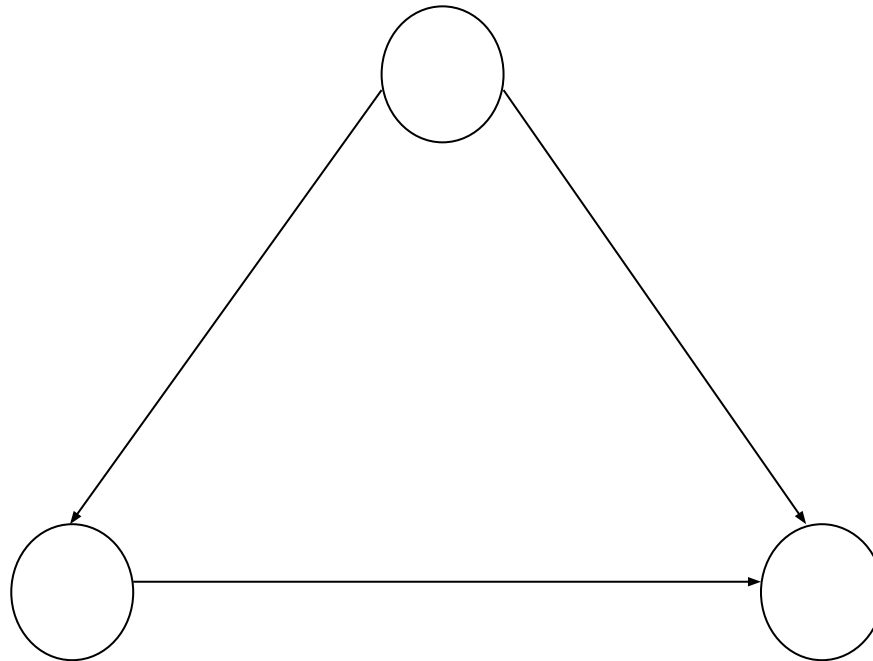


Каноническое проектирование ИС

Технология проектирования

Методологий
(концепция+метод)



Инструментальные средства проектирования

Организация проектирования

Технология проектирования ИС -
это совокупность методологии и
средств проектирования ЭИС , а
также методов и средств организации
проектирования

Цель методологии проектирования ИС 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.

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.

- **Под проектом ИС** следует понимать проектно-конструкторскую и технологическую документацию, в которой представлено описание проектных решений по созданию и эксплуатации ИС в конкретной программно-технической среде.

Под проектированием ИС следует понимать процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования объектов аналогичного назначения в соответствии с ГОСТом в проекте ИС. Следовательно, проектирование ИС сводится к последовательной формализации проектных решений на различных стадиях жизненного цикла ИС: планирования и анализа требований, технического и рабочего проектирования, внедрения и эксплуатации ИС.

Основные требования, предъявляемые к выбираемой технологии проектирования:

- созданный проект должен отвечать требованиям заказчика;
- выбранная технология должна максимально отражать все этапы цикла жизни проекта;
- выбираемая технология должна обеспечивать минимальные трудовые и стоимостные затраты на проектирование и сопровождение проекта;
- технология должна быть основой связи между проектированием и сопровождением проекта;
- технология должна способствовать росту производительности труда проектировщика;
- технология должна обеспечивать надежность процесса проектирования и эксплуатации проекта;
- технология должна способствовать простому ведению проектной документации.

Основная задача любого успешного проекта заключается в том, чтобы на момент запуска системы и в течение всего времени ее эксплуатации можно было обеспечить:

- требуемую функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;
- требуемую пропускную способность системы;
- требуемое время реакции системы на запрос;
- безотказную работу системы в требуемом режиме, иными словами - готовность и доступность системы для обработки запросов пользователей;
- простоту эксплуатации и поддержки системы;
- необходимую безопасность.

Классификация типовых методов проектирования



Последовательность шагов проектирования системы





Схема реального процесса разработки информационной системы по каскадной модели

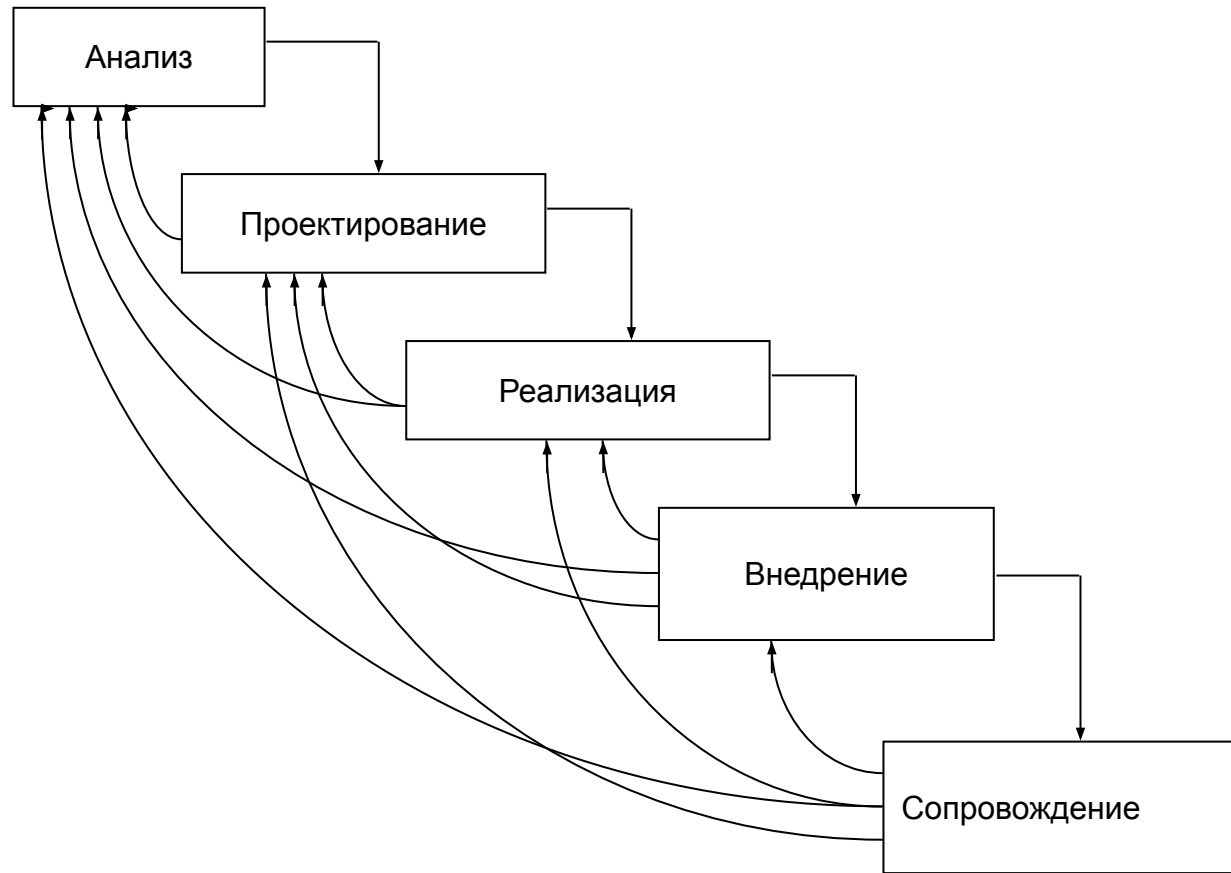
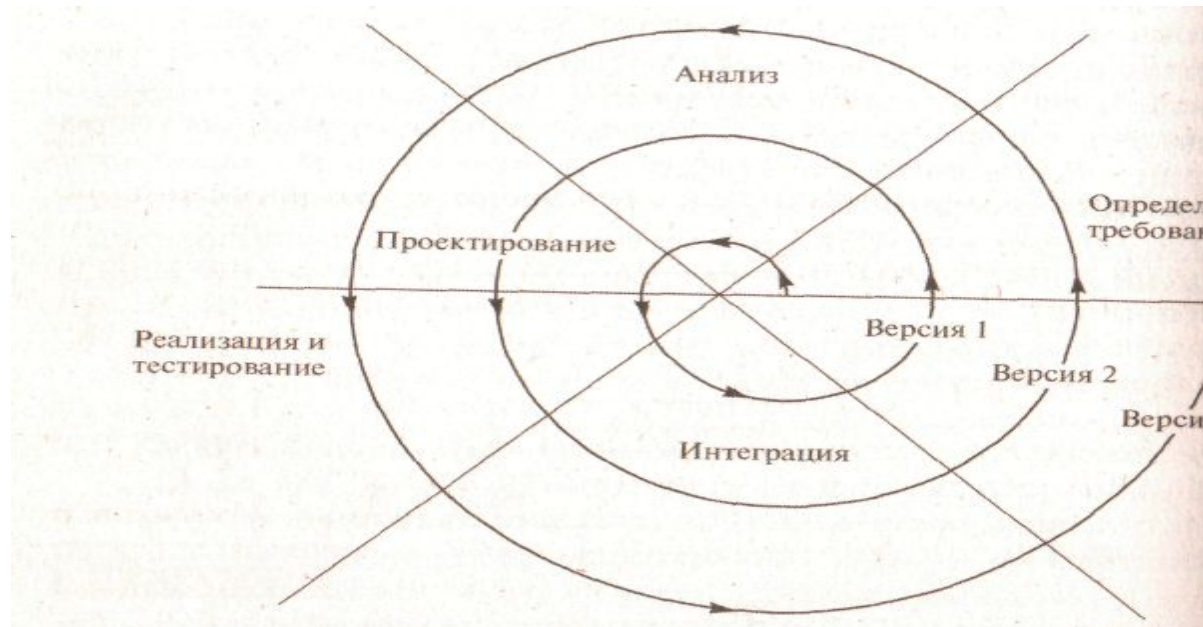


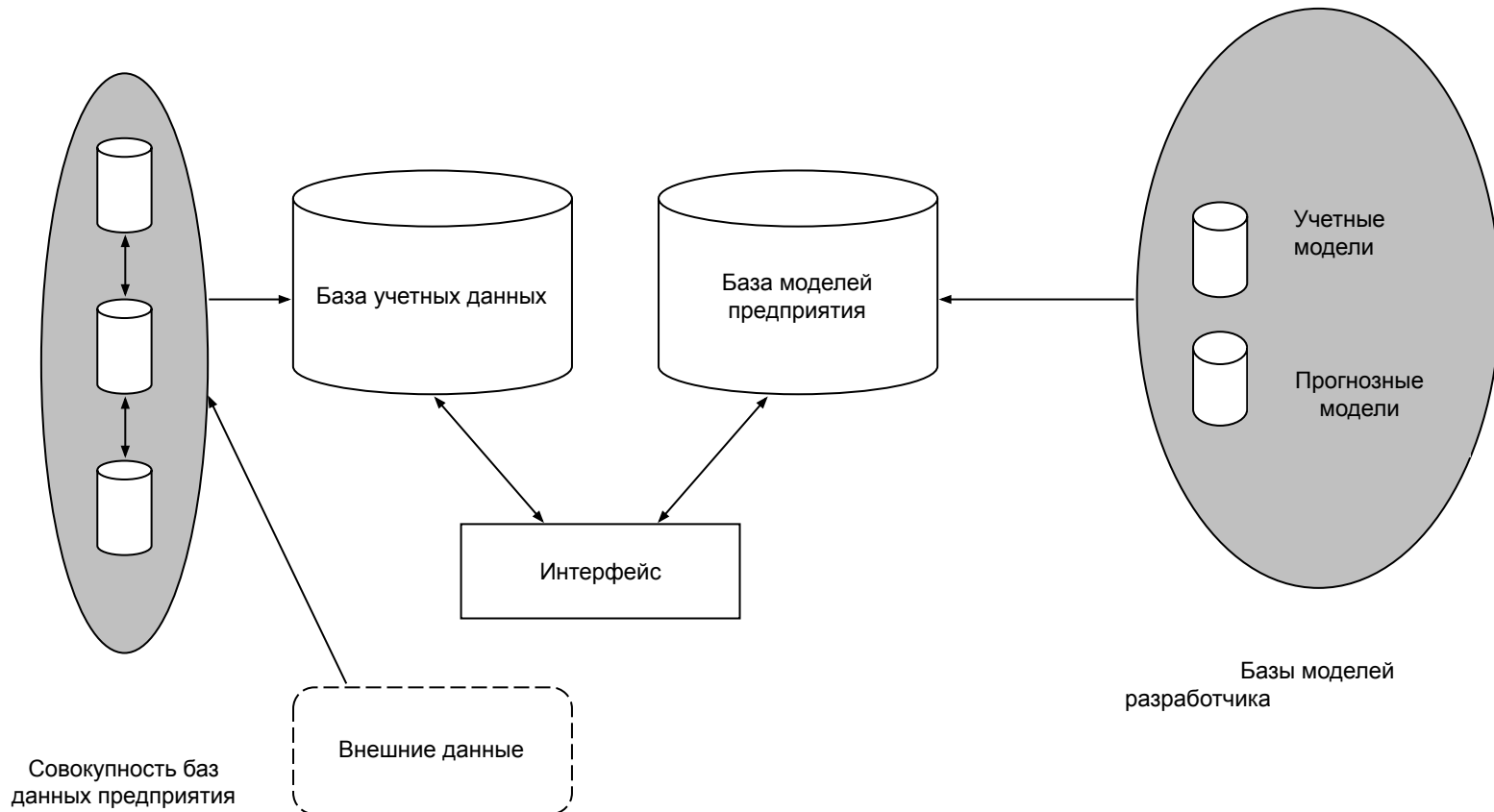
Схема реального процесса разработки информационной системы по спиральной модели



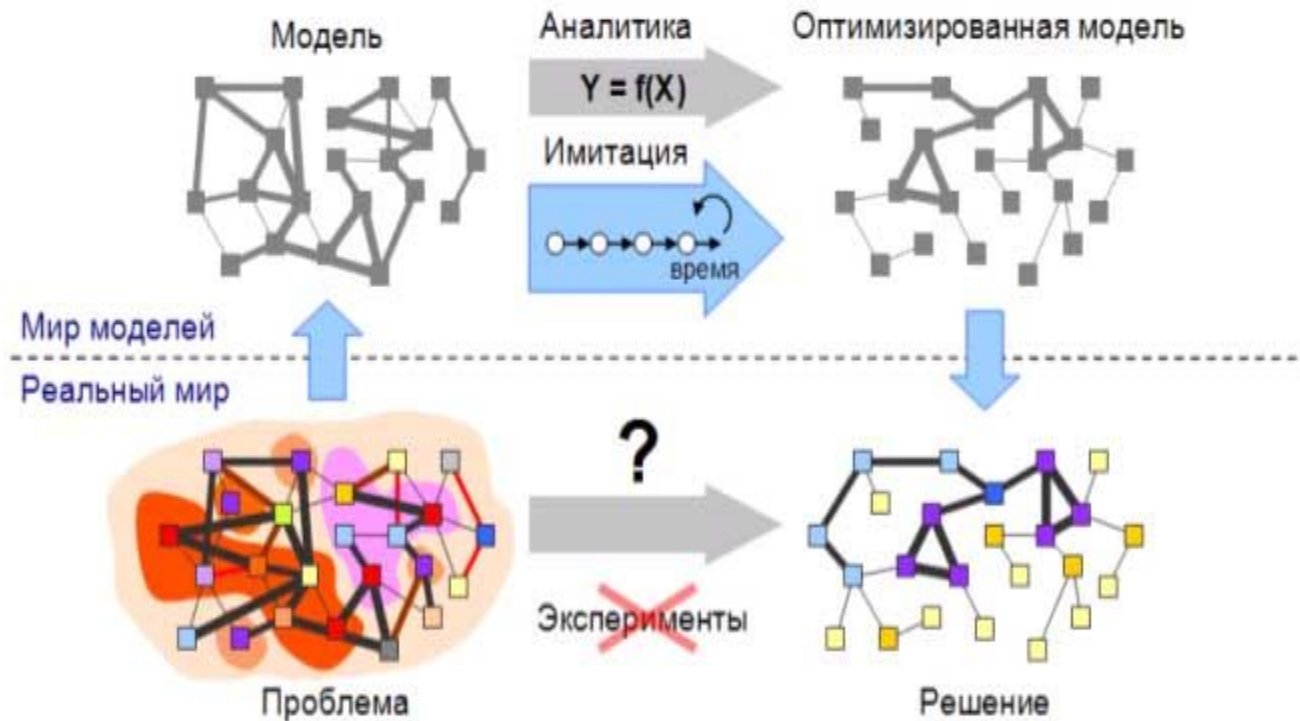
Основные возможности и преимущества быстрой разработки прототипа информационной системы



Построение ИС согласно принципу моделирования



Аналитическое (статическое) и имитационное (динамическое) моделирование



Моделирование — один из способов исследования и устранения проблем, возникающих в окружающем нас мире

