

Модели жизненного  
цикла,  
Каноническое и  
типовое  
проектирование ИС

Жизненный цикл  
информационной системы.  
Модели жизненного цикла

# Жизненный цикл информационной системы

Жизненный цикл ИС – это непрерывный процесс, который начинается с момента принятия решения о необходимости создания ИС и заканчивается в момент полного изъятия из эксплуатации ИС.

# Модели жизненного цикла

Под моделью ЖЦ ИС понимается структура, определяющая последовательность выполнения и взаимосвязи процессов действий и задач на протяжении жизненного цикла.

**Модель жизненного цикла ИС**— это структура, описывающая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения в течение всего жизненного цикла системы.

# Модели жизненного цикла

Выбор модели жизненного цикла зависит от специфики, масштаба, сложности проекта и набора условий, в которых АИС создается и функционирует.

В соответствии с известными моделями ЖЦ программного обеспечения определяют модели ЖЦ ИС:

- **каскадную;**
- **итерационную;**
- **спиральную.**

# Каскадная модель



описывает классический подход к разработке систем в любых предметных областях; широко использовалась в 1970—80-х гг.

# Каскадная модель



Каскадная модель предусматривает последовательную организацию работ, причем основной особенностью модели является разбиение всей работы на этапы.

Переход от предыдущего этапа к последующему происходит только после полного завершения всех работ предыдущего.

# Каскадная модель



Выделяют **пять** устойчивых этапов разработки, практически не зависящих от предметной области:

- На первом этапе проводится исследование проблемной области, формулируются требования заказчика. Результатом данного этапа является техническое задание (задание на разработку), согласованное со всеми заинтересованными сторонами.
- В ходе второго этапа, согласно требованиям технического задания, разрабатываются те или иные проектные решения. В результате появляется комплект проектной документации.
- Третий этап — реализация проекта; по существу, разработка программного обеспечения (кодирование) в соответствии с проектными решениями предыдущего этапа. Методы реализации при этом принципиального значения не имеют. Результатом выполнения этапа является готовый программный продукт.
- На четвертом этапе проводится проверка полученного программного обеспечения на предмет соответствия требованиям, заявленным в техническом задании. Опытная эксплуатация позволяет выявить различного рода скрытые недостатки, проявляющиеся в реальных условиях работы ИС.
- Последний этап — сдача готового проекта.



# Каскадная модель



На каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности. На заключительных этапах разрабатывается пользовательская документация, охватывающая все предусмотренные стандартами виды обеспечения АИС (организационное, информационное, программное, техническое и т. д.). Последовательное выполнение этапов работ позволяет планировать сроки завершения и соответствующие затраты.

При этом для каскадной модели необходимо отметить существенную задержку в получении результатов, сложность параллельного ведения работ по проекту и сложность управления проектом, и как следствие, высокий уровень риска и ненадежность вложений в ИС. Кроме того, ошибки и недоработки на любом из этапов проявляются, как правило, на последующих этапах работ, что приводит к необходимости возврата.

# Итерационная модель



- заключается в серии коротких циклов (шагов) по планированию, реализации, изучению, действию. Разработка ИС ведется *итерациями* с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.

