениями, Событиями или компенсацией.

Встроенный Подпроцесс (Embedded Sub-Process (Sub-Process)). Подпроцесс, как и Задача, изображается в виде прямоугольника с закругленным углами. Подпроцесс представляет собой прямоугольник с закругленным углами, который ДОЛЖЕН БЫТЬ выполнен одинарной тонкой линией.

Одинарная жирная линия в изображении границ Подпроцесса ДОЛЖНА означать использование данного графического элемента в качестве Действия Вызов (Подпроцесс).

Пунктирная линия в изображении границ Подпроцесса ДОЛЖНА означать использование данного графического элемента в качестве Событийного Подпроцесса.

Двойная линия в изображении границ Подпроцесса ДОЛЖНА означать использование данного графического элемента в качестве Подпроцесса Транзакции.

Подпроцесс может быть свернутым (Collapsed Sub-Process), при этом его детали скрыты. Подпроцесс также может быть развернутым (Expanded Sub-Process), при этом его детали отображаются внутри Процесса, в котором данный Подпроцесс содержится. В случае, если Подпроцесс является свернутым, то используется маркер, позволяющий отличить Подпроцесс от Задачи.

Событийный Подпроцесс (Event Sub-Process). Событийным Подпроцессом называется специфический Подпроцесс, используемый внутри Процесса (Подпроцесса). Такой Подпроцесс не является частью Стандартного потока операций, включенного в родительский Процесс, и не имеет входящих или исходящих Потоков операций.

Транзакция (Transaction). Транзакцией называется специфический тип Подпроцесса, который демонстрирует определенное поведение, контролируемое посредством протокола транзакции (например, WS-Transaction). Граница графического элемента Транзакция выполнена двойной линией. Транзакция представляет собой прямоугольник с закругленным углами, который ДОЛЖЕН БЫТЬ выполнен двойной тонкой линией.

Действие – общий термин, обозначающий работу, выполняемую исполнителем в ходе бизнес-процесса. Действия могут быть либо элементарными, либо неэлементарными (составными). Выделяют следующие виды действий, являющихся частью модели Процесса: Подпроцесс (Sub-Process) и Задача (Task). И Задача, и Подпроцесс изображаются в виде прямоугольников с закругленными углами. Все Действия могут являться элементами как стандартных Процессов, так и Хореографий.

Стандартный поток операций (Normal Flow) относится к потокам, берущим начало от Стартового события и следующим по ходу выполнения Действий.

Дорожка (Lane) используется для отображения распределения ролей и может быть как вертикальной, так и горизонтальной. Служит для упорядочивания и категоризации Действий.

# **Гибкие методологии разработки информационных систем (программ) – Agile software development. Методологии Scrum, DSDM, eXtreme programming.**

В настоящее время наибольшее распространение в практике все больше находят так называемые гибкие методологии разработки (англ. **Agile** software development, далее — agile -методы). **Гибкие методологии** — это семейство подходов к разработке программного обеспечения, которые ориентированы: на применение итеративной разработки; динамическое формирование требований; обеспечение реализации требований. Основная цель большинства гибких методологий — обеспечить минимизацию рисков. Это достигается организацией разработки в виде серии коротких циклов — итераций. Задача каждой итерации — представить пользователю версию программного продукта. Поэтому каждая итерация в результате представляет собой программный проект. По итогам каждой реализованной итерации проектная команда выполняет анализ и переоценку приоритетов разработки.