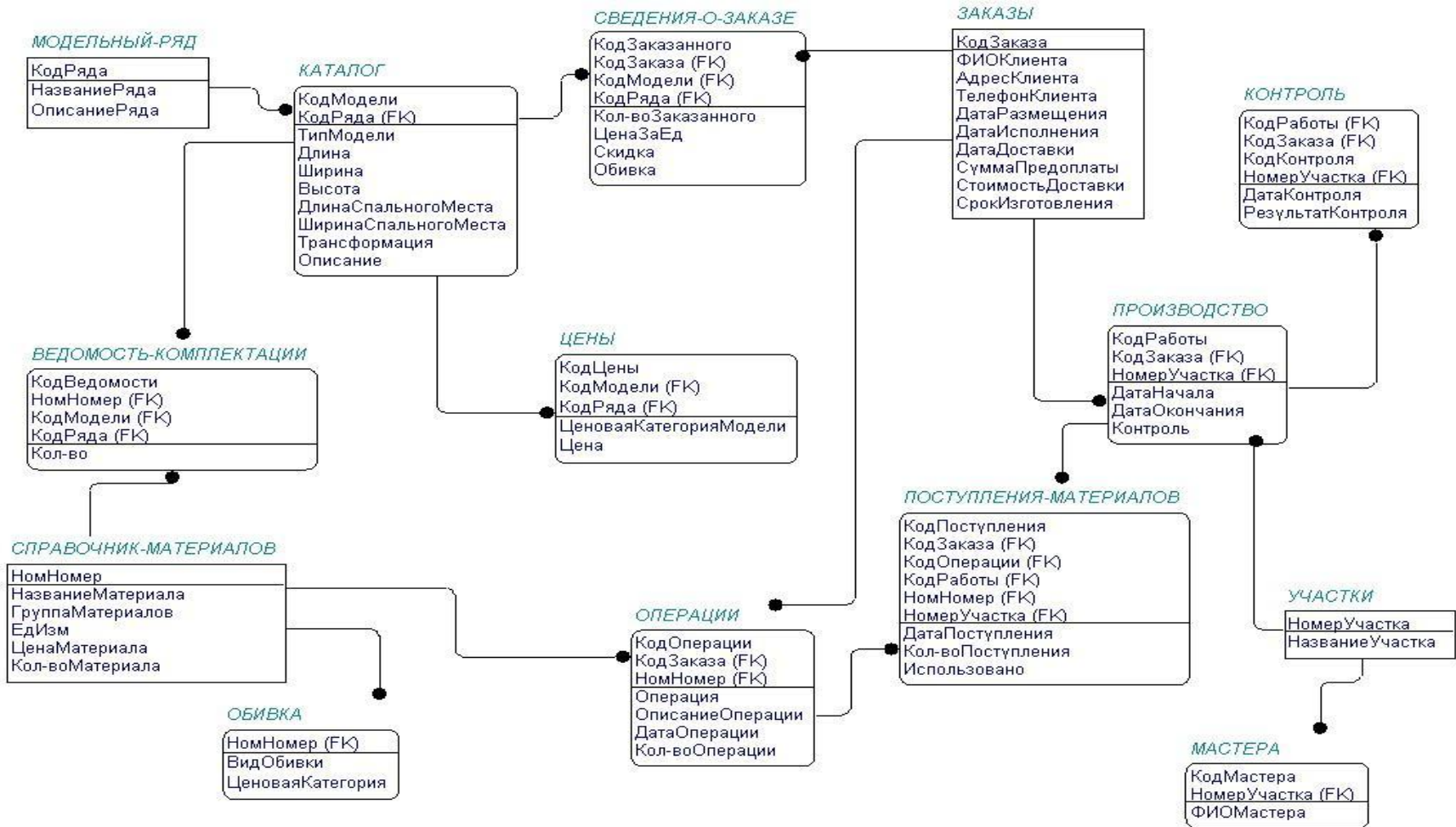


Три подхода имитационного моделирования

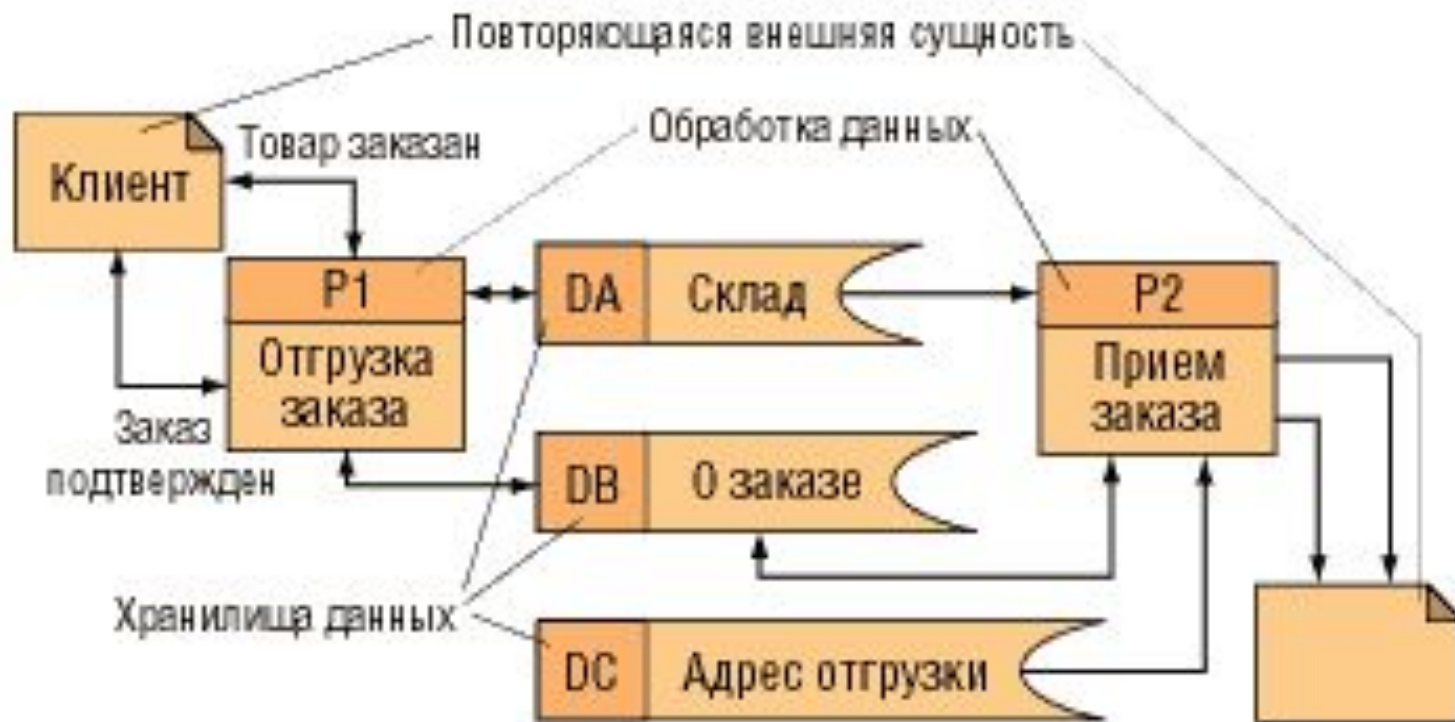


IDEF1X-диаграмма. Состав атрибутов сущностей

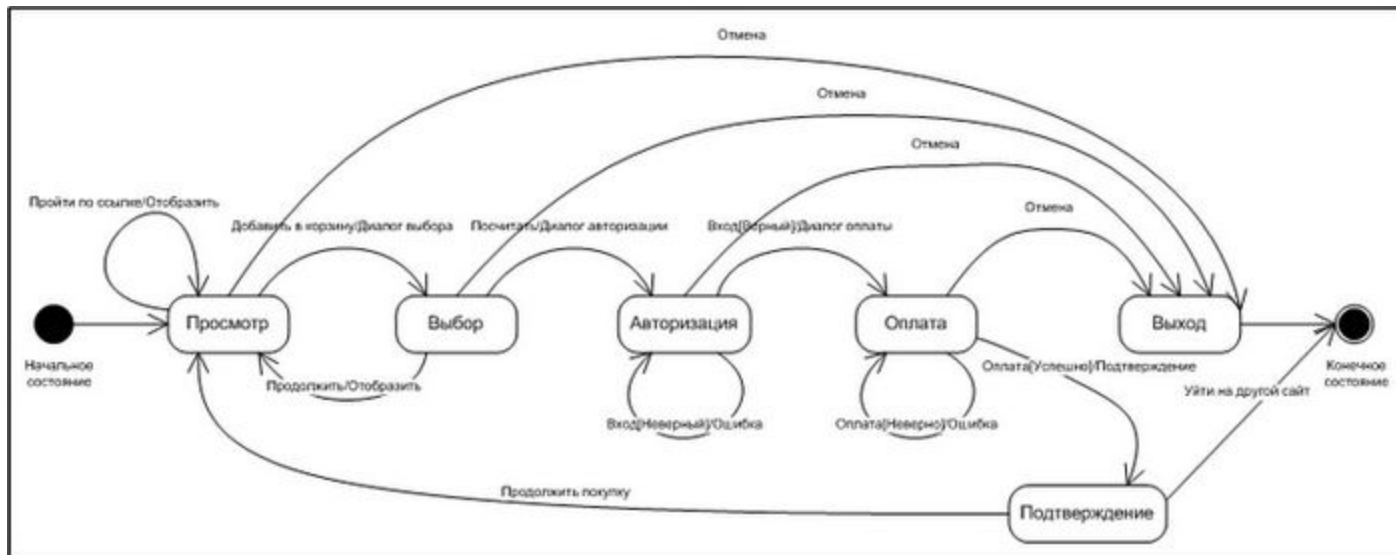
Диаграммы "сущность-связь" (Entity-Relationship Diagrams, ERD), служат для формализации информации о сущностях и их отношениях (



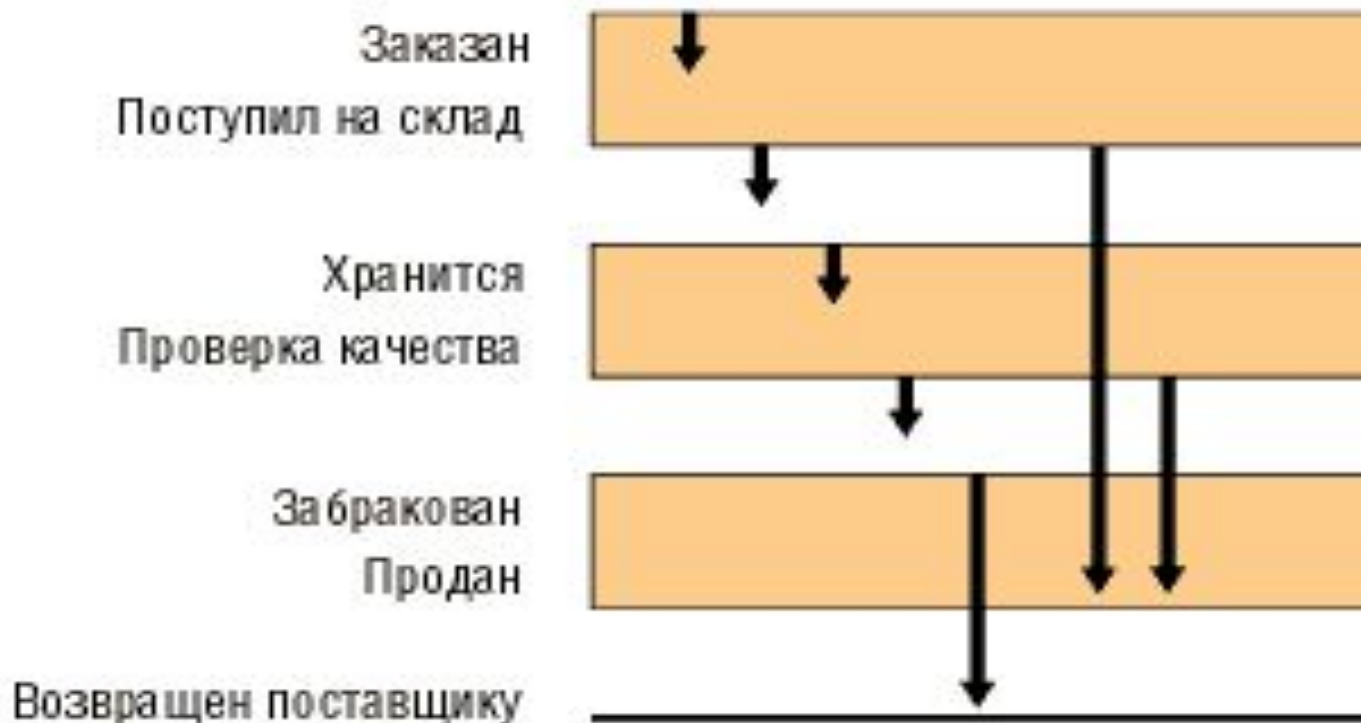
Диаграммы потоков данных (Data Flow Diagrams, DFD), которые служат для формализации представления функций системы



Пример диаграммы переходов состояний



Диаграммы переходов состояний (State Transition Diagrams, STD), которые отражают поведение системы, зависящее от времени; диаграммы жизненных циклов сущностей относятся именно к этому классу диаграмм.



Цели ИС

- *Цели*, стоящие перед информационной системой (ИС), должны соответствовать миссии и списку КФУ предприятия и, как правило, следуют из структурных проблем предприятия. ИС призвана устранить те проблемы, которые можно решить при помощи оптимизации информационного аспекта деятельности предприятия.

- В идеале, на предприятии должен работать единый программный комплекс, связывающий работу разных специалистов на различных рабочих местах в единую технологическую и информационную цепочку. Для этого составляется список первоначальных целей, стоящих перед ИС.

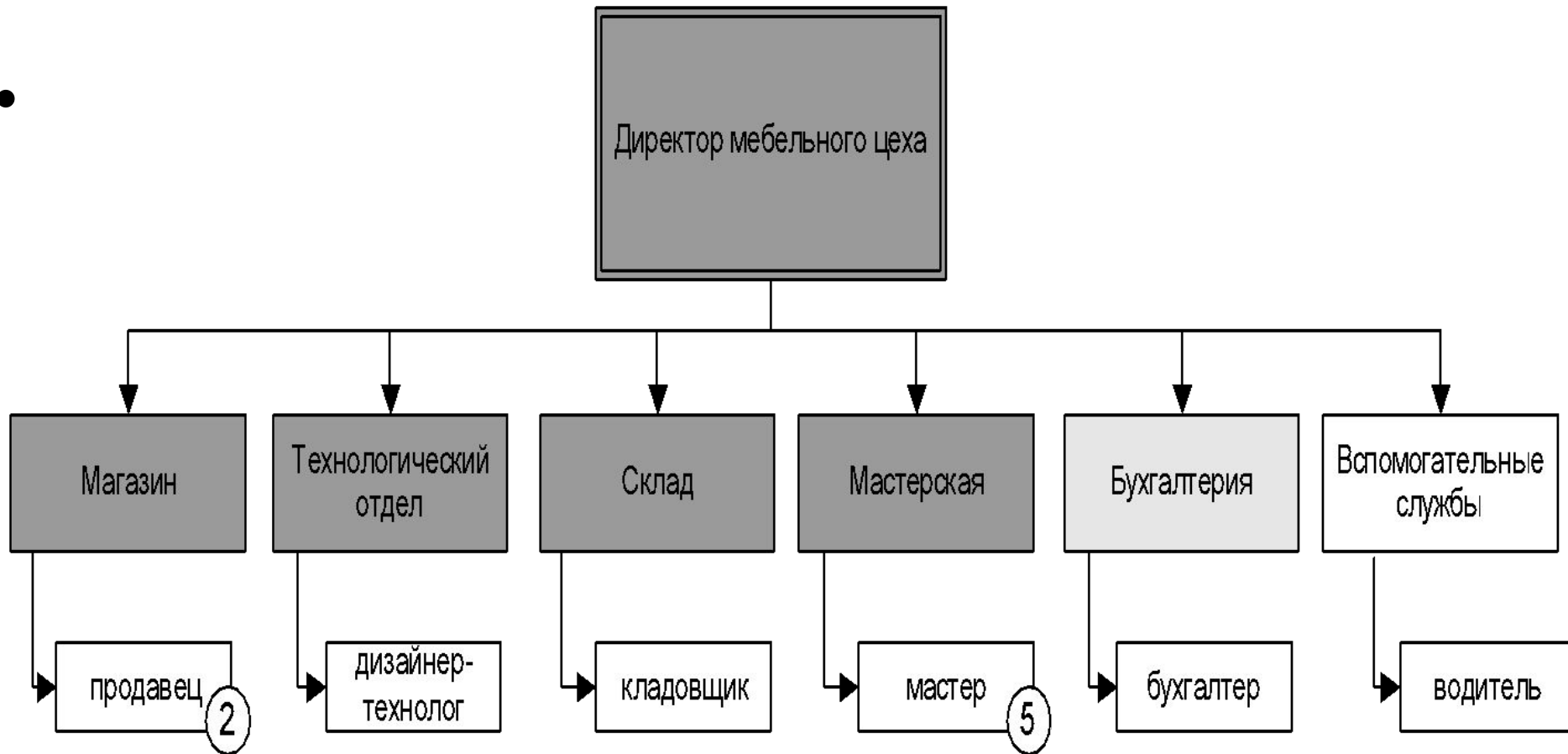
Список задач

- Проанализировать обмен информацией между отделами цеха и директором. По возможности упростить передачу, сделать ее более эффективной, устранить дублирование информации. Автоматизировать документооборот внутри предприятия. Это позволит сократить ручные операции, ускорить обработку информации, повысить точность учета и, самое главное, сократить время обработки заказов. Руководство предприятия в любой момент времени будет иметь необходимую оперативную информацию.

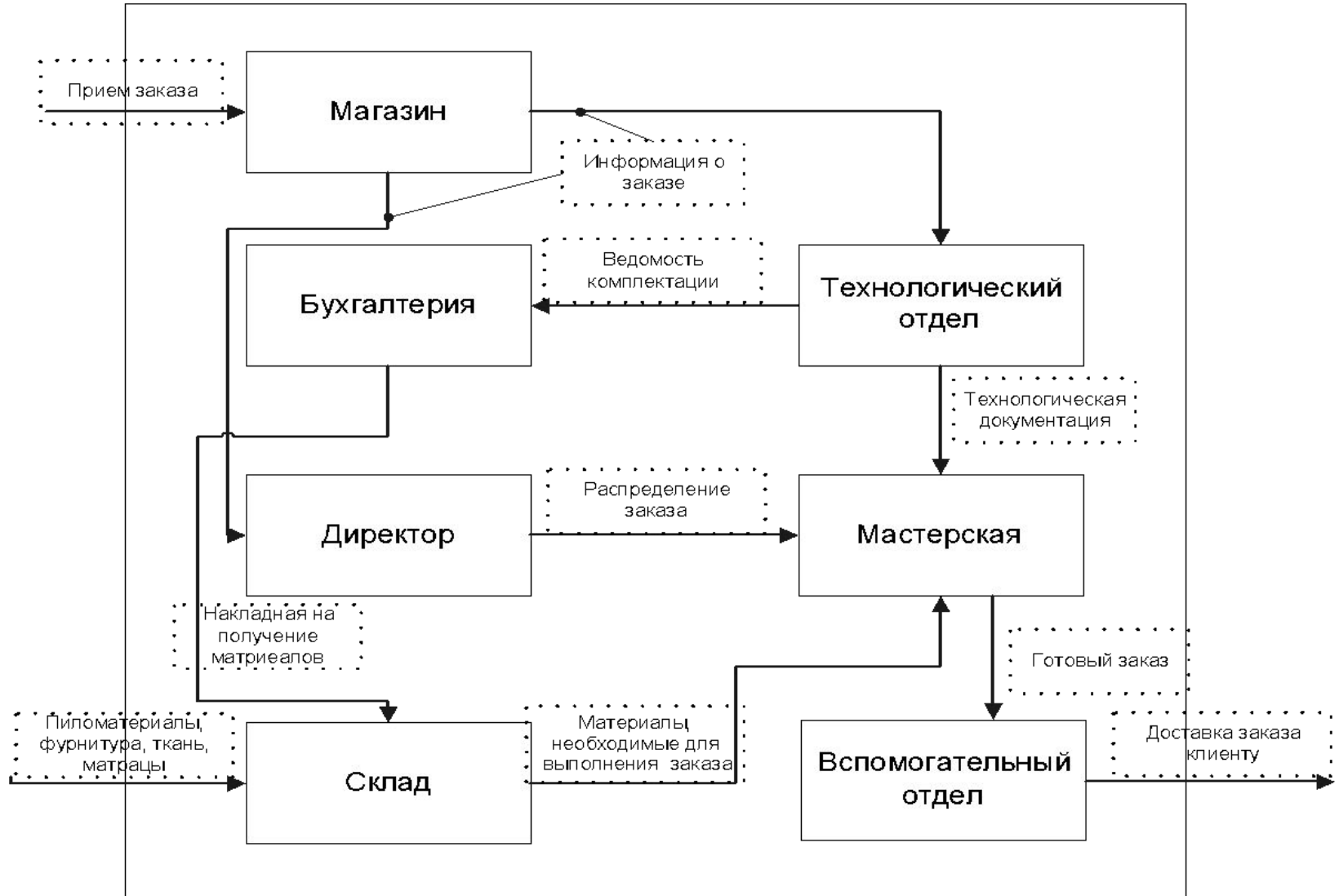
Структура предприятия

Построение структуры предприятия можно разбить на три шага: построение организационной модели, построение функциональной модели и построение информационной модели.