# Итерационная модель

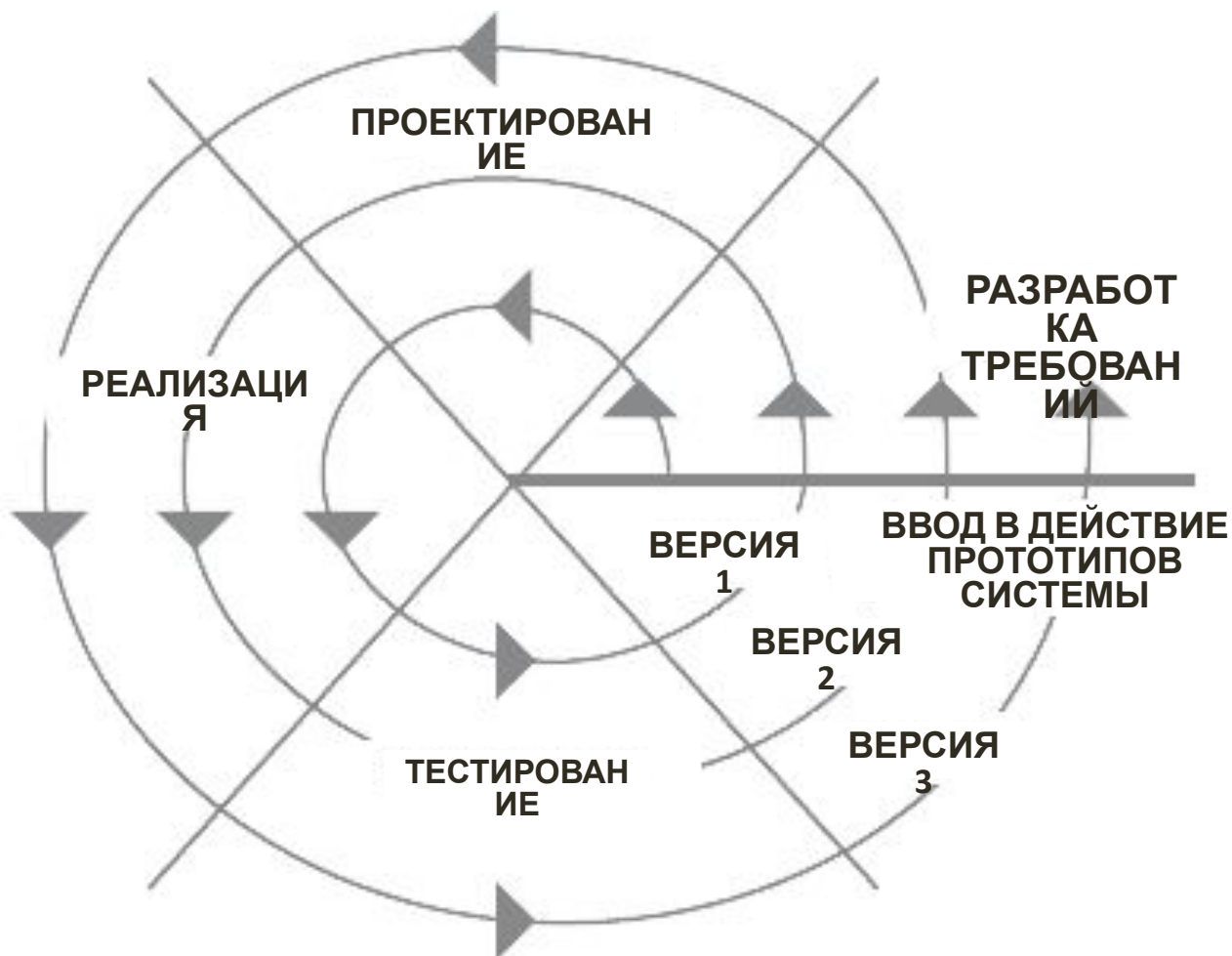


Создание сложных ИС предполагает проведение согласований проектных решений, полученных при реализации отдельных задач. Подход к проектированию «снизу — вверх» обуславливает необходимость таких итераций возвратов, когда проектные решения по отдельным задачам объединяются в общие системные. При этом возникает потребность в пересмотре ранее сформировавшихся требований.

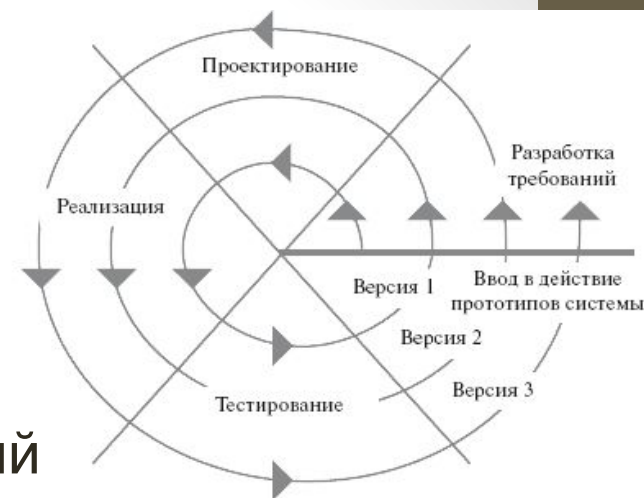
В итерационной модели межэтапные корректировки обеспечивают меньшую трудоемкость разработки по сравнению с каскадной моделью.