**Scrum** («скрам», от англ. — толкучка) — это методология управления проектами, которая применяется при проектировании ИС и позволяет реализовать гибкую технологию разработки ПО. Scrum представляет собой набор подходов, на основе использования которых реализуется процесс разработки, позволяющий в фиксированные и достаточно короткие по времени итерации — спринты (sprints) предоставлять пользователю работающее ПО с реализованными новыми возможностями. Характеристики возможностей ПО, реализуемые в очередном спринте, устанавливаются в начале спринта на этапе планирования. Эти предполагаемые возможности не могут изменяться на всем протяжении реализации итерации. Гибкость процесса разработки становится возможной благодаря строго фиксированной относительно короткой длительности спринта. Scrum может применяться не только для управления проектами по разработке ПО, но также в работе команд, обеспечивающих поддержку ПО, или как подход управления разработкой и сопровождением программ. Основные используемые инструменты: burndown (диаграмма), backlog (истории пользователей). Основное назначение диаграммы — отображать фактическую оставшуюся трудоемкость по задачам итерации и сопоставлять ее с идеальной оставшейся трудоемкостью.

**DSDM** ( Dynamic Systems Development Method, динамический метод разработки систем) представляет собой методику разработки ПО, основанную на концепции быстрой разработки приложений (RAD). DSDM стал одним из признанных подходов к проектированию и созданию приложений. DSDM реализует итеративный и инкрементный подход, который предполагает активное участие в процессе проектирования и разработки конечного пользователя.

**eXtreme Programming** (экстремальное программирование) — методология гибкой разработки, ориентированная на использование таких практик как TDD ( Test Driven Development, разработка через тестирование), парное программирование, непрерывная интеграция, рефакторинг, коллективное владение кодом, описание требуемой функциональности при помощи историй пользователей и др. Эта методология использует практику разработки программ, при которой вначале пишется тестовый код, а затем его реализация, при этом постоянно выполняется проверка работоспособности программного кода, с использованием написанных тестов. Эта практика позволяет создавать автоматически тестируемый код, безболезненно проводить рефакторинг и свободнее менять архитектуру или реализацию разрабатываемого приложения.

# **Сущность процессного подхода в разработке информационных систем. Процессный подход в ГОСТ Р ИСО 9000-2015 «Системы менеджмента качества. Основные положения и словарь» и ГОСТ Р ИСО 9001-2015 «Системы менеджмента качества. Требования».**

Процессный подход предполагает, что вся деятельность (или бОльшая ее часть) по обеспечению ИБ оформляется в логические блоки, каждый из которых имеет четкие цели реализации, ресурсы, входные и выходные данные (могут использоваться другими процессами), наборы утвержденных для выполнения процедур, метрики эффективности и др. атрибуты, характерные для любого бизнес-процесса.

9000-2015: Процессный подход

2.3.4.1 Формулировка: Последовательные и прогнозируемые результаты достигаются более эффективно и результативно, когда деятельность осознается и управляется как взаимосвязанные процессы, которые функционируют как согласованная система.

2.3.4.2 Обоснование: Система менеджмента качества состоит из взаимосвязанных процессов. Понимание того, каким образом этой системой создаются результаты, позволяет организации оптимизировать систему и ее результаты деятельности.

2.3.4.3 Основные преимущества:

- повышение способности сосредотачивать усилия на ключевых процессах и возможностях для улучшения;

- последовательные и прогнозируемые выходы в системе согласованных процессов;

- оптимизация деятельности посредством результативного менеджмента процессов, эффективного использования ресурсов и снижения межфункциональных барьеров;

- возможности для организации обеспечивать уверенность заинтересованных сторон в отношении согласованности, результативности и эффективности ее деятельности.

2.3.4.4 Возможные действия:

- определение целей системы и процессов, необходимых для их достижения;

- установление полномочий, ответственности и подотчетности для осуществления менеджмента процессов;

- осмысление возможностей организации и определение ограничений по ресурсам до начала осуществления действий;

- определение взаимозависимости процессов и анализ влияния изменений отдельного процесса на систему в целом;

- осуществление менеджмента процессов и их взаимосвязей как системы для результативного и эффективного достижения целей организации в области качества;

- обеспечение доступности информации, необходимой для функционирования и улучшения процессов, а также для мониторинга, анализа и проведения оценки результатов деятельности системы в целом;

- осуществление менеджмента рисков, которые могут оказать влияние на выходы процессов и общие выходы системы менеджмента качества.