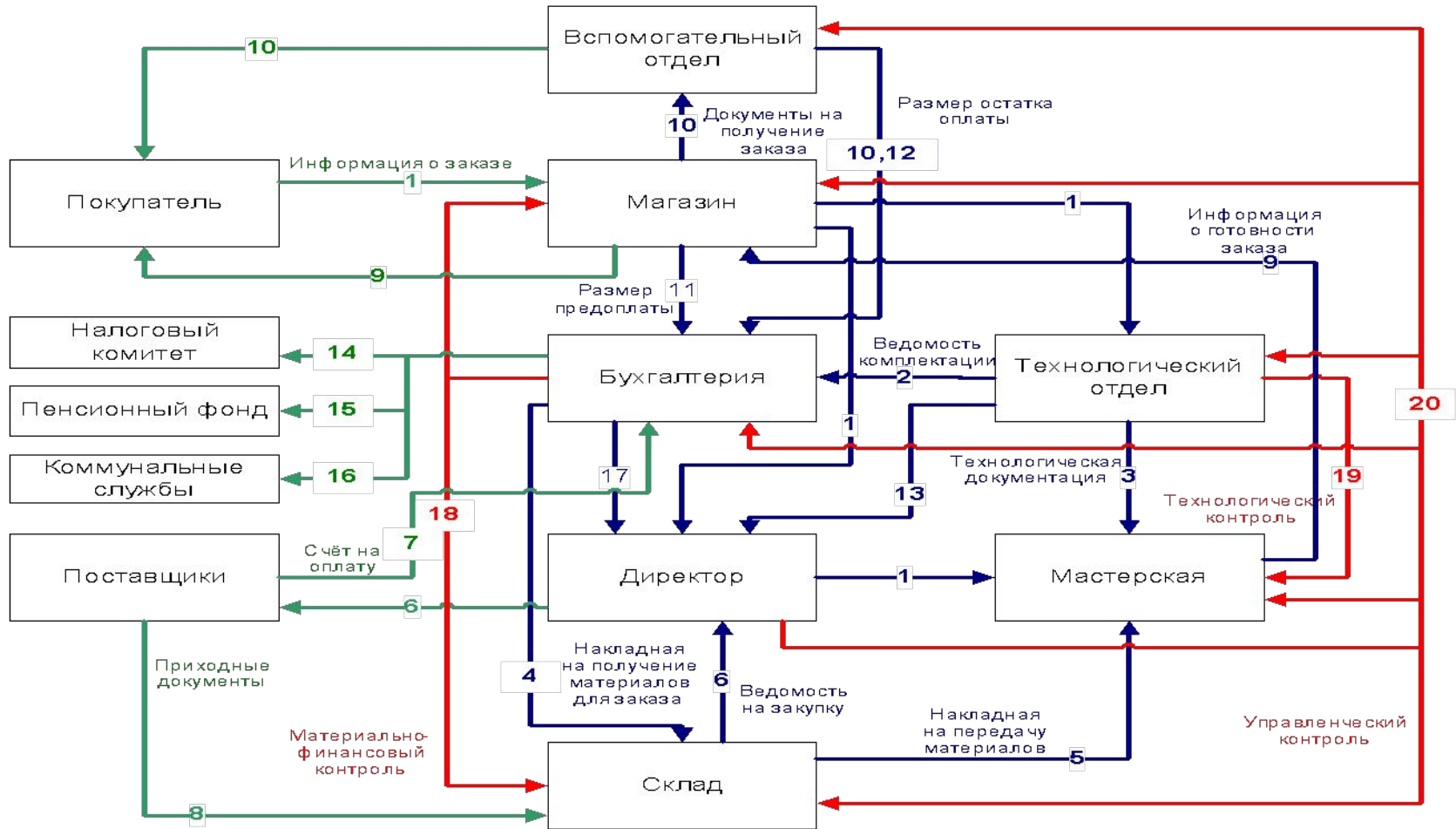
Организационная модель предприятия



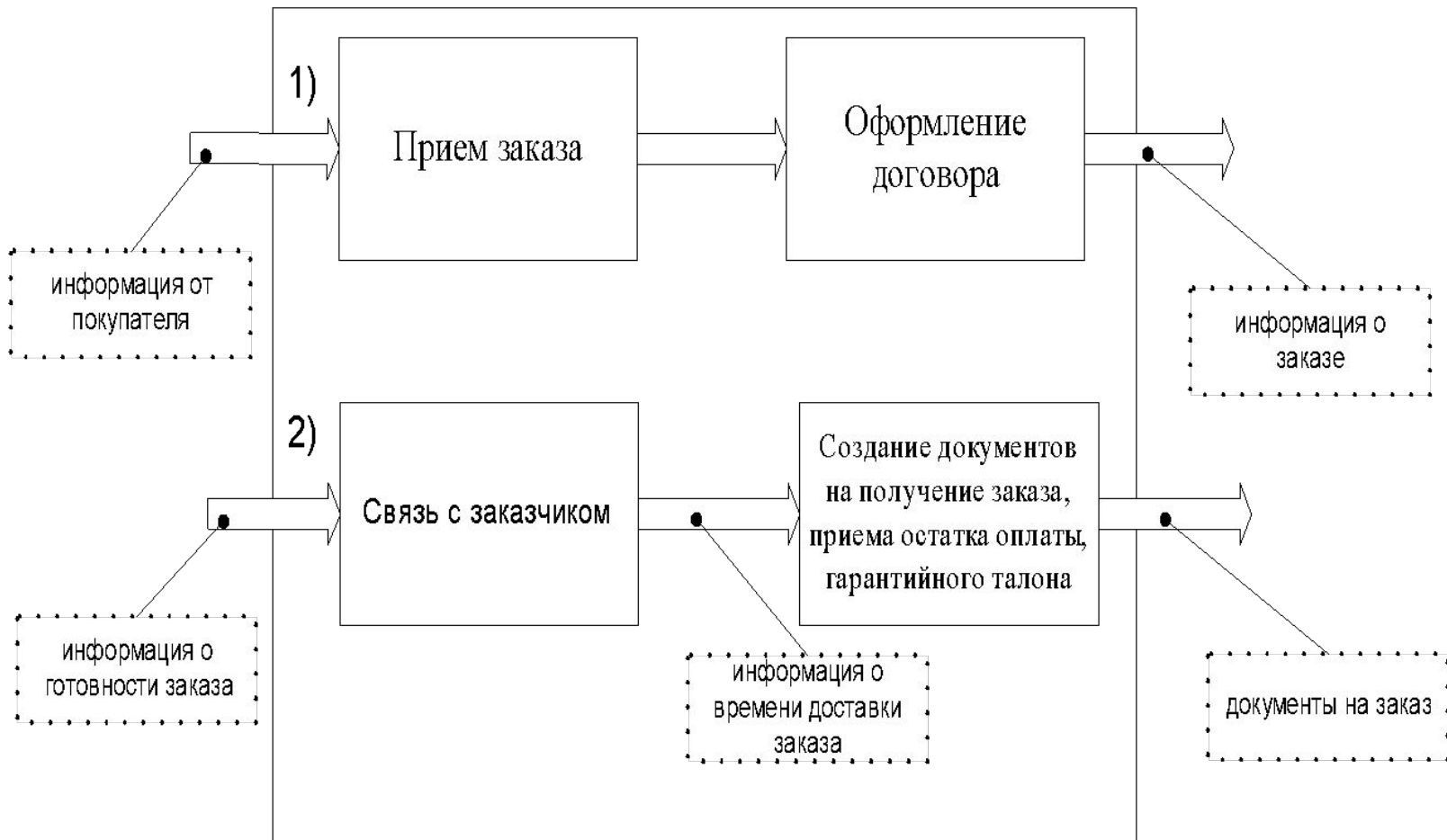
Функциональная модель предприятия



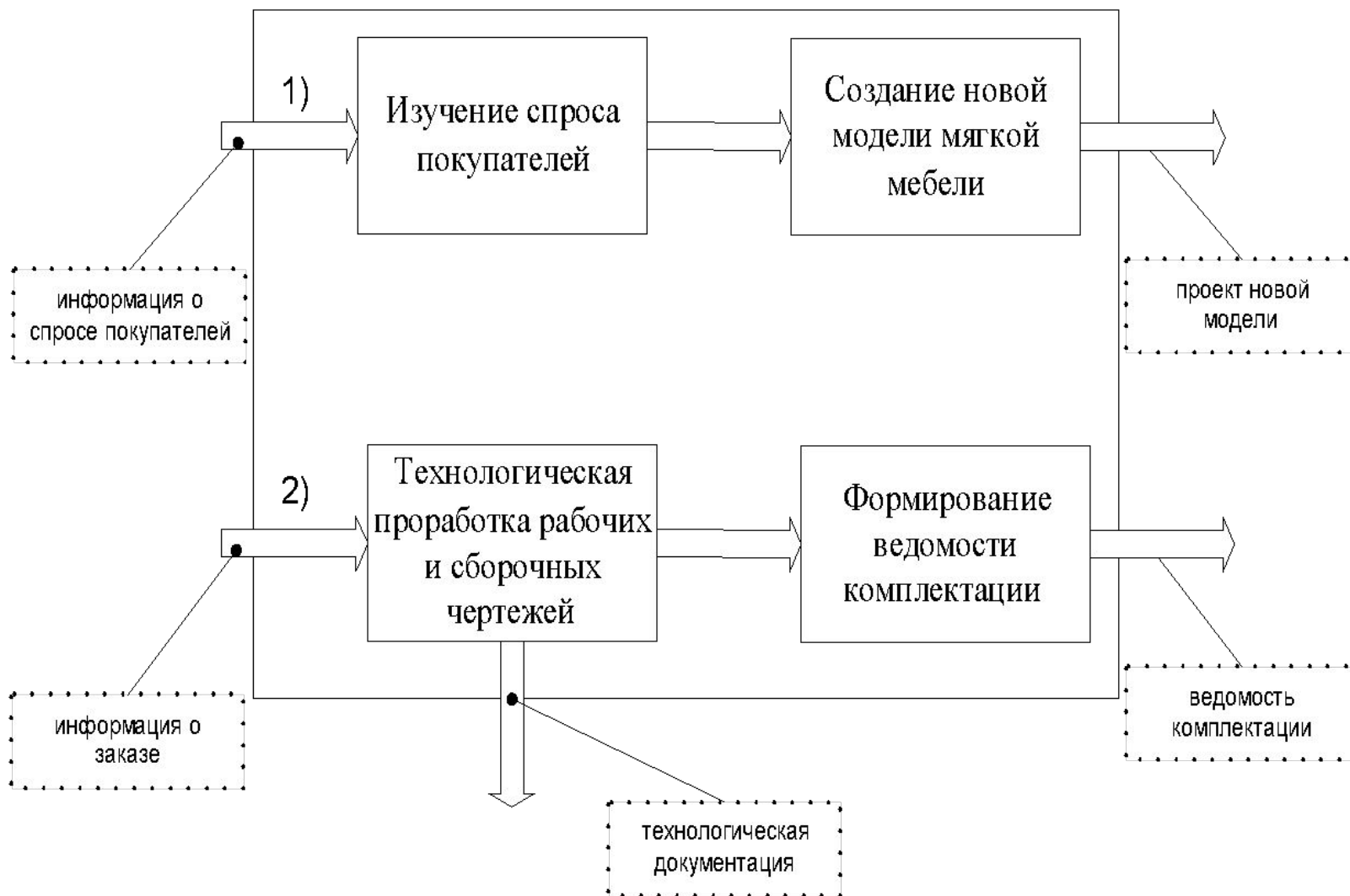
Информационная модель предприятия



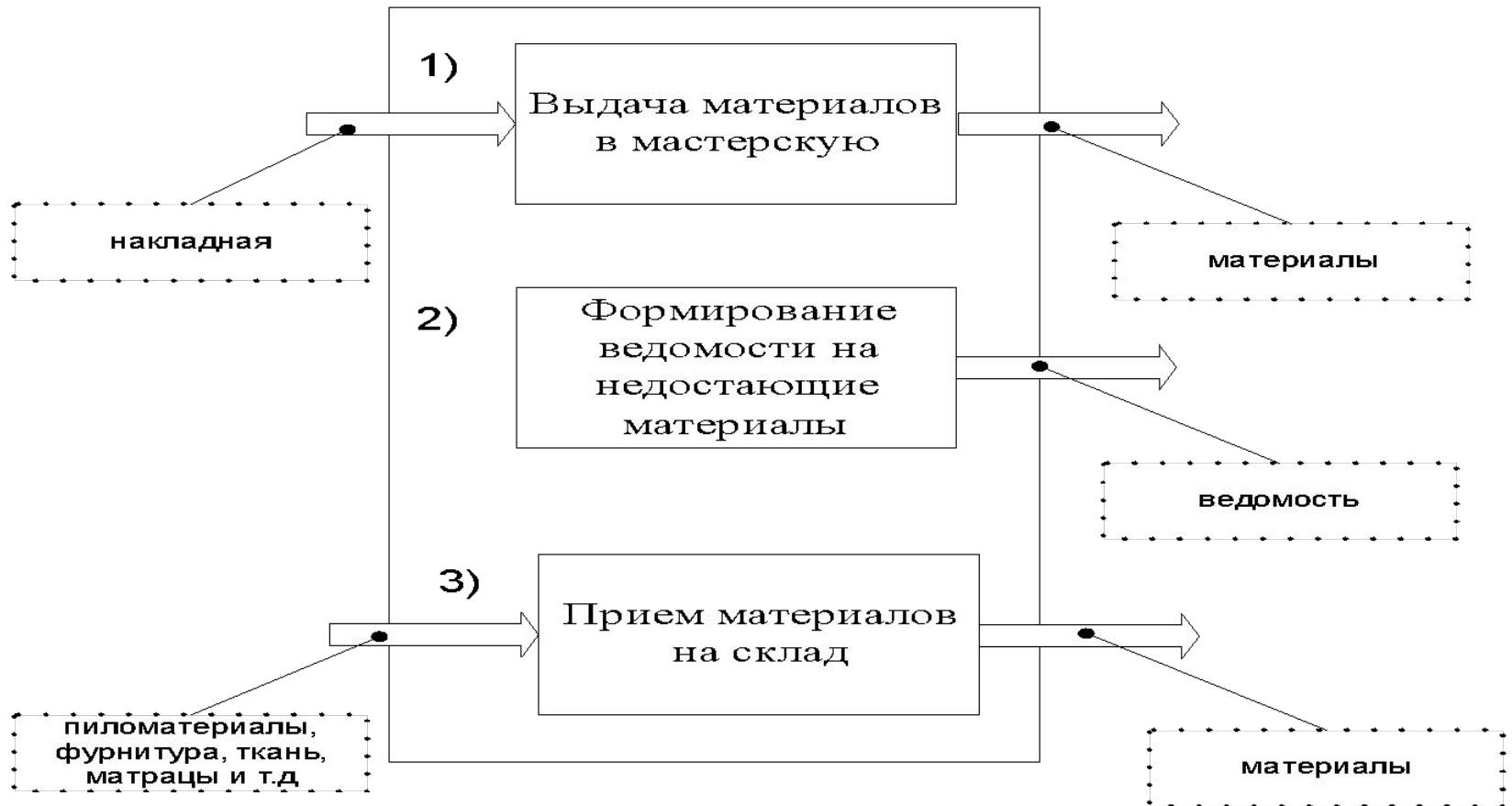
Функциональная модель магазина



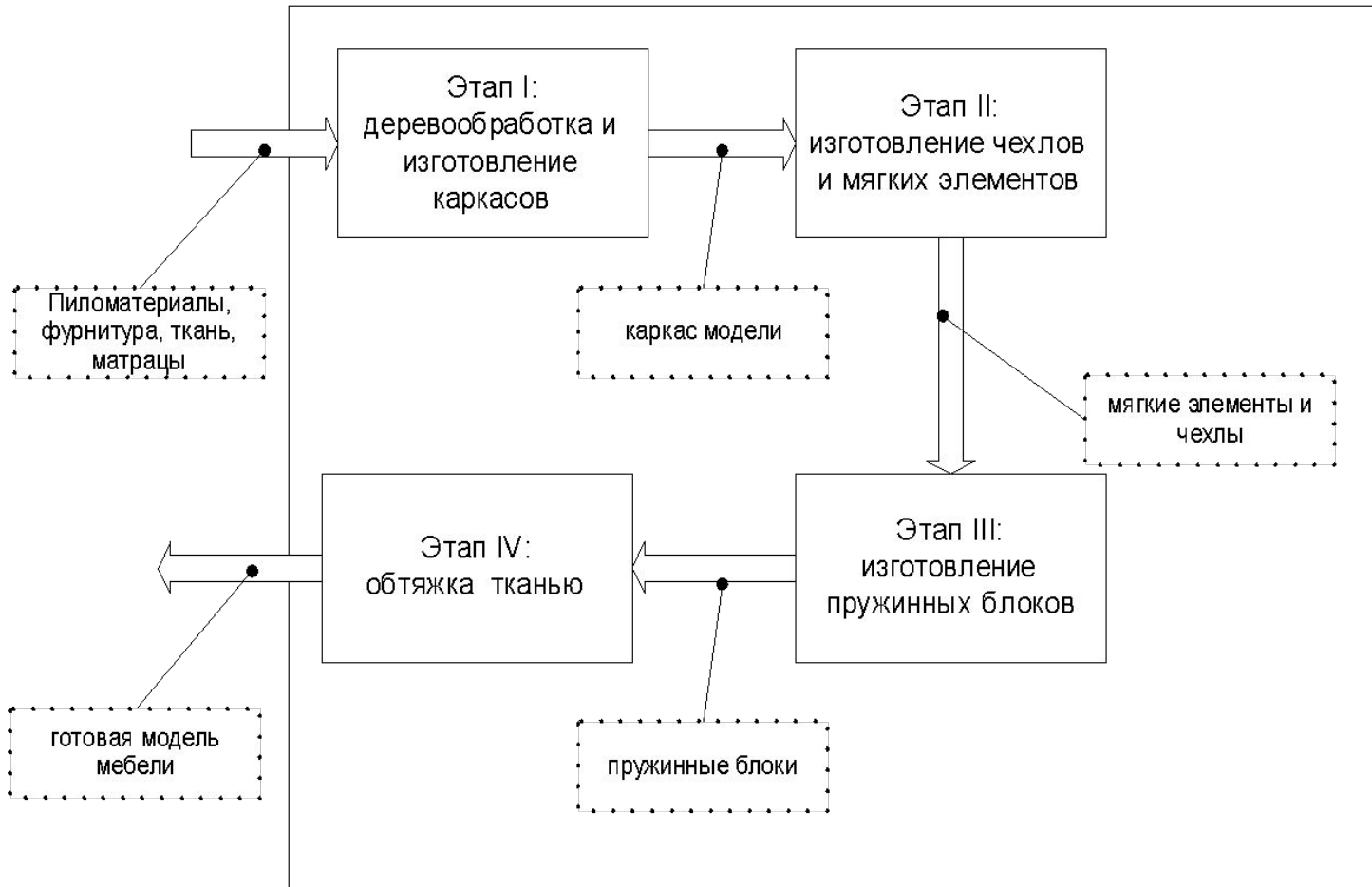
Функциональная модель технологического отдела



Функциональная модель складов



Функциональная модель цеха



Любой проект по автоматизации должен начинаться с самой важной и нелегкой задачи – согласования планируемых результатов проекта с миссией и стратегическими целями предприятия. Решение о целесообразности реализации проекта должно приниматься исходя из того, насколько он соответствует стратегии, целям компании и путям их достижения.

Потребительские свойства информационной системы:

- **Функциональная полнота**- система должна обеспечивать получение любой необходимой пользователю информации на некотором заданном интервале времени.
- **Временная обеспеченность** - возможность получения нужной информации в требуемое время.
- **Функциональная надежность**- получение безошибочной информации в заданные сроки.
- **Эффективность** - система должна приносить пользу.
- **Адаптивность** - система должна обладать способностью приспособливаться к частично изменившимся условиям объекта и обеспечивать устойчивое функционирование на большом интервале времени.
- **Иерархическая агрегируемость** - возможность быть составной частью с системой более высокого уровня.

Под целью автоматизации решения задачи подразумевается получение определенных значений экономического эффекта в сфере управления какими-либо процессами системы или снижение стоимостных и трудовых затрат на обработку информации, улучшение качества и достоверности получаемой информации, повышение оперативности ее обработки и т.д., т.е. получение косвенного и прямого эффекта от внедрения данной задачи.

Под экономической сущностью решаемой задачи понимаются состав экономических показателей, рассчитываемых при ее решении, документы, в которые заносятся эти показатели, перечень исходных показателей, необходимых для получения результатных и наименования тех первичных документов, в которых они содержатся.

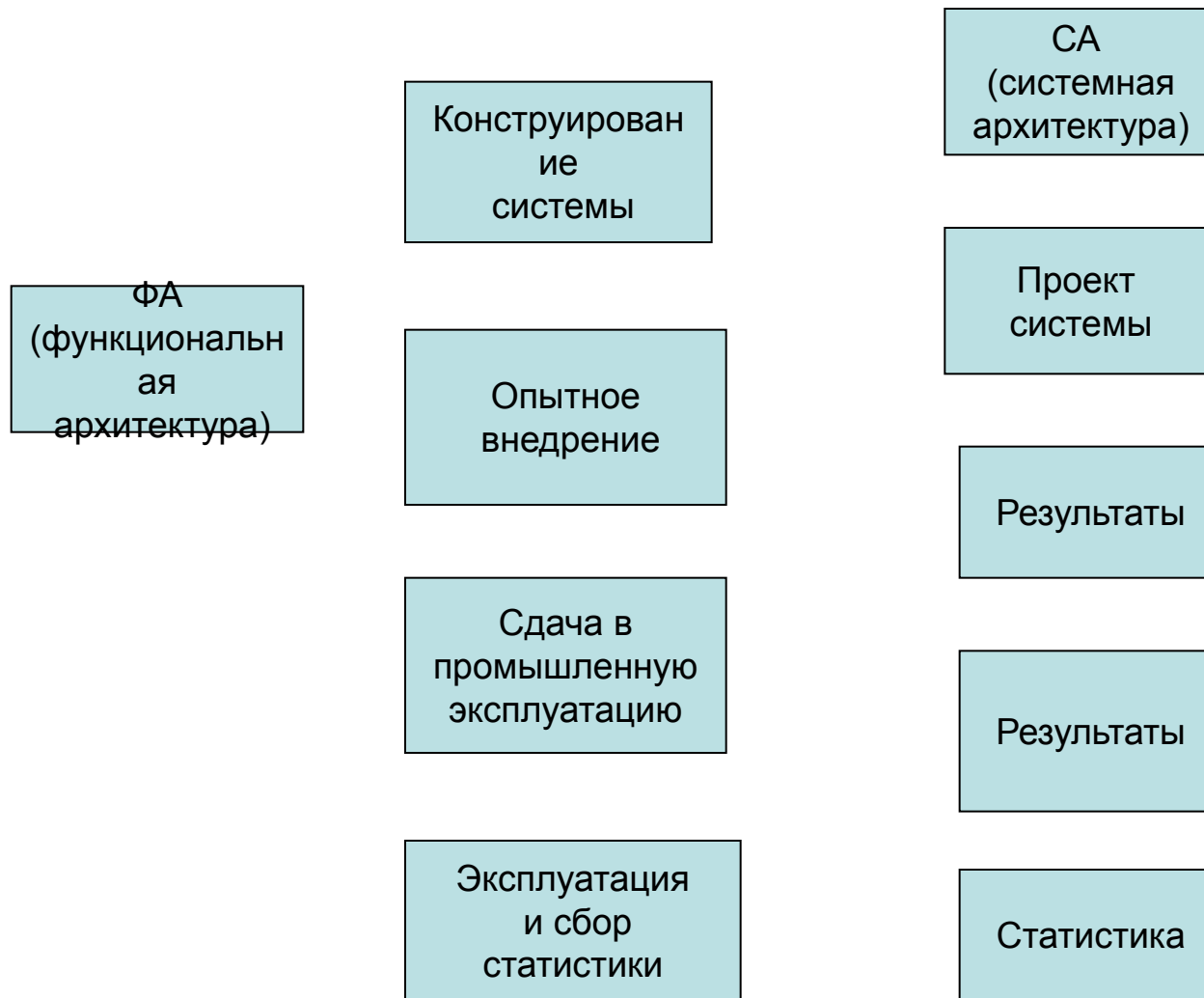
Характеристики классов технологий проектирования

Класс технологии проектирования	Степень автоматизации	Степень типизации	Степень адаптивности
Каноническое проектирование	Ручное проектирование	Оригинальное проектирование	Реконструкция
Индустриальное автоматизированное проектирование	Компьютерное проектирование	Оригинальное проектирование	Реструктуризация модели (генерация ИС)
Индустриальное типовое проектирование	Компьютерное проектирование	Типовое сборочное проектирование	Параметризация и реструктуризация модели (конфигурация ИС)

Средства проектирования должны быть:

- в своем классе инвариантными к объекту проектирования;
- охватывать в совокупности все этапы жизненного цикла ЭИС;
- технически, программно и информационно совместимыми;
- простыми в освоении и применении;
- экономически целесообразными.