При этом, время жизни каждого этапа растягивается на весь период разработки, вследствие большого числа итераций возникают рассогласования выполнения проектных решений и документации, возможно появление на стадии внедрения необходимости перепроектирования всей системы.

# Спиральная модель



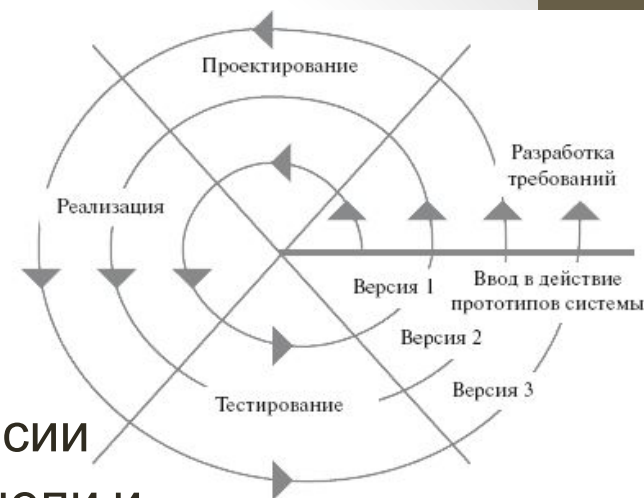
# Спиральная модель



в отличие от каскадной, но аналогично предыдущей предполагает итерационный процесс разработки ИС. При этом возрастает значение начальных этапов, таких как анализ и проектирование, на которых проверяется и обосновывается реализуемость технических решений путем создания прототипов.

Каждая итерация представляет собой законченный цикл разработки, приводящий к выпуску внутренней или внешней версии изделия (или подмножества конечного продукта), которое совершенствуется от итерации к итерации, чтобы стать законченной системой.

# Спиральная модель



Таким образом, каждый виток спирали соответствует созданию фрагмента или версии программного изделия, на нем уточняются цели и характеристики проекта, определяется его качество, планируются работы на следующем витке спирали. Каждая итерация служит для углубления и последовательной конкретизации деталей проекта, в результате этого выбирается обоснованный вариант окончательной реализации.

Использование спиральной модели позволяет осуществлять переход на следующий этап выполнения проекта, не дожидаясь полного завершения текущего, — недоделанную работу можно будет выполнить на следующей итерации. Главная задача каждой итерации — как можно быстрее создать работоспособный продукт для демонстрации пользователям. Таким образом, существенно упрощается процесс внесения уточнений и дополнений проект.

# Спиральная модель

Спиральный подход к разработке программного обеспечения позволяет преодолеть большинство недостатков каскадной модели, кроме того, обеспечивает ряд дополнительных возможностей, делая процесс разработки более гибким. При использовании спиральной модели существенно упрощается внесение изменений в проект при изменении требований заказчика, происходит снижение уровня рисков (уровень рисков максимален в начале разработки проекта, по мере продвижения разработки он снижается), обеспечивается большая гибкость в управлении проектом за счет возможности внесения тактических изменений в разрабатываемое изделие, возможность совершенствовать процесс разработки — в результате анализа в конце каждой итерации проводится оценка изменений в организации разработки; на следующей итерации она улучшается, упрощается повторное использование компонентов, поскольку гораздо проще выявить (идентифицировать) общие части проекта, когда они уже частично разработаны, чем пытаться выделить их в самом начале проекта. Спиральная модель позволяет получить более надежную и устойчивую систему. Это связано с тем, что по мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации. Одновременно корректируются критические параметры эффективности, что в случае каскадной модели доступно только перед внедрением системы. Вовлечение пользователей в процесс проектирования и копирования приложения позволяет получать замечания и дополнения к требованиям непосредственно в процессе проектирования приложения, сокращая время разработки. Представители заказчика получают возможность контролировать процесс создания системы и влиять на ее функциональное