9001-2015: Процессный подход  
Общие положения:

* Настоящий стандарт направлен на применение "процессного подхода" при разработке, внедрении и улучшении результативности системы менеджмента качества в целях повышения удовлетворенности потребителей путем выполнения их требований.
* Понимание и менеджмент взаимосвязанных процессов как системы способствует результативности и эффективности организации в достижении намеченных результатов. Этот подход позволяет организации управлять взаимосвязями и взаимозависимостями между процессами системы, так что общие результаты деятельности организации могут быть улучшены.
* Процессный подход включает в себя систематическое определение и менеджмент процессов и их взаимодействия таким образом, чтобы достигать намеченных результатов в соответствии с политикой в области качества и стратегическим направлением организации.
* Применение процессного подхода в системе менеджмента качества позволяет:  
  a) понимать и постоянно выполнять требования;  
  b) рассматривать процессы с точки зрения добавления ими ценности;  
  c) достигать результативного функционирования процессов;  
  d) улучшать процессы на основе оценивания данных и информации.

# **Язык графического моделирования UML: сущности и отношения для установления семантических связей. Типы отношений в диаграммах прецедентов и классов.**

Основными объектно-ориентированными блоками являются сущности. В языке UML имеется четыре вида сущностей: структурные, поведенческие, группирующие, аннотационные.

**Структурные сущности** – это имена существительные в моделях на языке UML. Они представляют собой статические части модели, соответствующие концептуальным или физическим элементам системы. Существует пять разновидностей концептуальных и логических сущностей:

* Класс (Class)
* Интерфейс (Interface)
* Кооперация (Collaboration)
* Прецедент (Use case)
* Активный класс (Active class)

# Изображение выглядит как стол Автоматически созданное описание

**Поведенческие сущности** (Behavioral things) являются динамическими составляющими модели UML. Это глаголы языка: они описывают поведение модели. Существует всего два типа поведенческих сущностей.

* Взаимодействие (Interaction)Изображение выглядит как текст

  Автоматически созданное описание
* Автомат (State machine)Изображение выглядит как текст

  Автоматически созданное описание

**Группирующие сущности** являются организующими частями модели, это блоки, на которые можно разложить модель. Основной группирующей сущностью является Пакет (Package). Существуют также вариации пакетов, например каркасы (Frameworks), модели и подсистемы.

Изображение выглядит как стол

Автоматически созданное описание**Аннотационные сущности** – пояснительные части модели UML. Это комментарии для дополнительного описания, разъяснения или замечания к любому элементу модели. Имеется только один базовый тип аннотационных элементов – примечание. Примечание (Note) – это символ для изображения комментариев, присоединенных к элементу или группе элементов

В языке UML определены четыре типа отношений: зависимость, ассоциация, обобщение, реализация. Эти отношения являются основными связующими строительными блоками в UML.

Изображение выглядит как антенна

Автоматически созданное описание**Зависимость (Dependency)** – это семантическое отношение между двумя сущностями, при котором изменение одной из них, независимой, может повлиять на семантику другой, зависимой.

**Ассоциация (Association)** – отношение, описывающее совокупность связей между объектами. Разновидностью ассоциации является агрегирование (Aggregation) – структурное отношение между целым и его частями. Графическое изображение ассоциации может включать кратность и имена ролей.

**Обобщение (Generalization)** – это отношение “специализация/обобщение”, при котором объект специализированного элемента (потомок) может быть подставлен вместо объекта обобщенного элемента (родителя или предка). Таким образом, потомок (Child) наследует структуру и поведение своего родителя (Parent).

**Реализация (Realization)** – это отношение между классификаторами, при котором один классификатор определяет “контракт”, а другой гарантирует его выполнение. Отношения реализации встречаются в двух случаях: во-первых, между интерфейсами и реализующими их классами или компонентами, а во-вторых, между прецедентами и реализующими их кооперациями.