Технологическая схема обобщенного жизненного цикла



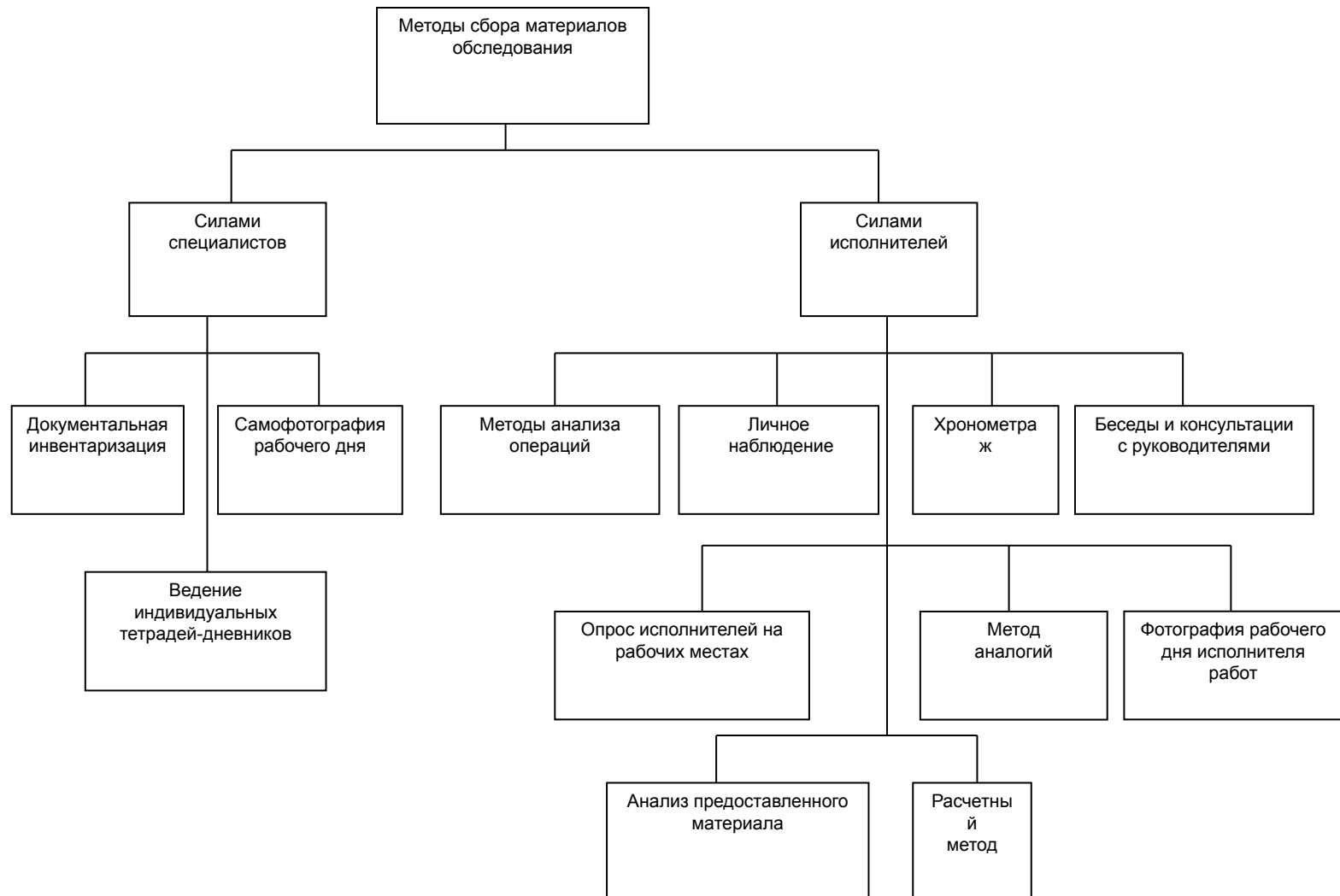
- Этап по составлению функциональной архитектуры (ФА), представляющей собой совокупность функциональных подсистем и связей между ними, является наиболее ответственным с точки зрения качества всей последующей разработки.
- Построение системной архитектуры (СА) на основе ФА предполагает выделение элементов и модулей информационного, технического, программного обеспечения и других обеспечивающих подсистем, определение связей по информации и управлению между выделенными элементами и разработку технологии обработки информации.

Внедрение разработанного проекта

Процесс предполагает выполнение следующих этапов: опытное внедрение и промышленное внедрение.

- Этап опытного внедрения заключается в проверке работоспособности элементов и модулей проекта, устранении ошибок на уровне элементов и связей между ними.
- Этап сдачи в промышленную эксплуатацию заключается в организации проверки проекта на уровне функций и контроля соответствия его требованиям, сформулированным на стадии системного анализа.

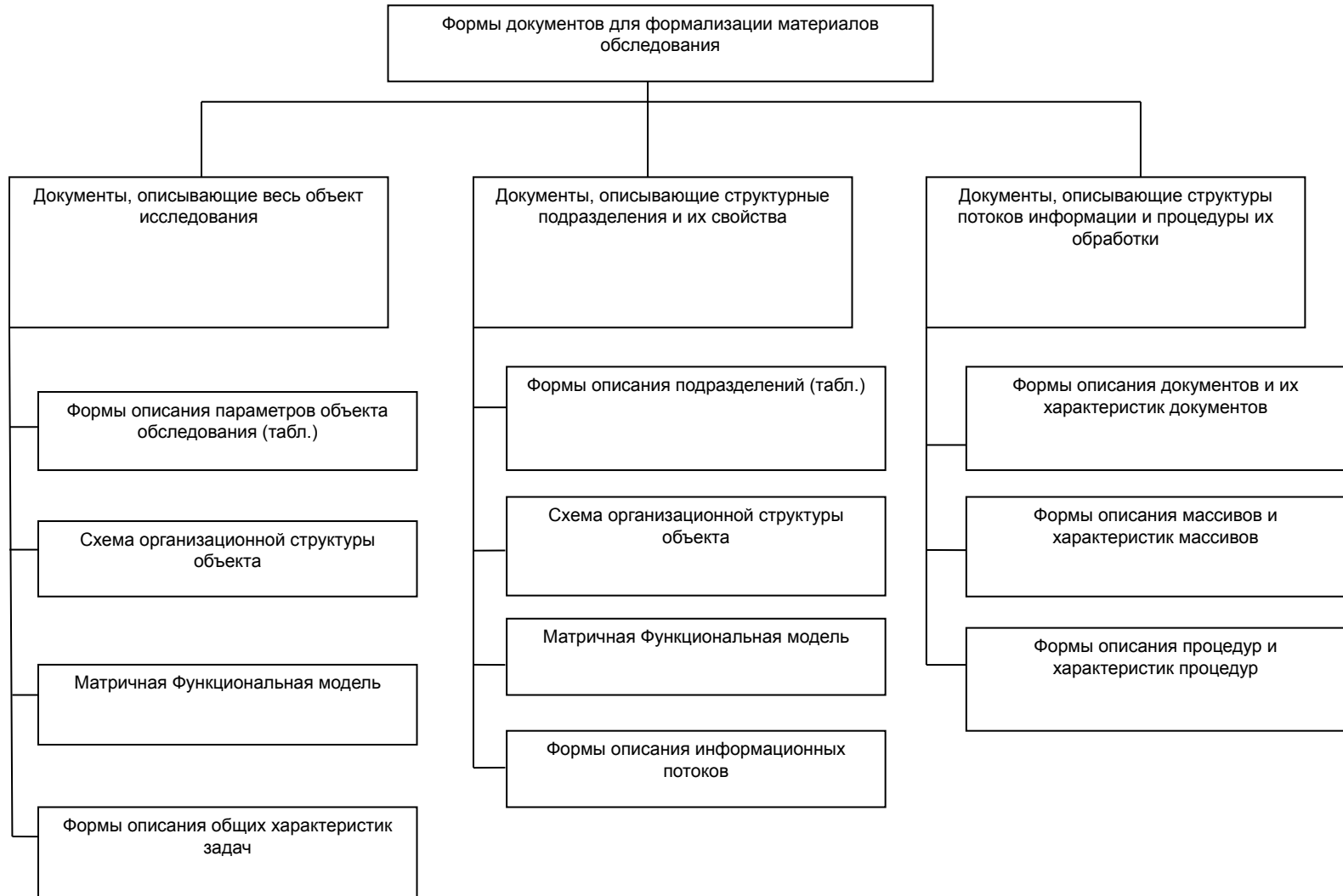
Схема классификации «Методы сбора материалов обследования»



Программа обследования

№ п/п	Наименование вопроса	Источник информации	Получатель информации
1	Цель функционирования объекта	Руководитель предприятия	Руководитель проекта
2	Основные параметры объекта	Руководитель предприятия	Руководитель проекта
3	Организационная структура объекта	Секретарь руководителя	Зам. руководителя проекта

Формы документов для формализации материалов обследования



Классификация информационных систем



Функциональное назначение модулей корпоративной ИС.

Подсистема маркетинга	Производственные подсистемы	Финансовые и учетные подсистемы	Подсистема кадров (человеческих ресурсов)	Прочие подсистемы (например, ИС руководства)
Исследование рынка и прогнозирование продаж	Планирование объемов работ и разработка календарных планов	Управление портфелем заказов	Анализ и прогнозирование потребности в трудовых ресурсах	Контроль за деятельностью фирмы
Управление продажами	Оперативный контроль и управление производством	Управление кредитной политикой	Ведение архивов записей о персонале	Выявление оперативных проблем
Рекомендации по производству новой продукции	Анализ работы оборудования	Разработка финансового плана	Анализ и планирование подготовки кадров	Анализ управленческих и стратегических ситуаций
Анализ и установление цены	Участие в формировании заказов поставщикам	Финансовый анализ и прогнозирование		Обеспечение процесса выработки стратегических решений
Учет заказов	Управление запасами	Контроль бюджета, бухгалтерский учет и расчет зарплаты		

Классификация рынка информационных систем

Локальные системы	Малые интегрированные системы	Средние интегрированные системы	Крупные интегрированные системы (IC)
•БЭСТ •Инотек •Инфософт •Супер-Менеджер •Турбо-Бухгалтер •Инфо-Бухгалтер	•Concorde XAL Exact •NS-2000 Platinum PRO/MIS •Scala SunSystems •БЭСТ-ПРО •1С-Предприятие •БОСС-Корпорация •Галактика •Парус •Ресурс •Эталон	•Microsoft-Business Solutions - Navision, •Axapta D Edwards (Robertson & Blums) •MFG-Pro (QAD/BMS) •SyteLine (СОКАП/SYMIX)	•SAP/R3 (SAP AG) •Baan (Baan) •BPCS (ITS/SSA) •Oracle Applications (oracle)

Проектирование ИС охватывает три основные области:

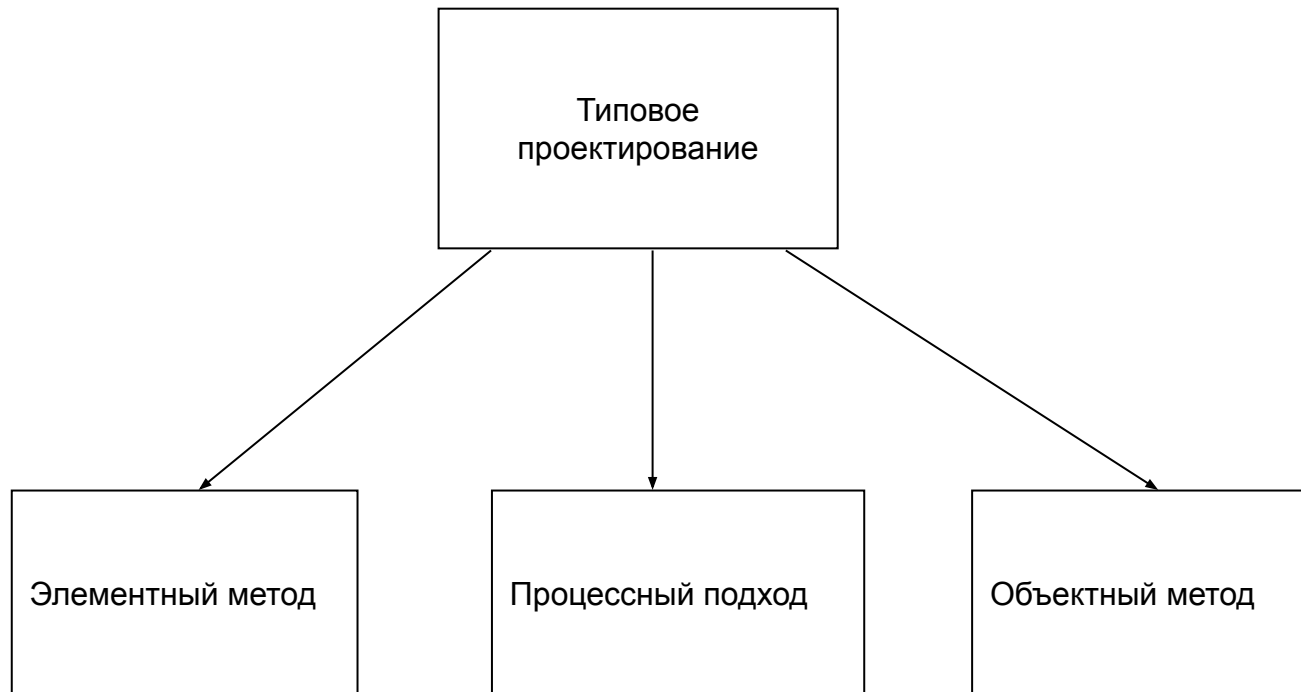
- проектирование объектов данных, которые будут реализованы в базе данных;
- проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т.п.

- Начальным этапом процесса создания ИС является моделирование бизнес-процессов, протекающих в организации и реализующих ее цели и задачи. Модель организации, описанная в терминах бизнес-процессов и бизнес-функций, позволяет сформулировать основные требования к ИС. Это фундаментальное положение методологии обеспечивает объективность в выработке требований к проектированию системы. Множество моделей описания требований к ИС затем преобразуется в систему моделей, описывающих концептуальный проект ИС. Формируются модели архитектуры ИС, требований к программному обеспечению (ПО) и информационному обеспечению (ИО). Затем формируется архитектура ПО и ИО, выделяются корпоративные БД и отдельные приложения, формируются модели требований к приложениям и проводится их разработка, тестирование и интеграция.

На этапе проектирования прежде всего формируются модели данных. Проектировщики в качестве исходной информации получают результаты анализа. Построение логической и физической моделей данных является основной частью проектирования базы данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных.

Главная цель проектирования процессов заключается в отображении функций, полученных на этапе анализа, в модули информационной системы. При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы.

Классификация типовых методов проектирования



Техническое задание должно быть:

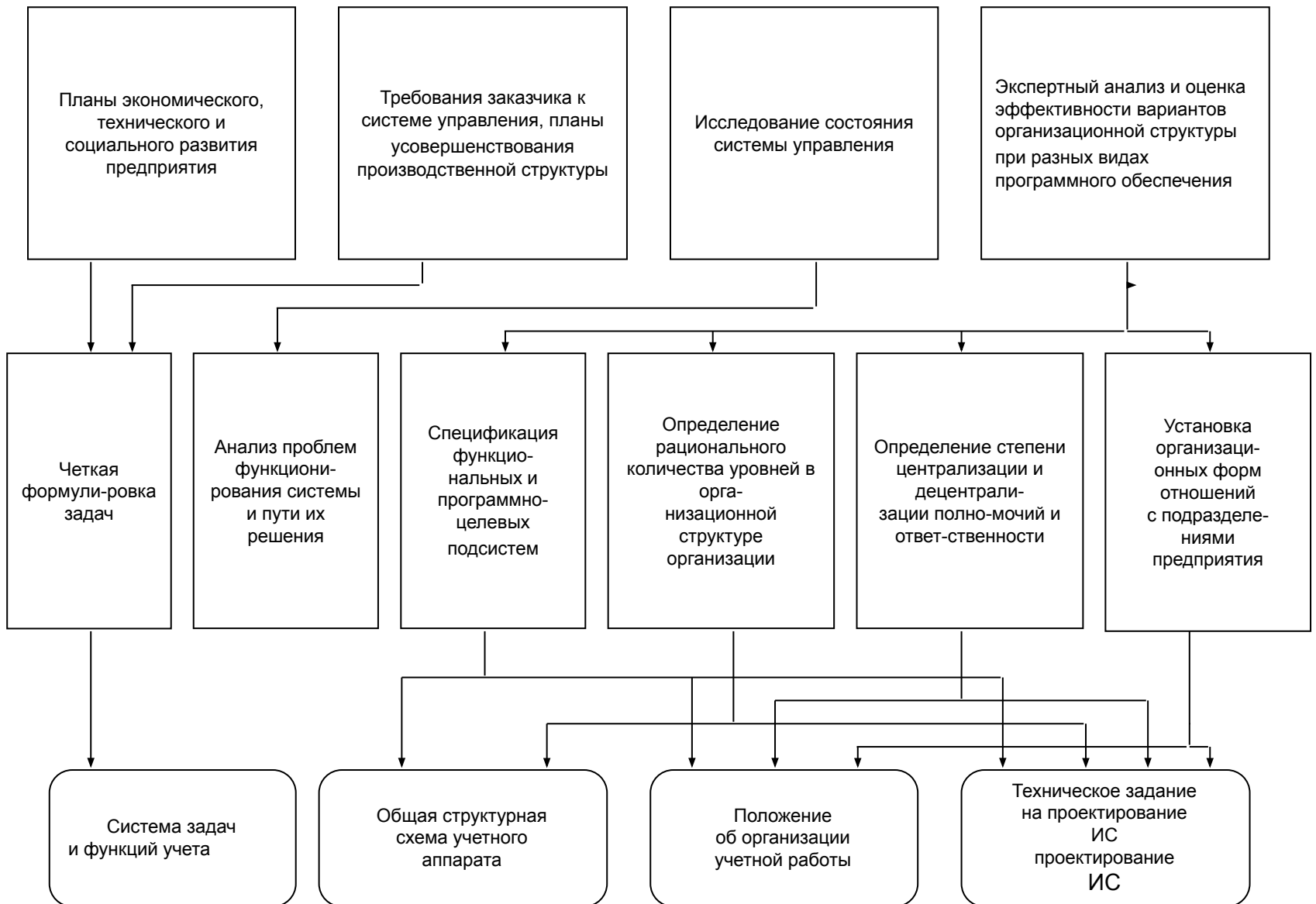
- точно сформулированным, что позволяет исключить неоднозначность его понимания разными исполнителями;
- должен быть полным, т.е. содержать описание всех аспектов функционирования системы, в том числе и ее реакцию на ошибочные действия пользователя;
- должен быть ясным, т.е. текст документа должен быть понятен и пользователю, и разработчику.