# Спиральная модель

Однако, организация проектирования ИС по спиральной модели обычно имеет высокую стоимость (поэтому ее имеет смысл использовать для сложных и дорогостоящих систем). Модель имеет сложную структуру, что может затруднить ее применение на практике неподготовленными специалистами и заказчиками. Спираль может продолжаться до бесконечности, поскольку каждая ответная реакция заказчика на созданную версию может породить новый цикл работ. Большое количество промежуточных стадий усложняет ведение документации проекта. Возможны затруднения в определении момента перехода на следующую итерацию цикла. Обычно вводят временные ограничения на выполнение итерации и каждого из ее этапов.



В некоторых ситуациях применение спиральной модели невозможно или ограничено, поскольку невозможно использование/тестирование продукта, обладающего неполной функциональностью (например, военные разработки, атомная энергетика и т.д.). Поэтапное итерационное внедрение корпоративных информационных систем обычно сопряжено с организационными сложностями (перенос данных, интеграция систем, изменение бизнес-процессов, учетной политики, обучение пользователей). Трудозатраты при поэтапном итерационном внедрении оказываются значительно выше, а неграмотное управление процессом внедрения может свести на нет все полученные результаты. По этой причине на этапе внедрения часто обходятся без итерационных моделей, внедряя систему «раз и навсегда».



# ЭТАПЫ СОЗДАНИЯ ИС

Процесс проектирования АИС регламентирован следующей документацией (стандартами, методологиями, моделями):

- **ГОСТ 34.601-90.** Введен в действие 01.01.1992. Устанавливает стадии и этапы создания автоматизированных систем и дает содержание работ на каждой стадии. Стадии и этапы работы, закрепленные в стандарте, соответствуют *каскадной модели* жизненного цикла.

# Процесс проектирования АИС регламентирован следующей документацией (стандартами, методологиями, моделями):

- ISO/IEC 12207:1995. Международный стандарт, описывающий процессы жизненного цикла программного обеспечения. Содержит описание более, чем 220 базовых работ, выполнение которых может потребоваться в процессе создания ИС. Все процессы ЖЦ ПО подразделяются на три большие группы:
  - Основные процессы (приобретение, поставка, разработка, эксплуатация, сопровождение);
  - Вспомогательные процессы (документирование, управление конфигурацией, обеспечение качества, разрешение проблем, аудит, аттестация, совместная оценка, верификация);
  - Организационные процессы (создание инфраструктуры, управление, обучение, усовершенствование).

Для реализации положений стандарта должны быть выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации жизненного цикла программного обеспечения и не противоречащие предварительно скомпонованному набору нормативных документов. Чтобы облегчить практическое применение стандарта, международной организацией по стандартизации были разработаны и утверждены следующие документы:

- ISO/IEC TR 15271:1998 – руководство по применению ISO/IEC 12207;
- ISO/IEC TR 16326:1999 – руководство по управлению проектами при использовании ISO/IEC 12207.

Процесс проектирования АИС регламентирован следующей документацией (стандартами, методологиями, моделями):

- **ISO/IEC 15288:2002.** Международный стандарт, описывающий возможные процессы жизненного цикла систем, созданных человеком. Был создан с учетом опыта проектирования автоматизированных информационных систем, а также с привлечением специалистов различных областей: системной инженерии, программирования, администрирования, управления качеством, безопасностью и т.д. Предполагается, что стандарт содержит полное множество процессов, которые могут протекать в ходе жизненного цикла системы. Таким образом, задача разработчика ИС заключается в формировании необходимого ему множества – среды процессов. В обзоре стандарта отмечается, что в нем не содержится описания методов и процедур, необходимых для обеспечения выполнения целей, задач и результатов указанных процессов. В 2003 году выпущено руководство по применению стандарта (ISO/IEC TR 19760:2003). В настоящее время продолжается работа над подготовкой новой редакции стандарта серии 15288.