Техническое задание включает:

- требования к информационному обеспечению и системам разработки;
- требования к системе документирования; требования к составу и содержанию работ (согласно календарному плану);
- требования к функциям, реализуемым системой;
- требования к порядку контроля и приемки системы.

Эскизный проект (техническое предложение) — это документ, где излагаются основные концепции построения автоматизированной системы или отдельных ее подсистем. Поскольку в техническом задании только обозначаются цели, но не указываются пути их решения, то эскизный проект охватывает всю систему и описывает избранные пути решения задачи.

Последовательность работ первого этапа создания ИС



К основным компонентам ТЭО относятся:

- характеристика исходных данных о предметной области;
- обоснование цели создания ИС;
- обоснование автоматизируемых подразделений, комплекса автоматизируемых задач, выбора комплекса технических средств, программного и информационного обеспечения;
- разработка перечня организационно-технических мероприятий по проектированию системы;
- расчет и обоснование эффективности выбранного проекта;
- выводы о техническом уровне проекта и возможности дальнейших разработок.

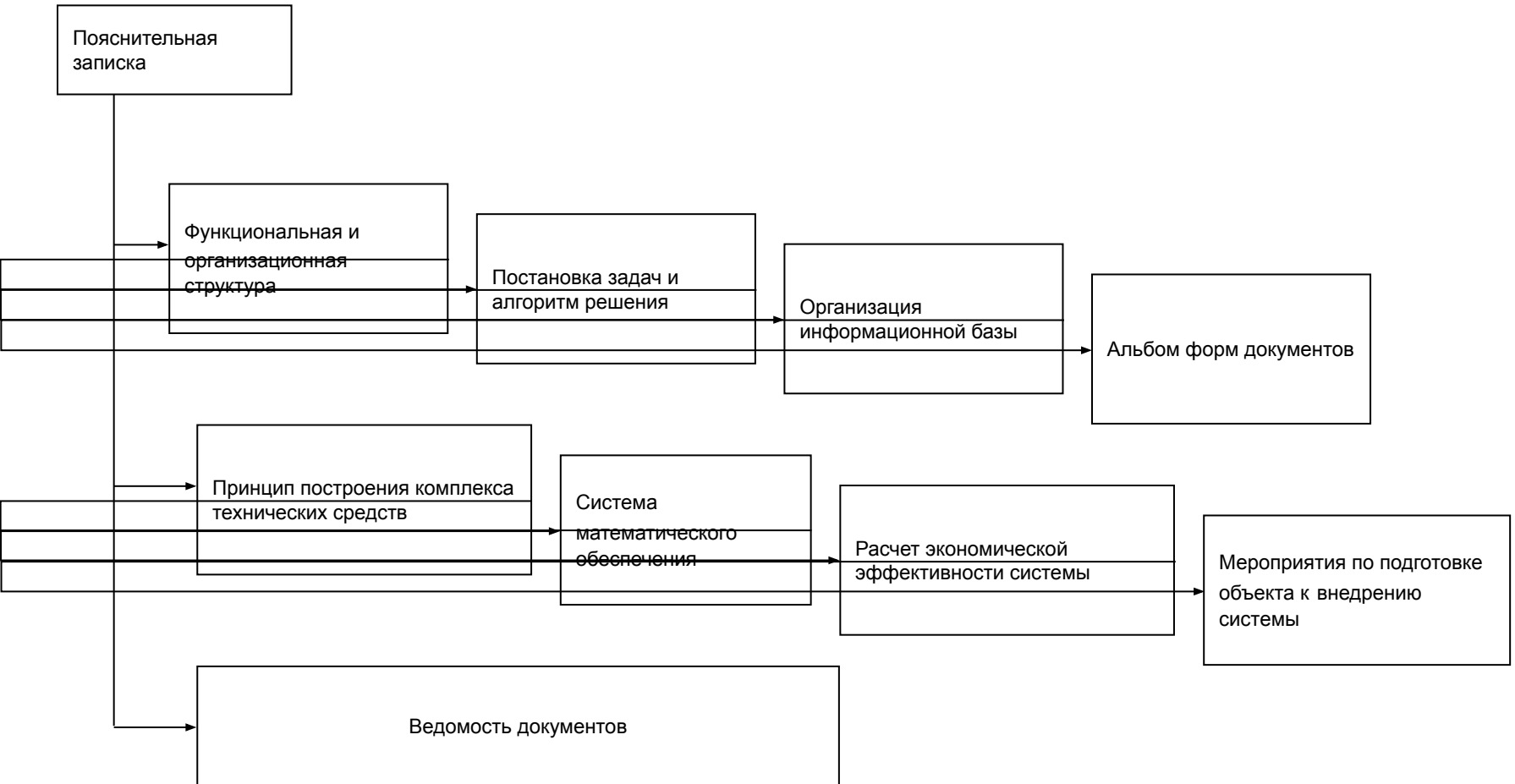
Схема структуры «Постановка задачи»

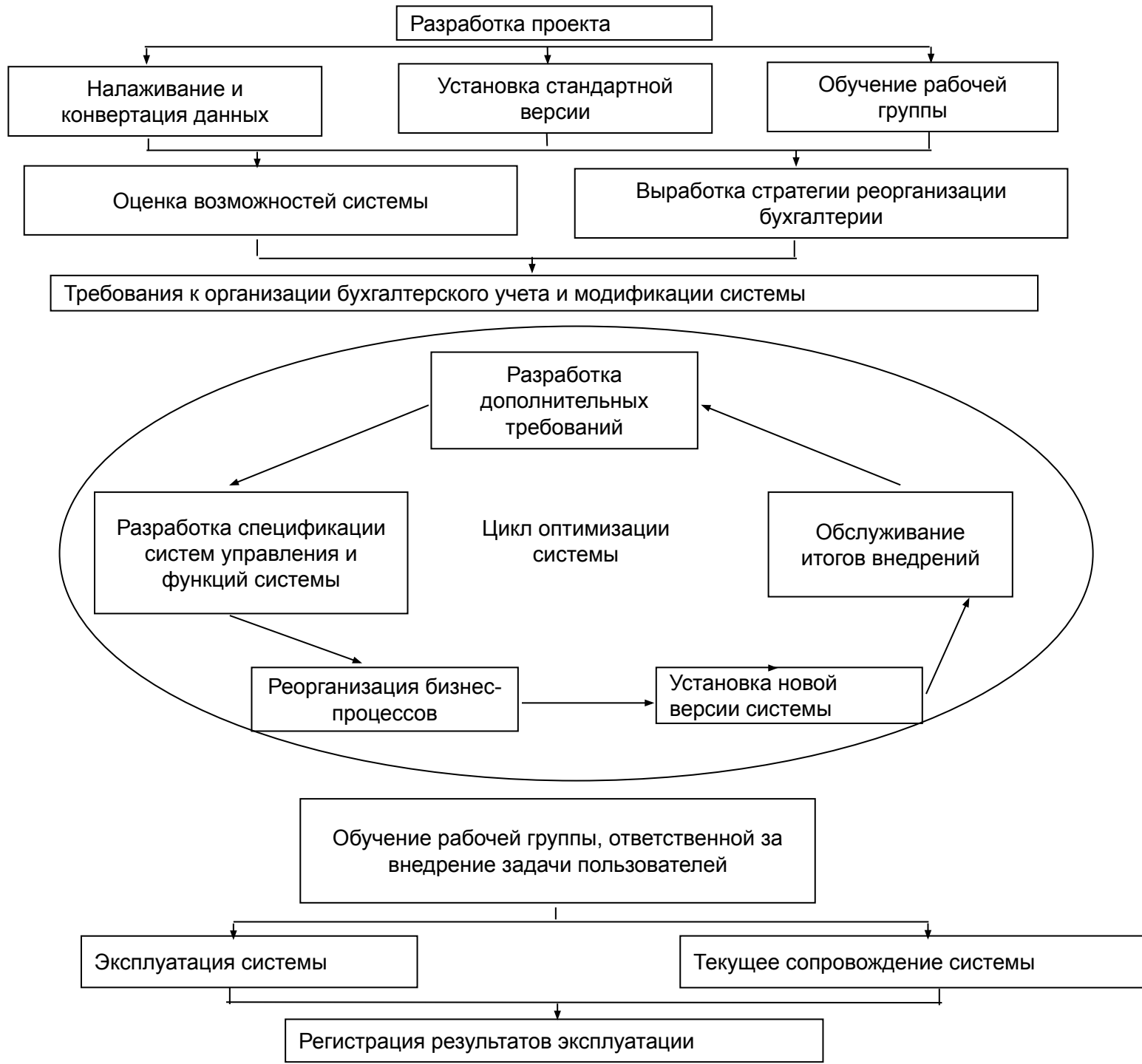


Технологическая документация разрабатывается в соответствии с требованиями ГОСТ 3.11.09 - 82 «Система технологической документации. Термины и определения основных понятий», и составляет содержание технологического обеспечения ИС, которое можно разделить на несколько типов в соответствии с выделением следующих классов задач, решаемых в ИС:

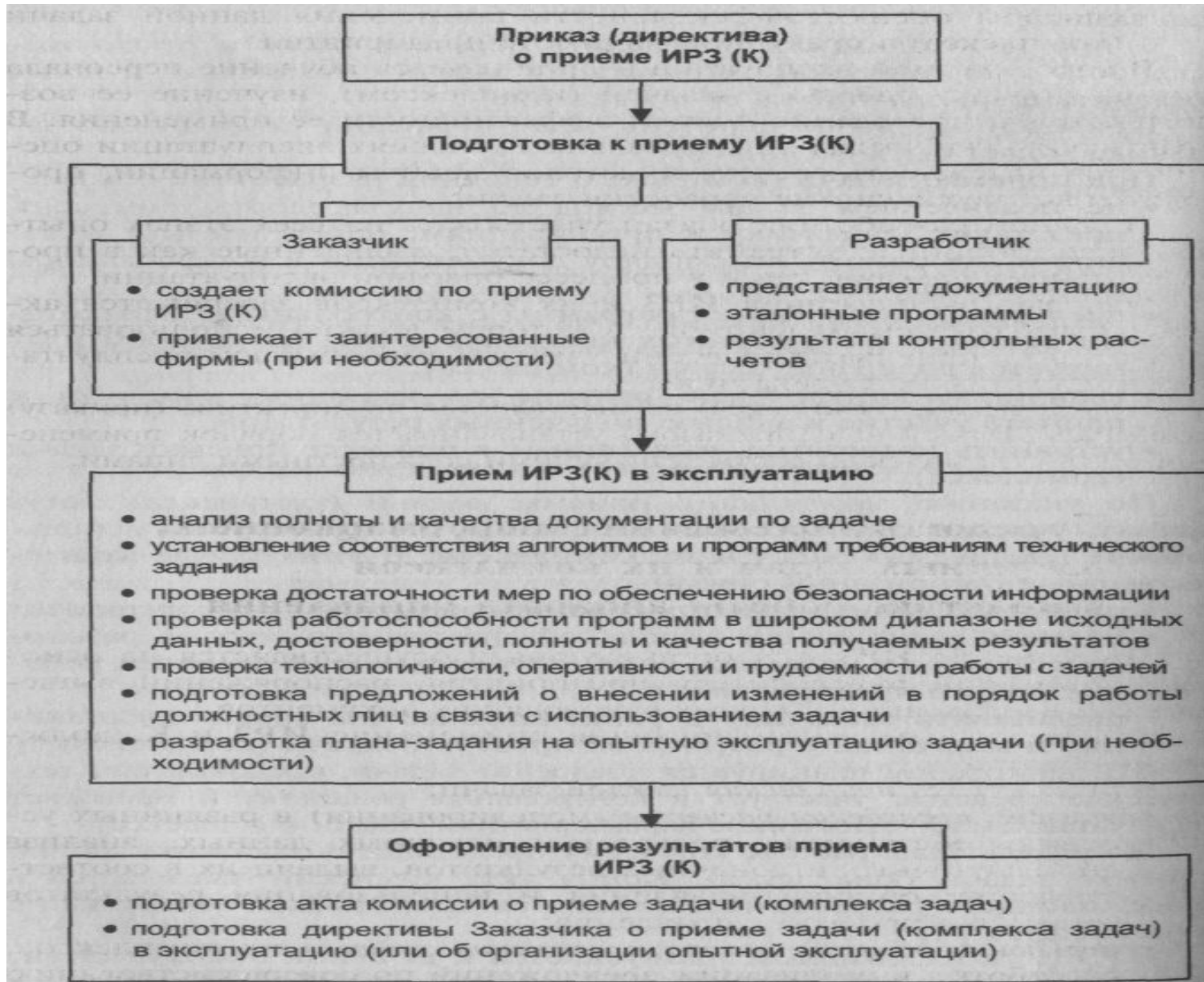
- системы обработки данных (СОД);
- системы поддержки принятия решений (СППР);
- системы автоматизированного проектирования новой продукции (САПР) и т.д.

Последовательность создания технического проекта





Внедрение проекта



На этапе «Сдача проекта в промышленную эксплуатацию» используют

следующую совокупность документов

- договорная документация;
- «Приказ на разработку информационной системы»;
- ТЭО и ТЗ;
- исправленный «Техно - рабочий проект»;
- «Приказ о начале промышленного внедрения»;
- «Программа проведения испытаний»;
- «Требования к научно-техническому уровню проекта системы».

В процессе сдачи проекта в промышленную эксплуатацию осуществляются следующие работы:

- проверка соответствия выполненной работы договорной документации по времени выполнения, объему проделанной работы и затратам денежных средств;
- проверка соответствия проектных решений по информационной системе требованиям технического задания;
- проверка соответствия проектной документации стандартам;
- проверка технологических процессов обработки данных по всем задачам и подсистемам;
- проверка качества функционирования информационной базы, оперативности и полноты ответов на запросы;
- выявление локальных и системных ошибок и их исправление.

Основные понятия и классификация CASE-технологий

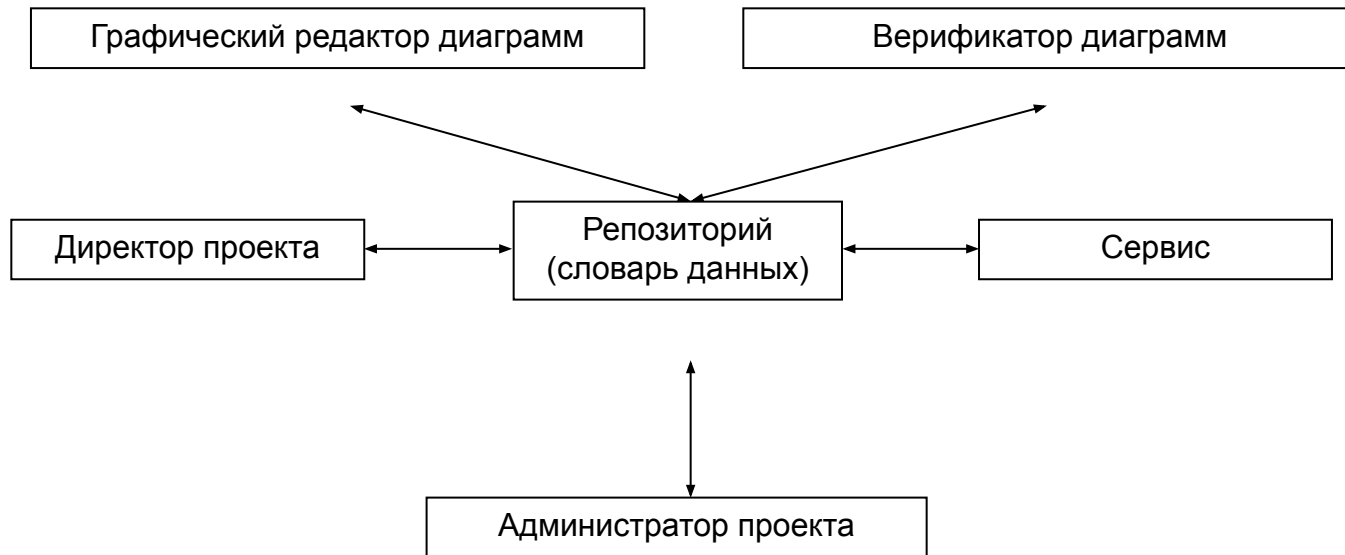
Большинство существующих CASE-систем ориентировано на автоматизацию проектирования программного обеспечения и основано на методологиях структурного (в основном) или объектно-ориентированного проектирования и программирования, использующих спецификации в виде диаграмм или текстов для описания системных требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств. В последнее время стали появляться CASE-системы, уделяющие основное внимание проблемам спецификации и моделирования технических средств.

Преимущества CASE-технологии по сравнению с традиционной технологией оригинального проектирования сводятся к следующему:

- улучшение качества разрабатываемого программного приложения за счет средств автоматического контроля и генерации;
- возможность повторного использования компонентов разработки;
- поддержание адаптивности и сопровождения ЭИС;
- снижение времени создания системы, что позволяет на ранних стадиях проектирования получить прототип будущей системы и оценить его;
- освобождение разработчиков от рутинной работы по документированию проекта, так как при этом используется встроенный документатор;
- возможность коллективной разработки ЭИС в режиме реального времени.

- CASE-технология в рамках методологии включает в себя методы, с помощью которых на основе графической нотации строятся диаграммы, поддерживаемые инструментальной средой.
- Методология определяет шаги и этапность реализации проекта, а также правила использования методов, с помощью которых разрабатывается проект.
- Метод - это процедура или техника генерации описаний компонентов ЭИС (например, проектирование потоков и структур данных).
- Нотация - отображение структуры системы, элементов данных, этапов обработки с помощью специальных графических символов диаграмм, а также описание проекта системы на формальных и естественных языках.
- Инструментальные средства CASE - специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования ИС.

Архитектуру CASE-средства



Определение потребностей в CASE-средствах



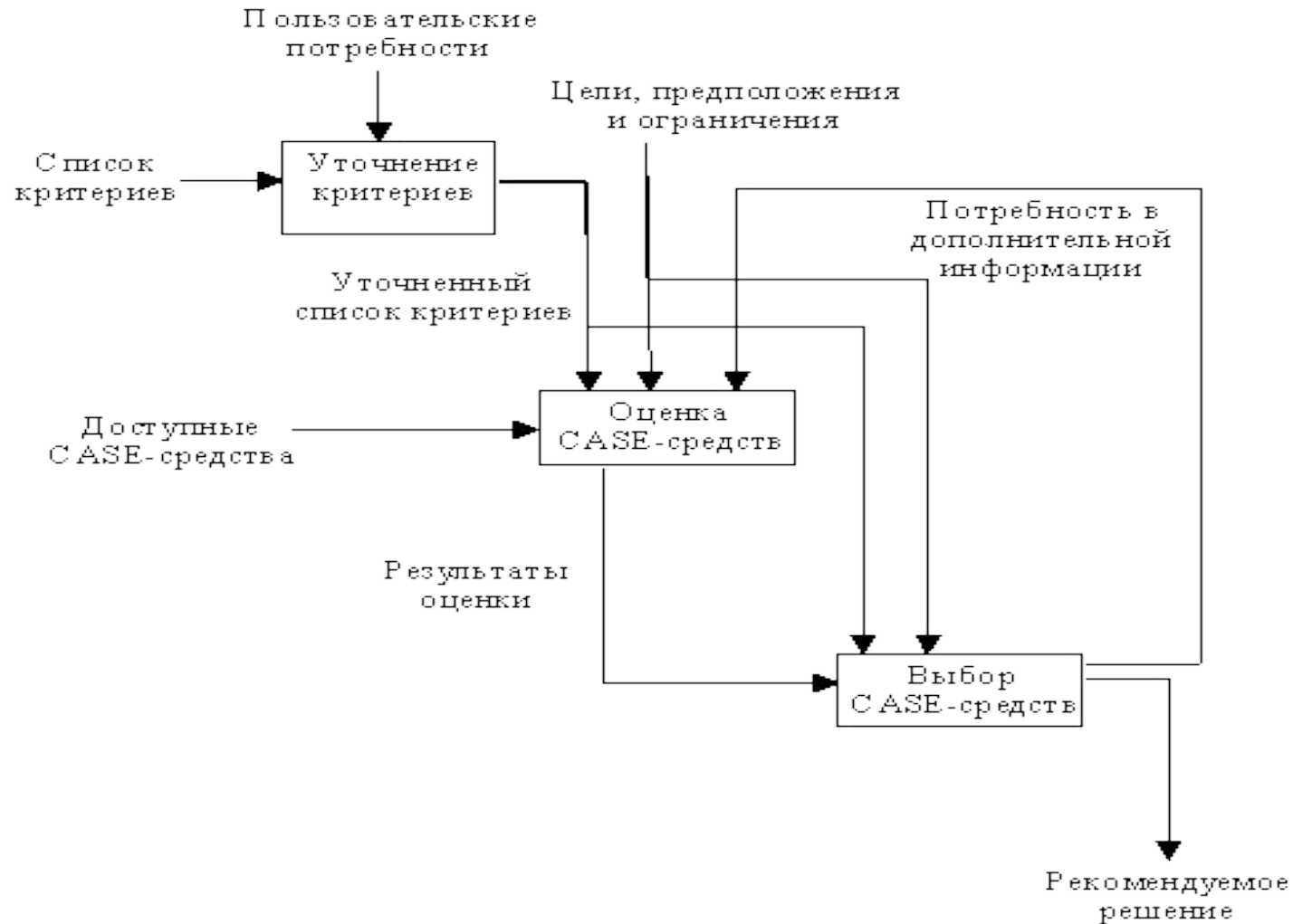
**Процесс внедрения CASE-средств состоит из
следующих этапов:**

- **определение потребностей в CASE-средствах;**
- **оценка и выбор CASE-средств;**
- **выполнение пилотного проекта;**
- **практическое внедрение CASE-средств.**

Процесс оценки и выбора может преследовать несколько целей:

- оценка нескольких CASE-средств и выбор одного или более из них;
- оценка одного или более CASE-средств и сохранение результатов для последующего использования;
- выбор одного или более CASE-средств с использованием результатов предыдущих оценок.

Модель процесса оценки и выбора



Сравнение функциональных возможностей систем

№ / №	Возможности/Инструментальная среда	ARIS Tooleset 5.0	BPWin 4.0
1.	Поддерживаемый стандарт	Частично DFD,URM, UML	IDEF(),IDEF3,DFD
2.	Система хранения данных модели	Объектная база данных	Модели хранятся в файлах
3.	Ограничения на размер базы данных	Нет. Размер базы данных ограничиваются ресурсами	Нет. Размер базы данных ограничиваются ресурсами
4.	Возможность групповой работы	Есть. Используется ARIS Server	Есть. Используется Model Art
5.	Ограничения на количество объектов на диаграмме	Нет	От 2 до 8
6.	Возможность декомпозиции	Неограниченная декомпозиция. Возможна декомпозиция на различные модели	Неограниченная декомпозиция. Возможен однократный переход на другую нотацию в процессе декомпозиции
7.	Формат представления моделей	Не регламентируется	Стандартный бланк IDEF с возможностью его отключения
8.	Удобство работы по созданию моделей	Сложная панель управления, есть выравнивание объектов, есть undo	Простая панель управления, нет выравнивание объектов, нет undo
9.	UDP-свойства –объектов, определяемые пользователем	Большое, но ограниченное количество свойств, количество типов ограничено	Количество UDP не ограничено. Количество типов ограничено
10.	Возможность анализа стоимости процессов	Есть. Возможность использовать ARIS ABC.	Упрощенный анализ стоимости по частоте использования в процессе. Возможность экспорта в Easy ABC.
1	Генерация отчетов	Стандартных отчетов на основе	RPTWin, возможность визуальной

Экспертная оценка применения инструментальных средств моделирования бизнес-процессов

Задача/Инструментальная среда	ARIS Toolset 5.0	BPWin 4.0
<p>Разовый проект по описанию бизнес-процессов, например:</p> <p>1) описание одного бизнес-процесса с точки зрения контроля и управления;</p> <p>2) описание функциональных новой системы управления на верхнем уровне возможностей</p>	<p>3</p> <p>3</p>	<p>5</p> <p>5</p>
<p>Длительный непрерывный проект по описанию деятельности компании с различных точек зрения (организационная структура, структура документов, большой объем базы данных процессов и т.д.)</p>	<p>5</p>	<p>3</p>
<p>Разработка системы автоматизации:</p> <p>1) описание функциональных возможностей системы;</p> <p>2) создание логической системы данных;</p> <p>3) создание физической модели данных.</p>	<p>3</p> <p>3</p> <p>-</p>	<p>5</p> <p>BPWin+ERWin</p> <p>+Paradigm</p>

Основными идеями функционально-ориентированной CASE-технологии являются идеи структурного анализа и проектирования информационных систем. Они заключаются в следующем:

- 1) декомпозиция всей системы на некоторое множество иерархически подчиненных функций;
- 2) представление всей информации в виде графической нотации. Систему всегда легче понять, если она изображена графически.

В качестве инструментальных средств структурного анализа и проектирования выступают следующие диаграммы:

- BFD (Business Function Diagram) - диаграмма бизнес-функций (функциональные спецификации);
- DFD (Data Flow Diagram) - диаграмма потоков данных;
- STD (State Transition Diagram) - диаграмма переходов состояний (матрицы перекрестных ссылок);
- ERD (Entity Relationship Diagram) - ER-модель данных предметной области (информационно-логические модели «сущность-связь»);
- SSD (System Structure Diagram) - диаграмма структуры программного приложения.

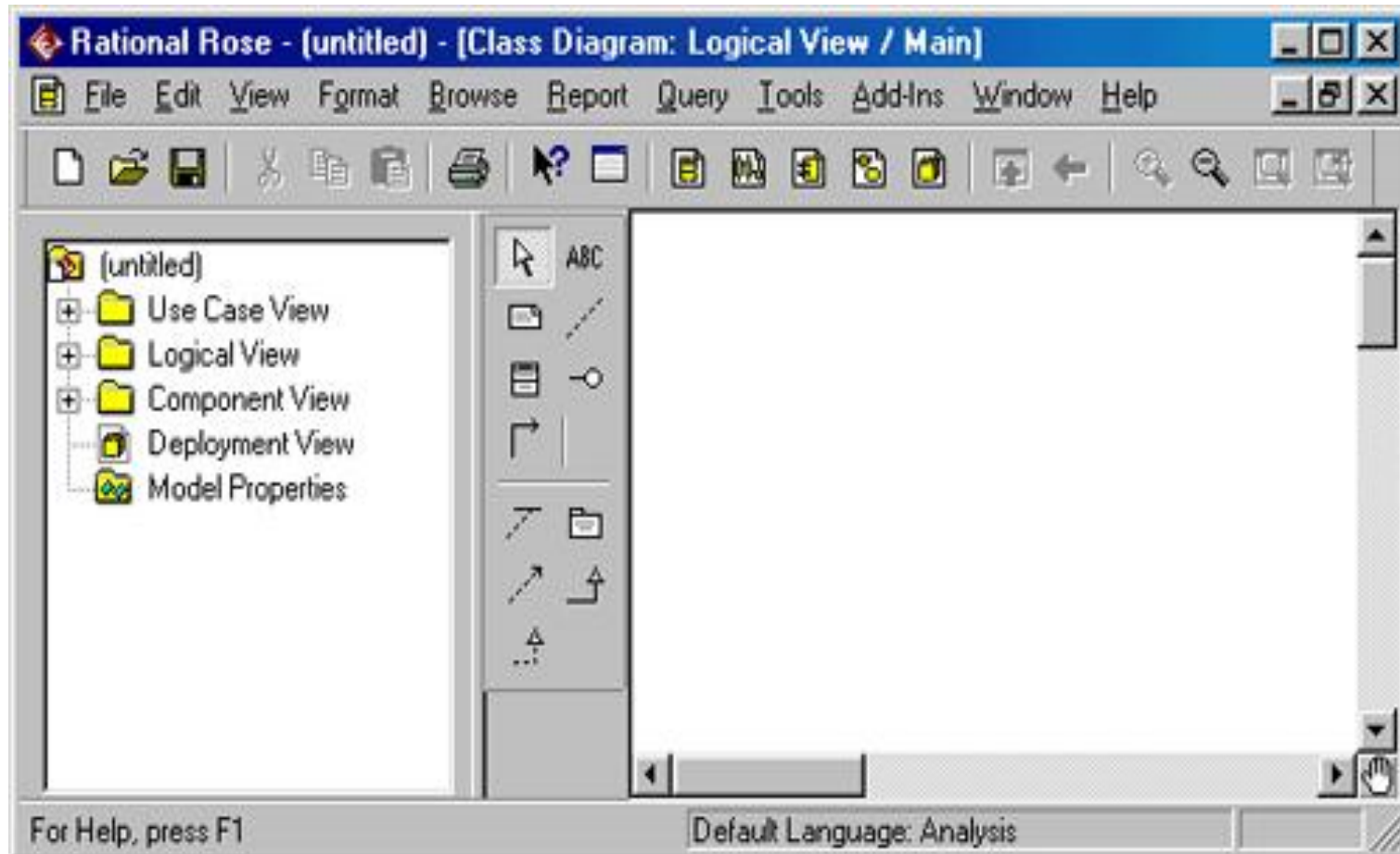
Объектно-ориентированное проектирование информационных систем

Структурная декомпозиция информационной системы на основе объектно-ориентированного подхода отличается от функционально-ориентированного подхода лучшей способностью отражать динамическое поведение системы в зависимости от возникающих событий. Модель проблемной области рассматривается как совокупность взаимодействующих во времени объектов.

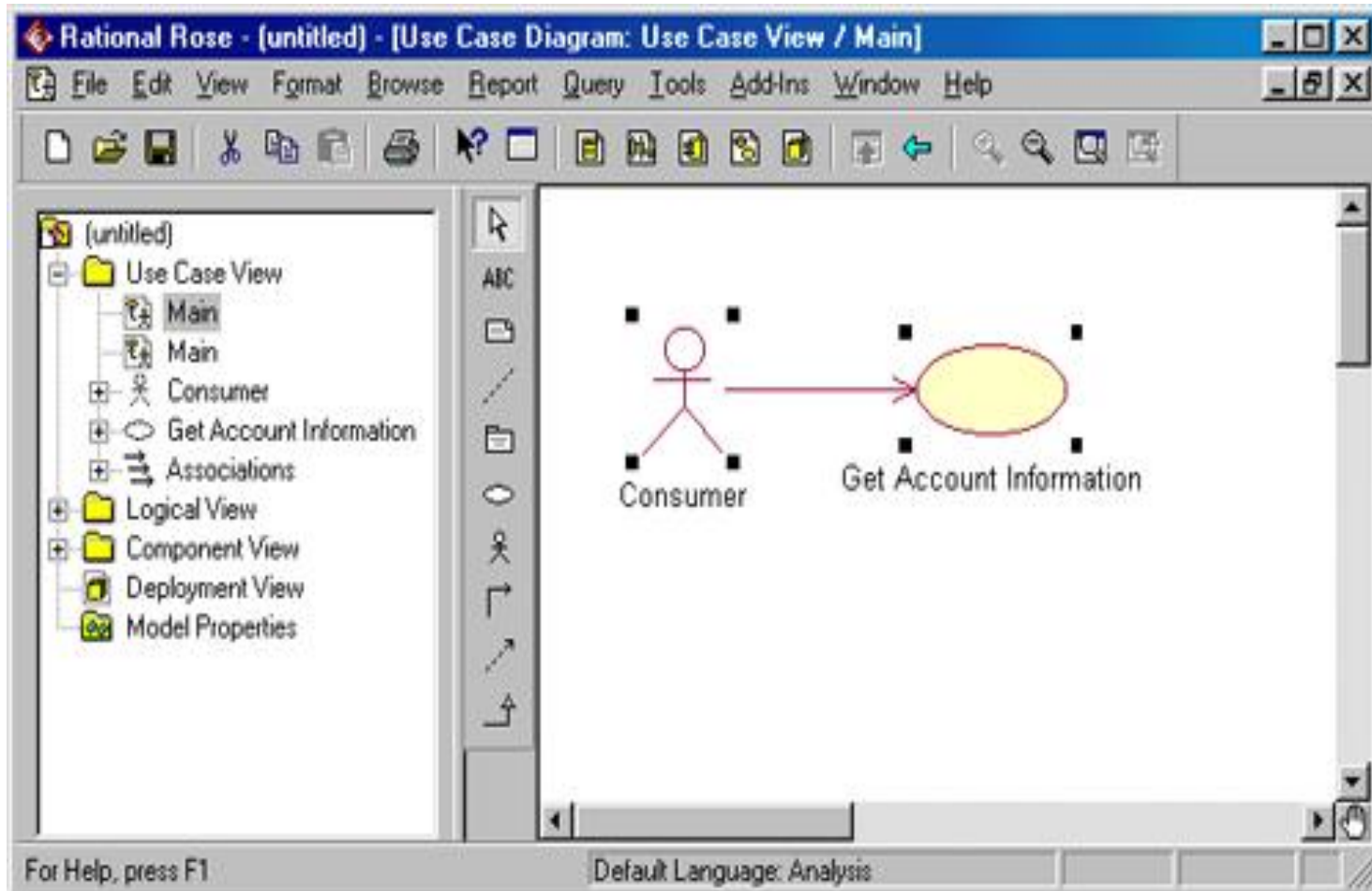
Система объектно-ориентированных моделей в соответствии с нотациями UML включает в себя следующие диаграммы:

- 1) диаграмму прецедентов использования (Use-case diagram), которая отображает функциональность информационной системы в виде совокупности выполняющихся последовательностей транзакций;
- 2) диаграмму классов объектов (Class diagram), которая отображает структуру совокупности взаимосвязанных классов объектов аналогично ER-диаграмме функционально-ориентированного подхода;
- 3) диаграммы состояний (Statechart diagram), каждая из которых отображает динамику состояний объектов одного класса и связанных с ними событий;
- 4) диаграммы взаимодействия объектов (Interaction diagram), каждая из которых отображает динамическое взаимодействие объектов в рамках одного прецедента использования;
- 5) диаграммы деятельности (Activity diagram), которые отображают потоки работ во взаимосвязанных прецедентах использования (могут декомпозироваться на более детальные диаграммы);
- 6) диаграммы пакетов (Package diagram), которые отображают распределение объектов по функциональным или обеспечивающим подсистемам (могут декомпозироваться на более детальные диаграммы);
- 7) диаграмму компонентов (Component diagram), которая отображает физические модули программного кода;
- 8) диаграмму размещения (Deployment diagram), которая отображает распределение объектов по узлам вычислительной сети.

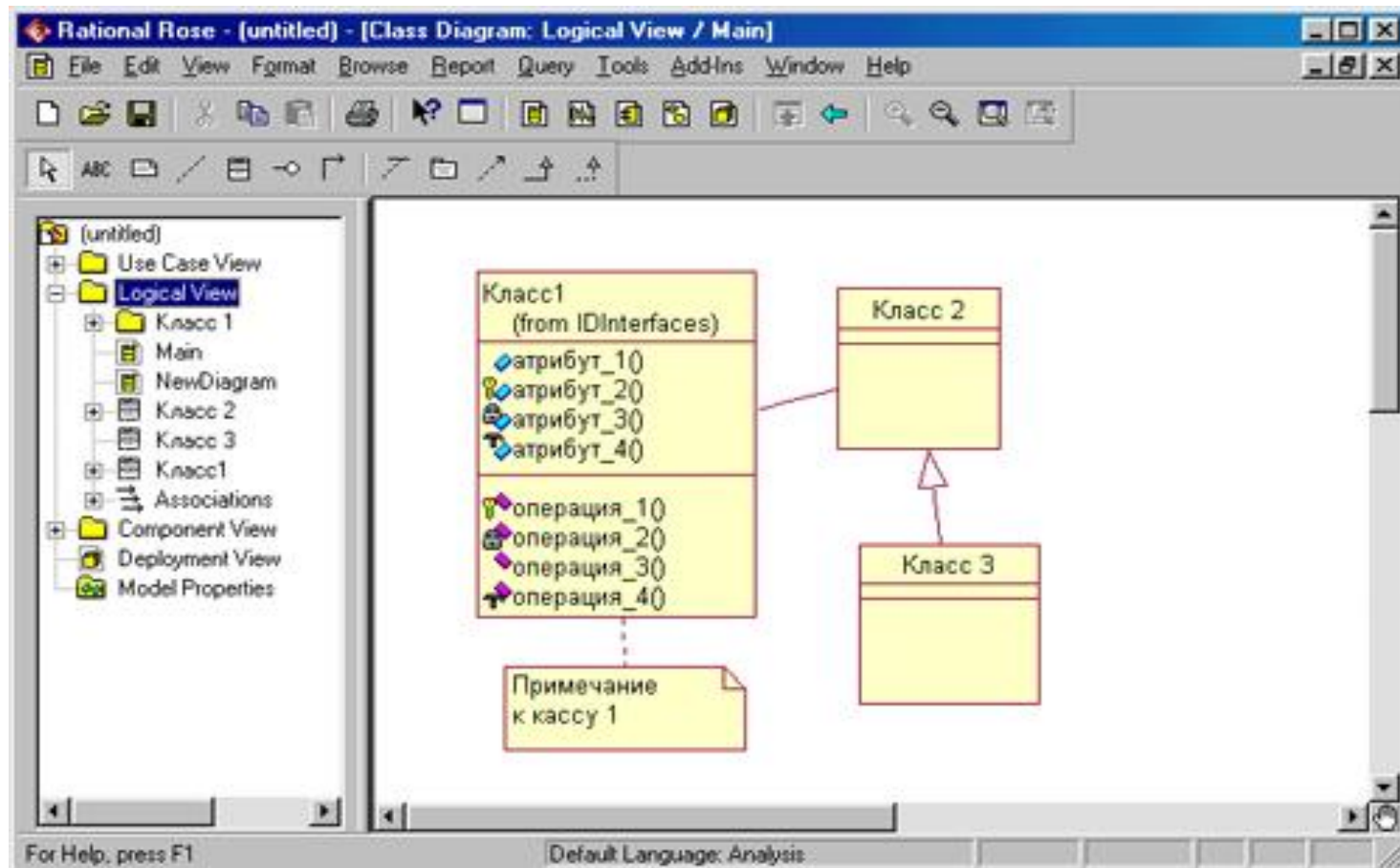
Рабочий интерфейс



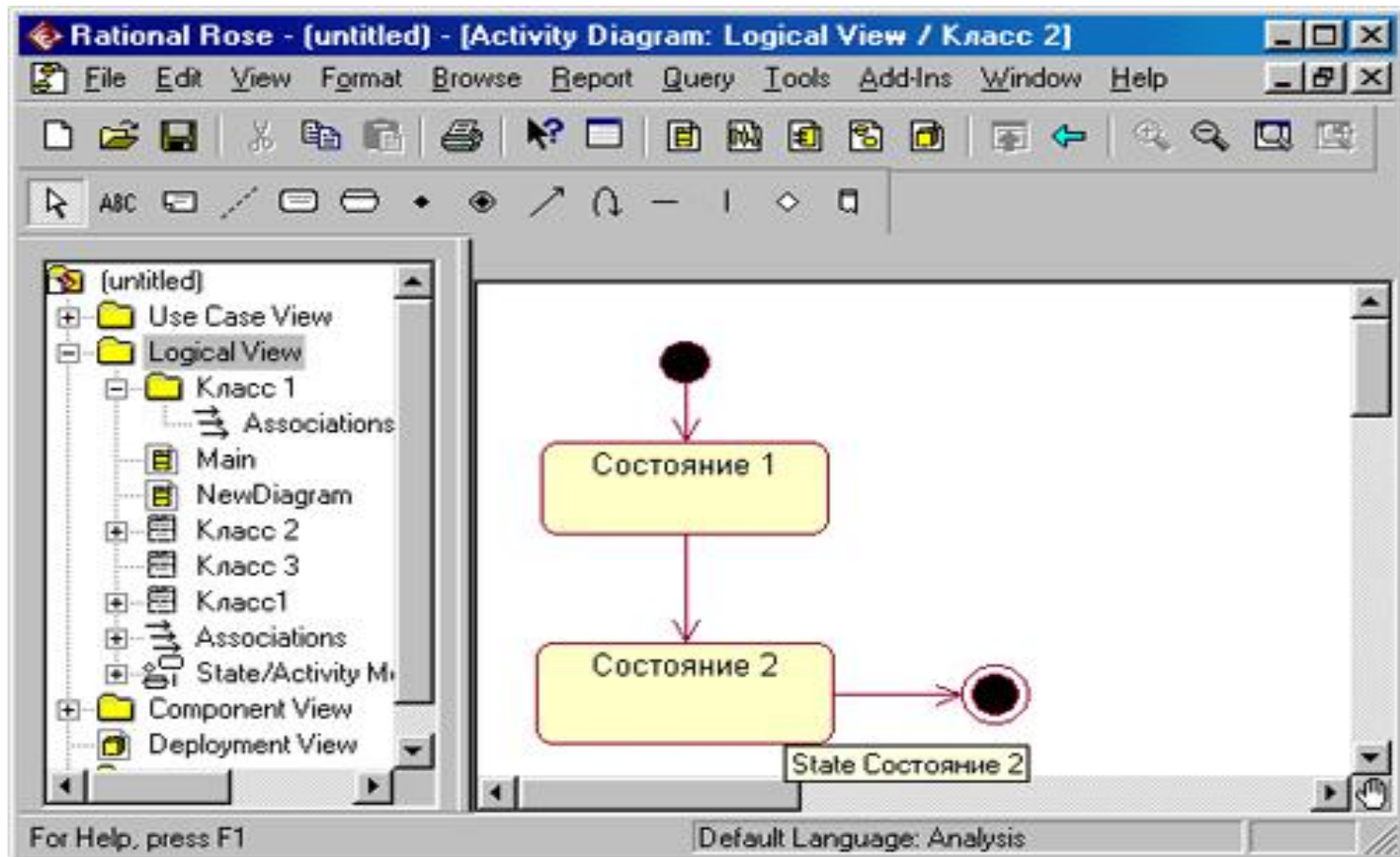
Пример разработки диаграммы вариантов использования в среде Rational Rose