Процесс проектирования АИС регламентирован следующей документацией (стандартами, методологиями, моделями):

- **Rational Unified Process (RUP)** – концепция итеративной (спиральной) разработки программного обеспечения, предложенная фирмой Rational Software (ныне – подразделение IBM). Жизненный цикл ИС представляет собой четыре фазы: начало (inception), исследование (elaboration), конструирование (construction) и внедрение (transition). Каждая фаза может содержать в себе несколько итераций. Кроме того, завершение всех четырех фаз не всегда означает завершение работы над проектом – его развитие может продолжиться новым циклом. В рамках итераций производится создание взаимосогласованных моделей, которые описываются на специально разработанном языке UML (Unified Modeling Language).

Процесс проектирования АИС регламентирован следующей документацией (стандартами, методологиями, моделями):

- **Microsoft Solution Framework (MSF)**. Итерационная методология разработки приложений, аналогичная RUP. Так же включает четыре фазы: анализ, проектирование, разработка, стабилизация и предполагает использование объектно-ориентированного моделирования. По сравнению с RUP в большей степени ориентирована на разработку ИС для бизнеса.

Процесс проектирования АИС регламентирован следующей документацией (стандартами, методологиями, моделями):

- **Extreme Programming (XP)**. Экстремальное программирование – это самая новая среди рассматриваемых методологий (первые идеи были сформированы в середине 1990-ых). Основные принципы: командная работа, эффективное взаимодействие между заказчиком и исполнителем в течение всего времени разработки ИС, использование последовательно дорабатываемых прототипов, достижение максимальной гибкости разработки (адаптация к изменяющимся требованиям заказчика).

# Стандарт ISO/IEC 12207

все процессы ЖЦ ПО делятся на три группы:

- **Основные процессы:**
  - приобретение;
  - поставка;
  - разработка;
  - эксплуатация;
  - сопровождение.
- **Вспомогательные процессы:**
  - документирование;
  - *управление конфигурацией*;
  - обеспечение качества;
  - разрешение проблем;
  - аудит;
  - аттестация;
  - совместная оценка;
  - верификация.
- **Организационные процессы:**
  - создание инфраструктуры;
  - управление;
  - обучение;
  - усовершенствование.



# Содержание основных процессов ЖЦ ПО ИС (ISO/IEC 12207)

Процесс (исполнитель процесса)	Действия	Вход	Результат
Приобретение (заказчик)	<ul style="list-style-type: none"> <li>Инициирование</li> <li>Подготовка заявочных предложений</li> <li>Подготовка договора</li> <li>Контроль деятельности поставщика</li> <li>Приемка ИС</li> </ul>	<ul style="list-style-type: none"> <li>Решение о начале работ по внедрению ИС</li> <li>Результаты обследования деятельности заказчика</li> <li>Результаты анализа рынка ИС/ тендера</li> <li>План поставки/ разработки</li> <li>Комплексный тест ИС</li> </ul>	<ul style="list-style-type: none"> <li>Технико-экономическое обоснование внедрения ИС</li> <li>Техническое задание на ИС</li> <li>Договор на поставку/ разработку</li> <li>Акты приемки этапов работы</li> <li>Акт приемно-сдаточных испытаний</li> </ul>
Поставка (разработчик ИС)	<ul style="list-style-type: none"> <li>Инициирование</li> <li>Ответ на заявочные предложения</li> <li>Подготовка договора</li> <li>Планирование исполнения</li> <li>Поставка ИС</li> </ul>	<ul style="list-style-type: none"> <li>Техническое задание на ИС</li> <li>Решение руководства об участии в разработке</li> <li>Результаты тендера</li> <li>Техническое задание на ИС</li> <li>План управления проектом</li> <li>Разработанная ИС и документация</li> </ul>	<ul style="list-style-type: none"> <li>Решение об участии в разработке</li> <li>Коммерческие предложения/ конкурсная заявка</li> <li>Договор на поставку/ разработку</li> <li>План управления проектом</li> <li>Реализация/ корректировка</li> <li>Акт приемно-сдаточных испытаний</li> </ul>
Разработка (разработчик ИС)	<ul style="list-style-type: none"> <li>Подготовка</li> <li>Анализ требований к ИС</li> <li>Проектирование архитектуры ИС</li> <li>Разработка требований к ПО</li> <li>Проектирование архитектуры ПО</li> <li>Детальное проектирование ПО</li> <li>Кодирование и тестирование ПО</li> <li>Интеграция ПО и квалификационное тестирование ПО</li> <li>Интеграция ИС и квалификационное тестирование ИС</li> </ul>	<ul style="list-style-type: none"> <li>Техническое задание на ИС</li> <li>Техническое задание на ИС, модель ЖЦ</li> <li>Подсистемы ИС</li> <li>Спецификации требования к компонентам ПО</li> <li>Архитектура ПО</li> <li>Материалы детального проектирования ПО</li> <li>План интеграции ПО, тесты</li> <li>Архитектура ИС, ПО, документация на ИС, тесты</li> </ul>	<ul style="list-style-type: none"> <li>Используемая модель ЖЦ, стандарты разработки</li> <li>План работ</li> <li>Состав подсистем, компоненты оборудования</li> <li>Спецификации требования к компонентам ПО</li> <li>Состав компонентов ПО, интерфейсы с БД, план интеграции ПО</li> <li>Проект БД, спецификации интерфейсов между компонентами ПО, требования к тестам</li> <li>Тексты модулей ПО, акты автономного тестирования</li> <li>Оценка соответствия комплекса ПО требованиям ТЗ</li> <li>Оценка соответствия ПО, БД, технического комплекса и комплекта документации</li> </ul>