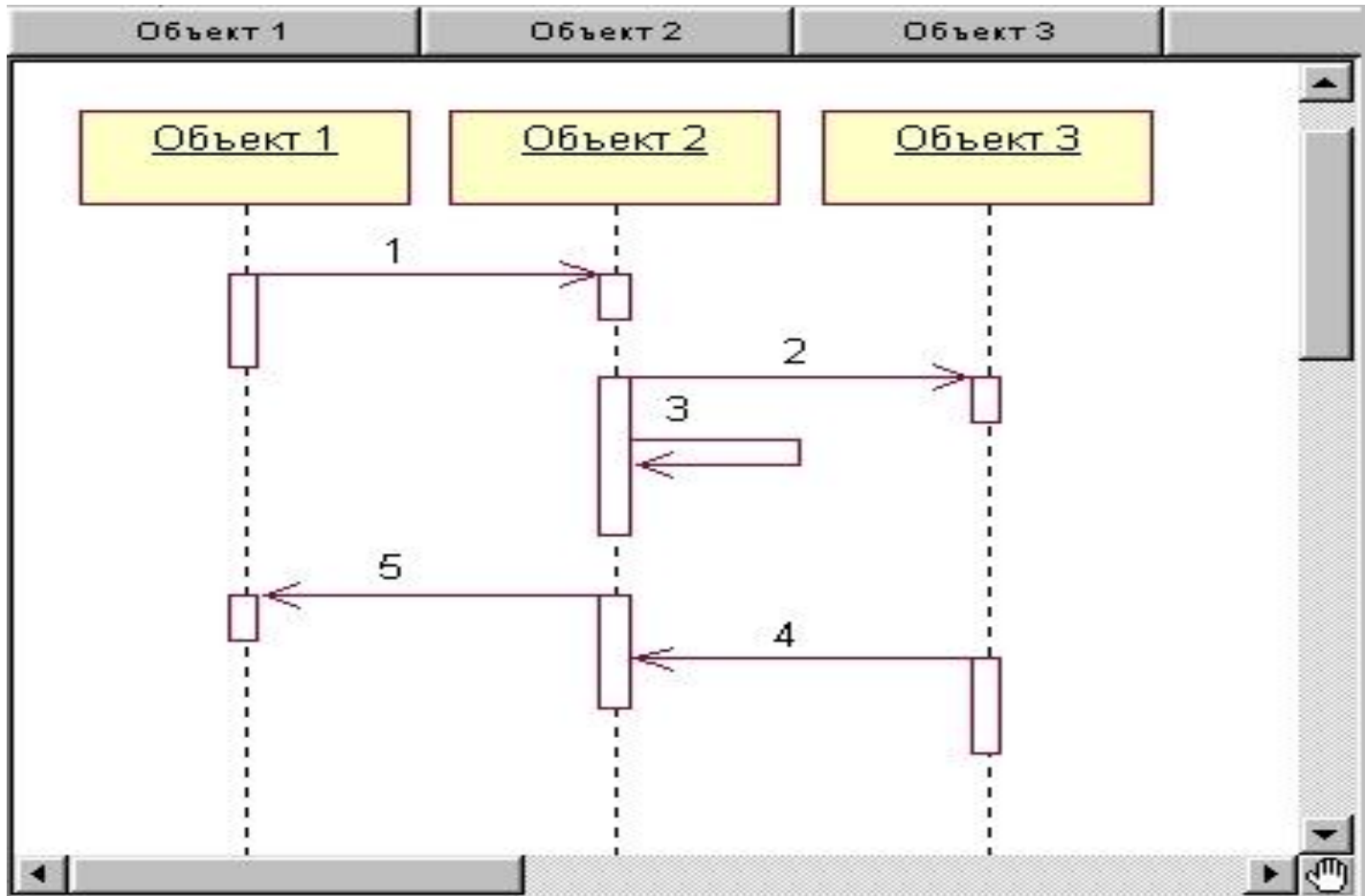
Пример графического изображения диаграммы классов в среде Rational Rose



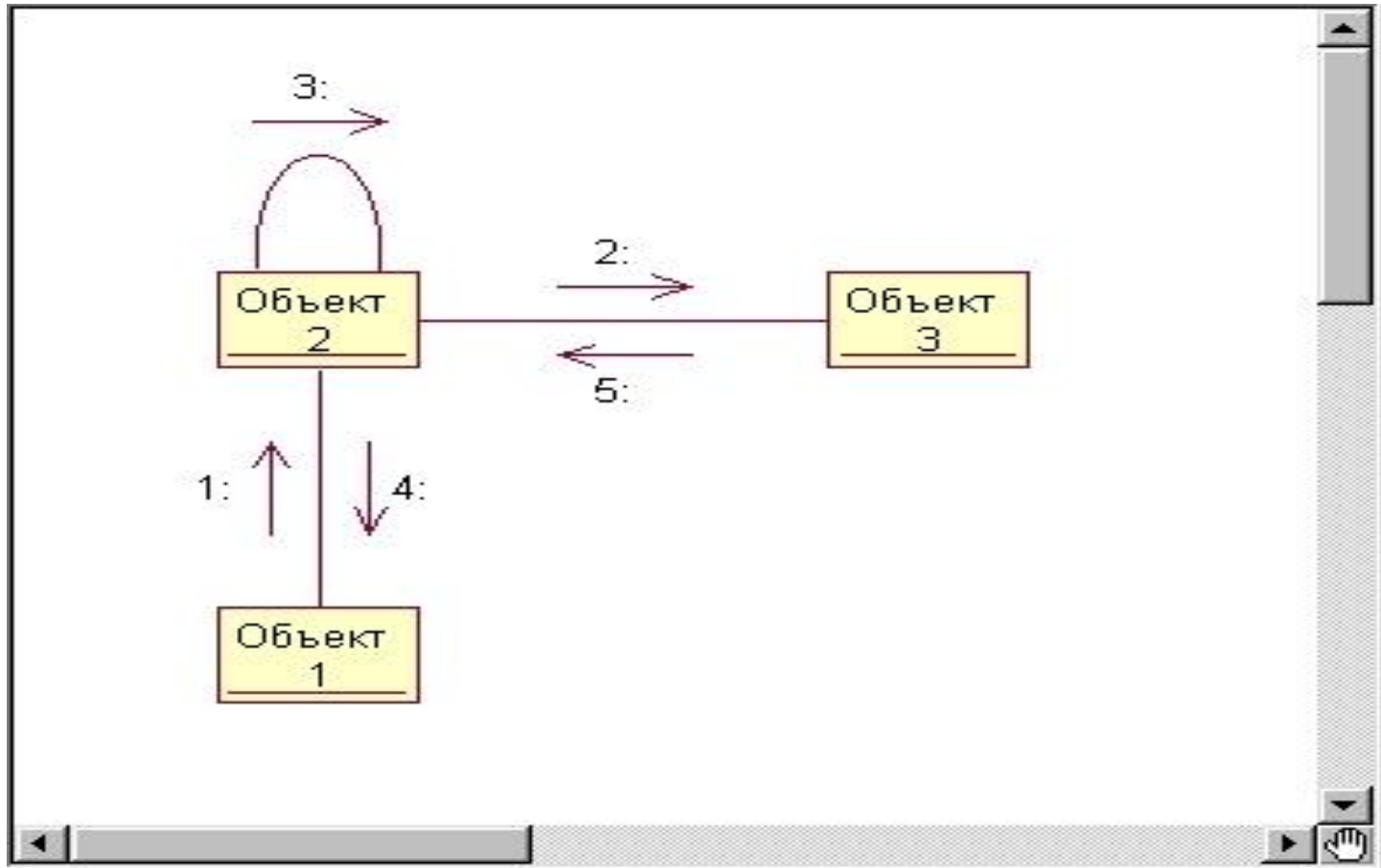
Пример графического изображения диаграммы состояний в среде Rational Rose



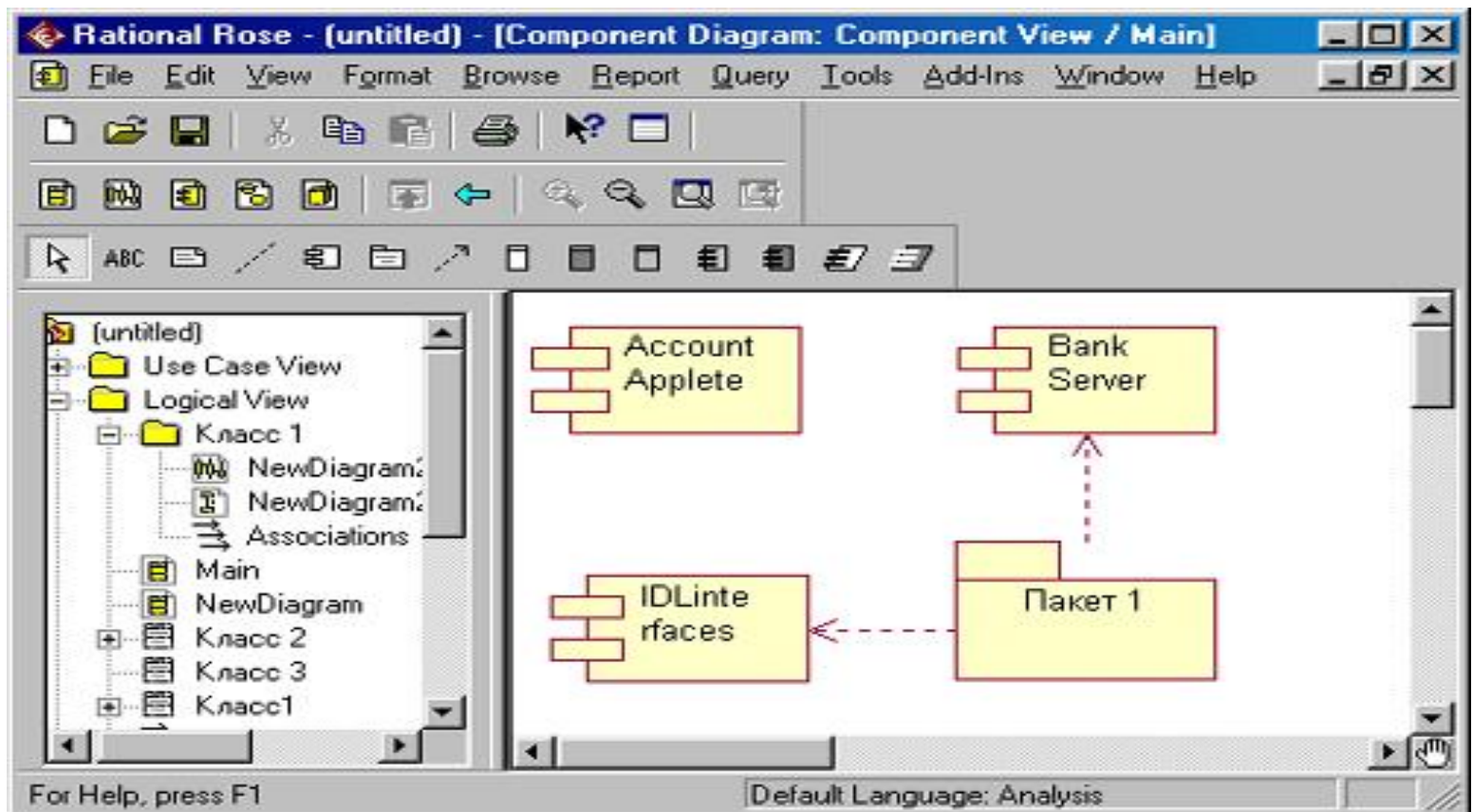
Пример графического изображения диаграммы последовательности в окне диаграммы среды Rational Rose



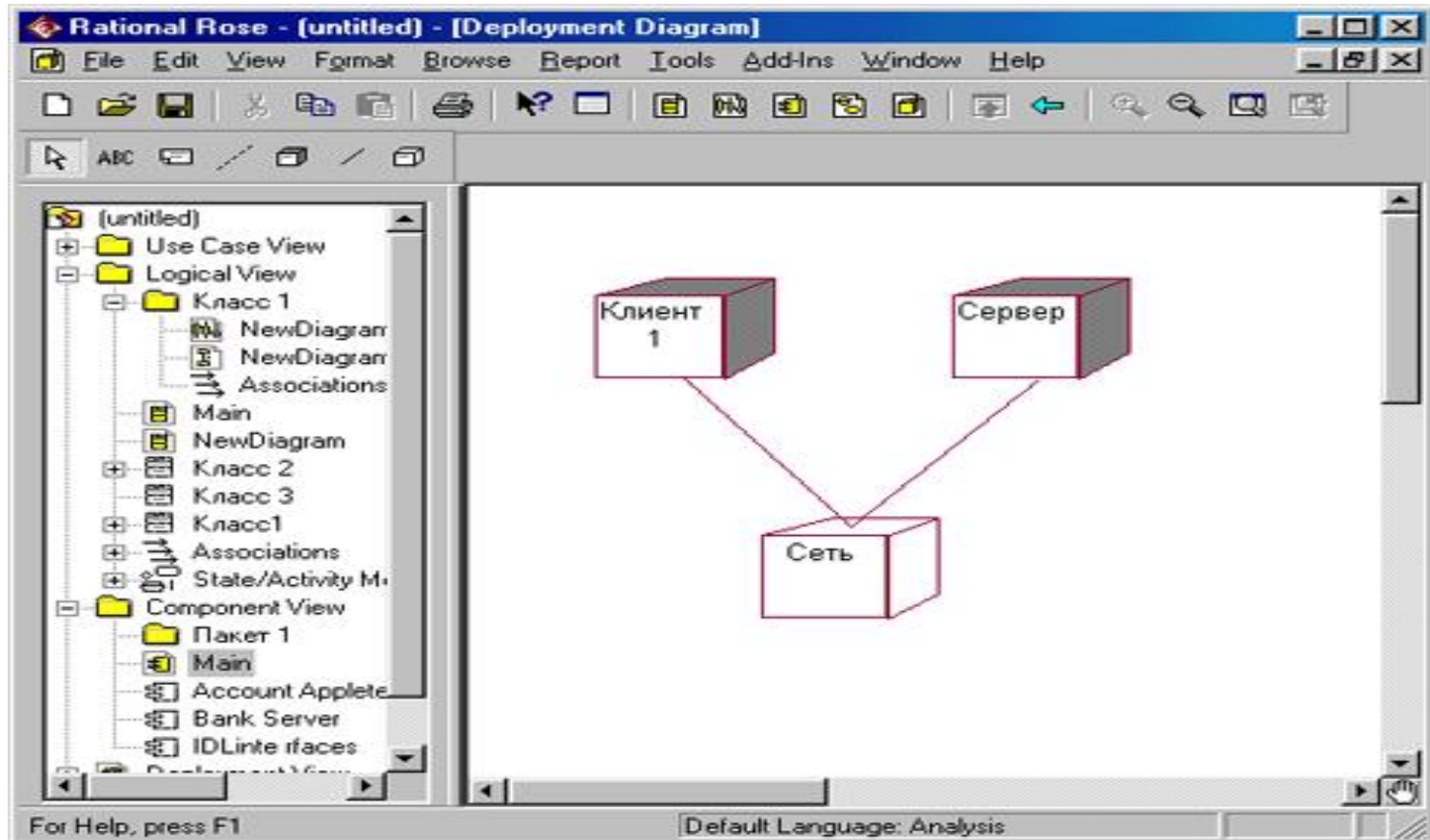
Пример графического изображения диаграммы кооперации



Пример графического изображения диаграммы КОМПОНЕНТОВ



Пример графического изображения диаграммы развертывания в среде Rational Rose



CASE-средстве BPwin

- BPwin - CASE-средство верхнего уровня, помогающее проводить анализ и реорганизацию бизнес-процессов. Поддерживается методология:
 - IDEF0 (функциональная модель)
 - IDEF3 (Work Flow Diagram)
 - DFD (Data Flow Diagram).

- Функциональная модель предназначена для описания существующих бизнес-процессов на предприятии (так называемая модель AS-IS) и идеального положения вещей – того, к чему надо стремиться (модель TO-BE).