# Стандарт ISO/IEC 15288

## ГРУППЫ ПРОЦЕССОВ СТРУКТУРЫ ЖЦ:

- **Договорные процессы:**
  - приобретение (внутренние решения или решения внешнего поставщика);
  - поставка (внутренние решения или решения внешнего поставщика).
- **Процессы предприятия:**
  - управление окружающей средой предприятия;
  - инвестиционное управление;
  - управление ЖЦ ИС;
  - управление ресурсами;
  - управление качеством.
- **Проектные процессы:**
  - планирование проекта;
  - оценка проекта;
  - контроль проекта;
  - управление рисками;
  - *управление конфигурацией*;
  - управление информационными потоками;
  - принятие решений.
- **Технические процессы:**
  - определение требований;
  - *анализ требований*;
  - разработка архитектуры;
  - внедрение;
  - интеграция;
  - верификация;
  - переход;
  - аттестация;
  - эксплуатация;
  - сопровождение;
  - утилизация.
- **Специальные процессы:**
  - определение и установка взаимосвязей исходя из задач и целей.

# Стадии создания систем (ISO/IEC 15288)

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

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ИС

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. 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, прописываются в договорах и технических заданиях на выполнение работ:

- Стадия 1. **Формирование требований к ИС.**

На начальной стадии проектирования выделяют следующие этапы *работ*:

- *обследование* объекта и обоснование необходимости создания ИС;
- формирование требований пользователей к ИС;
- оформление отчета о выполненной работе и тактико- *технического задания* на разработку.

- Стадия 2. **Разработка концепции ИС.**

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

Стадии и этапы создания ИС, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ:

- 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.

Стадии и этапы создания ИС, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ:

- **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.



Стадии и этапы создания ИС, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ:

- **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;
  - послегарантийное обслуживание.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

**Обследование** - это изучение и диагностический анализ организационной структуры предприятия, его деятельности и существующей системы обработки информации. Материалы, полученные в результате *обследования*, используются для:

- обоснования разработки и поэтапного внедрения систем;
- составления *технического задания* на разработку систем;
- разработки технического и рабочего проектов систем.
- На этапе *обследования* целесообразно выделить две составляющие: *определение* стратегии внедрения ИС и *детальный анализ* деятельности организации.

Основная задача первого этапа *обследования* - оценка реального объема проекта, его целей и задач на основе выявленных функций и информационных элементов автоматизируемого объекта высокого уровня. Эти задачи могут быть реализованы или заказчиком ИС самостоятельно, или с привлечением консалтинговых организаций. Этап предполагает тесное взаимодействие с основными потенциальными пользователями системы и бизнес-экспертами. Основная задача взаимодействия - получить полное и однозначное понимание *требований заказчика*. Как правило, *нужная информация* может быть получена в результате интервью, бесед или семинаров с руководством, экспертами и пользователями.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

По завершении этой стадии *обследования* появляется возможность определить вероятные технические подходы к созданию системы и оценить *затраты* на ее реализацию (*затраты* на аппаратное обеспечение, закупаемое программное обеспечение и разработку нового программного обеспечения ).