Процесс построения информационной модели в VRwin состоит из следующих шагов

- построение контекстной диаграммы
- проводится функциональная декомпозиция
- после каждого сеанса декомпозиции проводится сеанс экспертизы

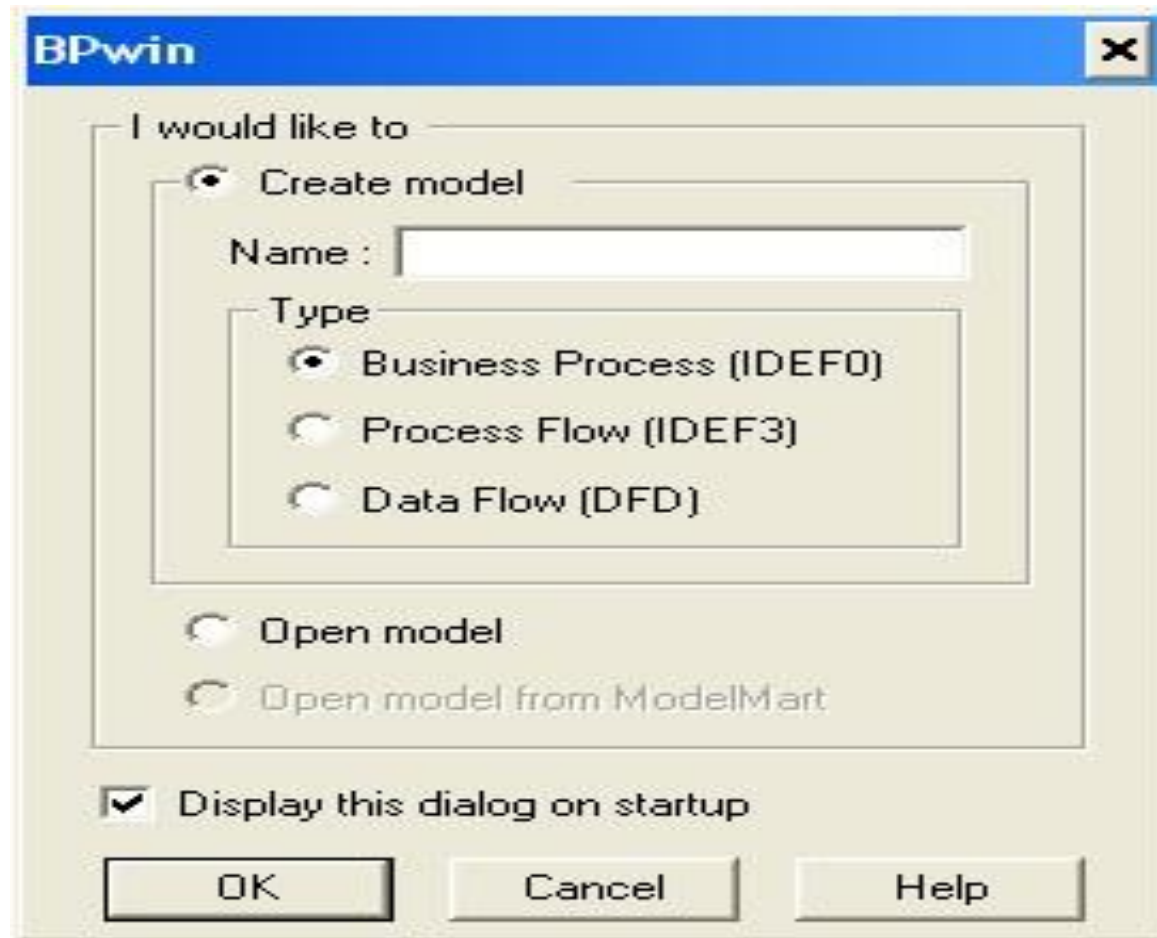
Инструментальная среда VRwin



Панель инструментов представлена следующими кнопками (слева направо):

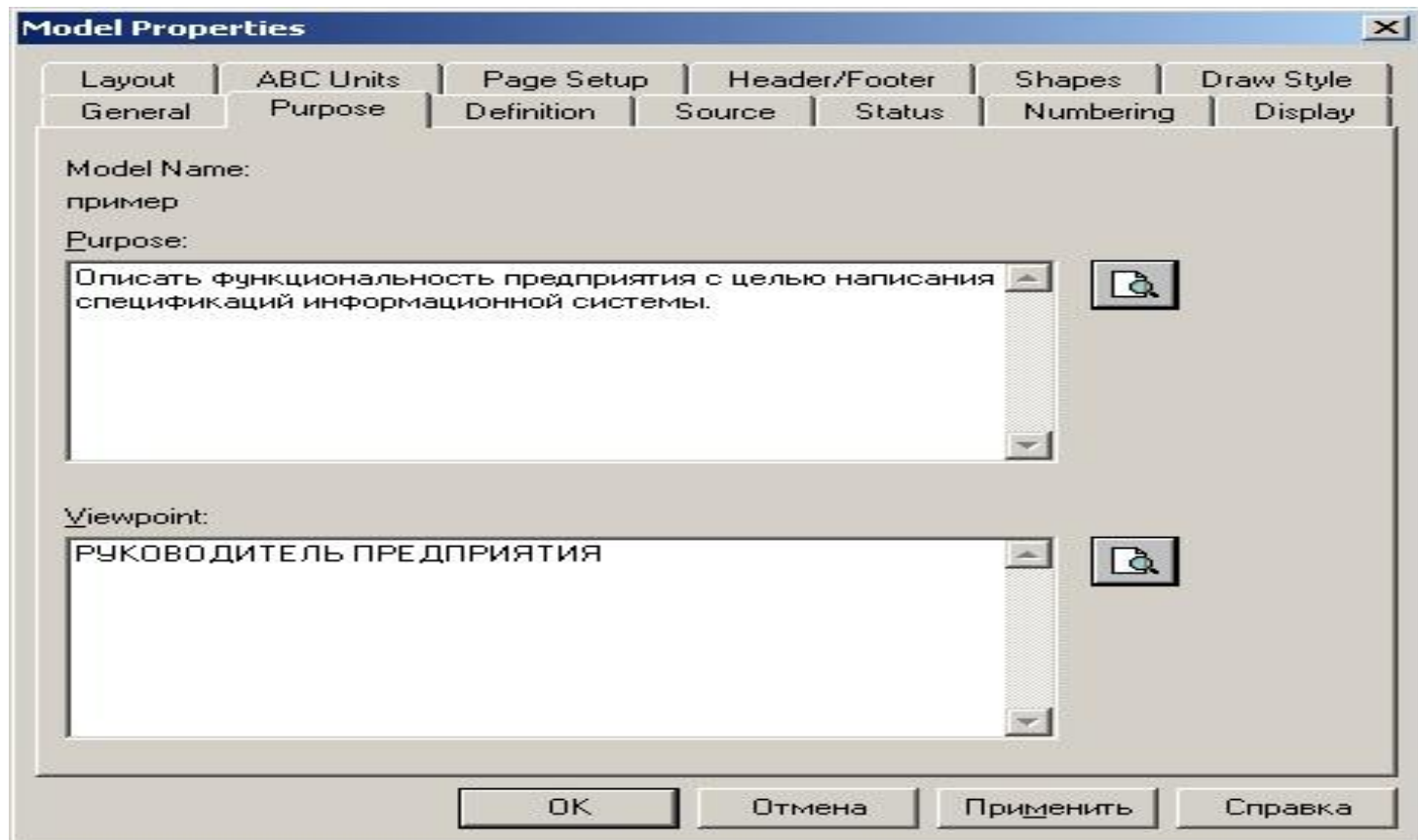
- создать модель (пункт меню File/New);
- открыть модель (пункт меню File/Open);
- сохранить модель (пункт меню File/Save);
- напечатать модель (пункт меню File/Print);
- выбор масштаба (View/Zoom);
- уменьшить модель (View/Zoom);
- увеличить модель (View/Zoom);
- проверить правописание (Tools/Spelling);
- включение и выключение навигатора модели (View/Model Explorer);
- включение и выключение дополнительной панели инструментов работы с Model Mart (Model Mart).

Диалог создания модели



Построение модели IDEF0. Контекстная диаграмма

- Для определения контекста модели в VPwin следует выбрать пункт меню Model/Model Properties
- В закладке General указывается наименование и сведения об авторе модели, в закладку Purpose следует внести цель и точку зрения, а в закладку Definition – определение модели и описание области

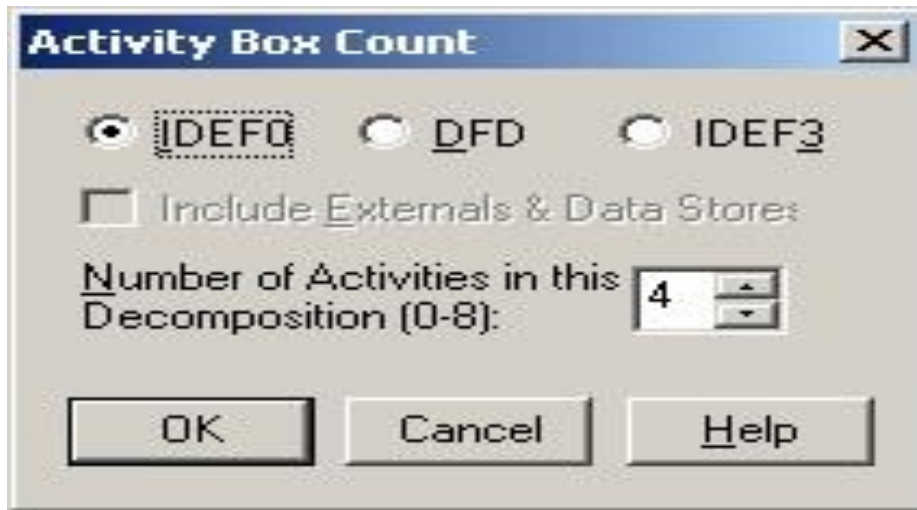


В диаграммах IDEF0 имеется четыре основных типа стрелок

- Стрелки входа всегда направляются в левую сторону блока. Стрелки входа необязательны, так как не все действия могут преобразовать или изменять (заменять) что-либо
- Управление всегда входит в вершину блока. Управление, как правило, представляется в виде правил, инструкций, политики компании, процедур или стандартов. Оно влияет на деятельность без фактического преобразования чего-либо. Управление может также использоваться для описания процедуры начала или окончания выполнения действия
- Стрелки выхода (выпуска) — это материал или информация, произведенная блоком. Каждый блок должен иметь по крайней мере одну стрелку выхода (выпуска). Процессы, которые не производят продукции (выпуска), лучше не моделировать вообще.

Декомпозиция

- Декомпозиционное разложение модели используется в моделировании бизнес-процессов, для того чтобы дать более подробное описание блоков. При каждой декомпозиции блока создается новая диаграмма.
- В диалоге Activity Box Count, в котором следует указать нотацию новой диаграммы и количество блоков на ней. Для IDEF0 рекомендуется 3-6 блоков



- Имя блока и другие его свойства вводятся в закладке «Name» списка свойств блока. Для вывода свойств блока на экран достаточно дважды щелкнуть мышью на блоке.
- Следующим шагом при создании диаграммы должно быть соединение всех использованных на диаграмме блоков с помощью стрелок, представляющих входы, результаты работы, средства управления и механизмы. Для этого достаточно соединить исходящую точку стрелки с точкой ее окончания. Окончанием стрелки может быть как одна из сторон функциональных блоков, так и граница диаграммы.
- Для рисования стрелки пользуются инструментом из комплекта инструментов. Задание имени стрелки производится в закладке «Name» диалога свойств стрелок. Для вызова этого диалога достаточно дважды щелкнуть мышью на нужной стрелке.
- Если действие не было декомпозировано, в верхнем левом углу блока будет появляться символ «листа». После декомпозиции данного блока символ «листа» исчезнет.

Нумерация блоков производится автоматически при их создании. Номера могут быть относительными или постоянными, они отражают иерархическое положение блока в пределах модели

Языки программирования

- На данный момент доминируют объектно-ориентированные языки программирования. Объектно-ориентированное программирование (далее – ООП) включает в себя лучшие принципы структурного программирования с новыми мощными концепциями, базовые из которых называются инкапсуляцией, полиморфизмом и наследованием.

В таблице приведены данные популярности языков (из более чем 8500, существующих сегодня).

Apr 2010	Position Apr 2009	Programming Language	Ratings Apr 2010	Delta Apr 2009	
1	2	C	18.058%	+2.59%	
2	1	Java	18.051%	-1.29%	
3	3	C++	9.707%	-1.03%	
4	4	PHP	9.662%	-0.23%	
5	5	(Visual) Basic	6.392%	-2.70%	
6	7	C#	4.435%	+0.38%	
7	6	Python	4.205%	-1.88%	
8	9	Perl	3.553%	+0.09%	
9	11	Delphi	2.715%	+0.44%	
10	8	JavaScript	2.469%	-1.21%	
11	42	Objective-C	2.288%	+2.15%	
12	10	Ruby	2.221%	-0.35%	
13	14	SAS	0.717%	0.07%	

СИ

Язык программирования Си отличается минимализмом. Авторы языка хотели, чтобы программы на нем легко компилировались с помощью однопроходного компилятора, после компиляции каждой элементарной составляющей программы соответствовало весьма небольшое число машинных команд, а использование базовых элементов языка не задействовало библиотеку времени выполнения. Код на Си можно легко писать на низком уровне абстракции, почти как на ассемблере.

- Компиляторы разрабатываются сравнительно легко благодаря относительно низкому уровню языка и скромному набору элементов. Поэтому данный язык доступен на самых различных платформах (возможно, круг этих платформ шире, чем у любого другого существующего языка).
- К тому же, несмотря на свою низкоуровневую природу, язык позволяет создавать переносимые программы и поддерживает программиста в этом. Программы, соответствующие стандарту языка, могут компилироваться на самых различных компьютерах. Одним из последствий высокой эффективности и переносимости Си стало то, что многие компиляторы, интерпретаторы и библиотеки других языков высокого уровня часто выполнены на языке Си.

Язык программирования Java

Java – объектно ориентированный язык программирования, разрабатываемый компанией Sun Microsystems. Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор, но с тем отличием, что байтовый код, в отличие от текста, обрабатывается значительно быстрее. Достоинство подобного способа выполнения программ в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина.

Особенность Java

Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание. К недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java.

Ряд усовершенствований Java:

- применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде;
 - широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках;
 - аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами фирмы ARM).

Язык программирования Python

Python — высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода. Название языка произошло в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона». Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Основные архитектурные черты

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Питоне организовывается в функции и классы, которые могут объединяться в модули (которые в свою очередь могут быть объединены в пакеты).

Особенности языка

Python — активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют ANSI, ISO или другие официальные стандарты, их роль выполняет CPython. Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется свободно под очень либеральной лицензией, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные.

- Появившись сравнительно поздно, Python создавался под влиянием множества языков программирования: ABC, Modula-3, C, C++, Smalltalk, Lisp, Fortran, Miranda, Java, Icon. Большая часть других возможностей Python (например, байт-компиляция исходного кода) также была реализована ранее в других языках.
- Python портируем и работает почти на всех известных платформах — от КПК до мейнфреймов. Существуют порты под Microsoft Windows, практически все варианты UNIX (включая FreeBSD и Linux), Plan 9, Mac OS и Mac OS X, iPhone OS 2.0 и выше, Palm OS, OS/2, Amiga, AS/400 и даже OS/390, Symbian и Android.
- При этом, в отличие от многих портируемых систем, для всех основных платформ Python имеет поддержку характерных для данной платформы технологий (например, Microsoft COM/DCOM). Более того, существует специальная версия Питона для виртуальной машины Java — Jython, что позволяет интерпретатору выполняться на любой системе, поддерживающей Java, при этом классы Java могут непосредственно использоваться из Питона и даже быть написанными на Питоне. Также несколько проектов обеспечивают интеграцию с платформой Microsoft .NET, основные из которых — IronPython и Python.Net.

Состав и содержание технического задания (ГОСТ 34.602-89)

№ п/п	Раздел	Содержание
1	Общие сведения	<ul style="list-style-type: none"> • полное наименование системы и ее условное обозначение • шифр темы или шифр (номер) договора • наименование предприятий разработчика и заказчика системы, их реквизиты • перечень документов, на основании которых создается ИС • плановые сроки начала и окончания работ • сведения об источниках и порядке финансирования работ • порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств
2	Назначение и цели создания (развития) системы	<ul style="list-style-type: none"> • вид автоматизируемой деятельности • перечень объектов, на которых предполагается использование системы • наименования и требуемые значения технических, технологических, производственно-экономических и др. показателей объекта, которые должны быть достигнуты при внедрении ИС
3	Характеристика объектов автоматизации	<ul style="list-style-type: none"> • краткие сведения об объекте автоматизации • сведения об условиях эксплуатации и характеристиках окружающей среды

№ п/п	Раздел	Содержание
4	Требования к системе	<p>Требования к системе в целом:</p> <ul style="list-style-type: none"> • требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы) • требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки) • показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров) • требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации <p>Требования к функциям (по подсистемам):</p> <ul style="list-style-type: none"> • перечень подлежащих автоматизации задач • временной регламент реализации каждой функции • требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов • перечень и критерии отказов <p>Требования к видам обеспечения:</p> <ul style="list-style-type: none"> • математическому (состав и область применения математических моделей и методов, типовых и разрабатываемых алгоритмов) • информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным -

№ п/п	Раздел	Содержание
4		<ul style="list-style-type: none"> • лингвистическому (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода-вывода); • программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ); • техническому; • метрологическому; • организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала); • методическому (состав нормативно-технической документации).
5	Состав и содержание работ по созданию системы	<ul style="list-style-type: none"> • перечень стадий и описание работ; • сроки исполнения; • состав организаций-исполнителей работ; • вид и порядок экспертизы технической документации; • программа и обеспечение надежности; • программа метрологического обследования.
6	Порядок контроля и приемки системы	<ul style="list-style-type: none"> • виды, состав, объем и методы испытаний системы; • общие требования к приемке работ по стадиям; • статус приемной комиссии.
7	Требования к составу и содержанию работ по подготовке системы в действие	<ul style="list-style-type: none"> • перечень подлежащих разработке документов; • перечень документов на машинных носителях.
8	Источники разработки	Документы и информационные материалы, на основании которых разрабатывается ТЗ и система.

на этапе эскизного проектирования определяются:

- функции ИС;
- функции подсистем, их цели и ожидаемый эффект от внедрения;
- состав комплексов задач и отдельных задач;
- концепция информационной базы и ее укрупненная структура;
- функции системы управления базой данных;
- состав вычислительной системы и других технических средств;
- функции и параметры основных программных средств.

Содержание технического проекта

№ п/п	Раздел	Содержание
1	Пояснительная записка	<ul style="list-style-type: none">• основания для разработки системы• перечень организаций разработчиков• краткая характеристика объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами• краткие сведения об основных объектных решениях по функциональной и обеспечивающим частям системы
2	Функциональная и организационная структура системы	<ul style="list-style-type: none">• обоснование выделяемых подсистем, их перечень и назначение• перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания• схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы

№ п/п*	Раздел*	Содержание*
4	4	<ul style="list-style-type: none"> • перечень массивов НСИ, их объем, порядок и чистота корректировки информации • структура фонда НСИ с описанием связи между его элементами; требования к технологии создания и ведения фонда • методы хранения, поиска, внесения изменений и контроля • определение объемов и потоков информации НСИ • контрольный пример по внесению изменений в НСИ • предложения по унификации информации
5	Альбом форм документов*	4
6	Система математического обеспечения*	<ul style="list-style-type: none"> • обоснование структуры математического обеспечения • обоснование выбора системы программирования • перечень стандартных программ
7	Принцип построения комплекса технических средств*	<ul style="list-style-type: none"> • описание и обоснование схемы технического процесса обработки данных • обоснование и выбор структуры комплекса технических средств и его функциональных групп • обоснование требований к разработке нестандартного оборудования • комплекс мероприятий по обеспечению надежности функционирования технических средств
8	Расчет экономической эффективности системы*	<ul style="list-style-type: none"> • сводная смета затрат, связанных с эксплуатацией системы • расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства, снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений

№ п/п	Раздел	Содержание
9	Мероприятия по подготовке проекта к внедрению системы	<ul style="list-style-type: none"> • перечень организационных мероприятий по совершенствованию бизнес-процессов • перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц
10	Ведомость документов	•

Достоинства и недостатки ТПР

Класс ТПР Реализация ТПР	Достоинства	Недостатки
Элементные ТПР. Библиотеки методо- ориентирован- ных программ	<ul style="list-style-type: none"> • обеспечивается при- менение модульного под- хода к проектированию и документированию ИС 	<ul style="list-style-type: none"> • большие затраты времени на сопряжение разнород- ных элементов вследствие информационной, про- граммной и технической несовместимости • большие затраты времени на доработку ТПР отдель- ных элементов
Подсистемные ТПР Пакеты приклад- ных программ	<ul style="list-style-type: none"> • достигается высокая степень интеграции элементов ИС • позволяют осущест- влять: модульное про- ектирование; параме- трическую настройку программных компо- нентов на различные объекты управления 	<ul style="list-style-type: none"> • адаптивность ТПР не- достаточна с позиции не- прерывного инжиниринга деловых процессов

Класс ТПР Реализация ТПР	Достоинства	Недостатки
•	<ul style="list-style-type: none"> • обеспечивают: сокращение затрат на проектирование и программирование взаимосвязанных компонентов; хорошее документирование отображаемых процессов обработки информации 	<ul style="list-style-type: none"> • возникают проблемы в комплексировании различных функциональных подсистем, особенно в случае использования решений нескольких производителей программного обеспечения
Объектные ТПР. Отраслевые проекты ИС	<ul style="list-style-type: none"> • комплексирование всех компонентов ИС за счет методологического единства и информационной, программной и технической совместимости • открытость архитектуры – позволяет устанавливать ТПР на разных программно-технических платформах • масштабируемость – допускает конфигурацию ИС для переменного числа рабочих мест • конфигурируемость – позволяет выбирать необходимое подмножество компонентов 	<ul style="list-style-type: none"> • проблемы привязки типового проекта к конкретному объекту управления, что вызывает в некоторых случаях даже необходимость изменения организационно-экономической структуры объекта автоматизации

Реализация типового проекта предусматривает выполнение следующих операций

- установку глобальных параметров системы
- задание структуры объекта автоматизации
- определение структуры основных данных
- задание перечня реализуемых функций и процессов
- описание интерфейсов
- описание отчетов;
- настройку авторизации доступа;
- настройку системы архивирования.

- *Профиль* - это совокупность нескольких (или подмножество одного) базовых стандартов с четко определенными и гармонизированными подмножествами обязательных и факультативных возможностей, предназначенная для реализации заданной функции или группы функций.

Использование профилей информационных систем призвано решить следующие задачи

- снижение трудоемкости проектов;
- повышение качества компонентов информационной системы;
- обеспечение расширяемости и масштабируемости разрабатываемых систем;
- обеспечение возможности функциональной интеграции в информационную систему задач, которые раньше решались раздельно;
- обеспечение переносимости прикладного программного обеспечения. В зависимости от того, какие из указанных задач являются наиболее приоритетными, производится выбор стандартов и документов для формирования профиля.