Результатом этапа определения стратегии является документ ( **технико-экономическое обоснование проекта** ), где четко сформулировано, что получит заказчик, если согласится финансировать проект, когда он получит готовый продукт (*график* выполнения работ) и сколько это будет стоить (для крупных проектов должен быть составлен *график* финансирования на разных этапах работ). В документе желательно отразить не только *затраты*, но и выгоду проекта, например время окупаемости проекта, ожидаемый экономический эффект (если его удастся оценить).

Ориентировочное содержание этого документа:

- ограничения, риски, критические факторы, которые могут повлиять на успешность проекта;
- совокупность условий, при которых предполагается эксплуатировать будущую систему: архитектура системы, аппаратные и программные ресурсы, условия функционирования, обслуживающий персонал и пользователи системы;
- сроки завершения отдельных этапов, форма приемки/сдачи работ, привлекаемые ресурсы, меры по защите информации;
- описание выполняемых системой функций;
- возможности развития системы;
- информационные объекты системы;
- интерфейсы и *распределение функций* между человеком и системой;
- требования к программным и информационным компонентам ПО, требования к СУБД;
- что не будет реализовано в рамках проекта.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

На этапе детального анализа деятельности организации изучаются задачи, обеспечивающие реализацию функций управления, организационная структура, штаты и содержание *работ по* управлению предприятием, а также характер подчиненности вышестоящим органам управления. На этом этапе должны быть выявлены:

- инструктивно-методические и директивные материалы, на основании которых определяются состав подсистем и перечень задач;
- возможности применения новых методов решения задач.

Аналитики собирают и фиксируют информацию в двух взаимосвязанных формах:

- функции - информация о событиях и процессах, которые происходят в бизнесе;
- сущности - информация о вещах, имеющих значение для организации и о которых что-то известно.

При изучении каждой функциональной задачи управления определяются:

- наименование задачи; сроки и периодичность ее решения;
- степень *формализуемости задачи*;
- источники информации, необходимые для решения задачи;
- показатели и их количественные характеристики;
- порядок корректировки информации;
- действующие алгоритмы расчета показателей и возможные методы контроля;
- действующие средства сбора, передачи и обработки информации;
- действующие средства связи;
- принятая точность решения задачи;
- трудоемкость решения задачи;
- действующие формы представления исходных данных и результатов их обработки в виде документов;
- потребители результатной информации по задаче.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Одной из наиболее трудоемких, хотя и хорошо формализуемых задач этого этапа является описание документооборота организации. При *обследовании* документооборота составляется схема маршрута движения документов, которая должна отразить:

- количество документов;
- место формирования показателей документа;
- взаимосвязь документов при их формировании;
- маршрут и длительность движения документа;
- место использования и хранения данного документа;
- внутренние и внешние информационные связи;
- объем документа в знаках.

По результатам *обследования* устанавливается перечень задач управления, решение которых целесообразно автоматизировать, и очередность их разработки.

На этапе *обследования* следует классифицировать планируемые функции системы по степени важности. Один из возможных форматов представления такой классификации - MuSCoW.

Эта аббревиатура расшифровывается так: Must have - необходимые функции; Should have - желательные функции; Could have - возможные функции; Won't have - отсутствующие функции.

Функции первой категории обеспечивают критичные для успешной работы системы возможности.

Реализация функций второй и третьей категорий ограничивается временными и финансовыми рамками: разрабатывается то, что необходимо, а также максимально возможное в порядке приоритета число функций второй и третьей категорий.

Последняя *категория функций* особенно важна, поскольку необходимо четко представлять границы проекта и набор функций, которые будут отсутствовать в системе.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Модели деятельности организации создаются в двух видах:

- **модель "как есть"** ("as-is")- отражает существующие в организации бизнес-процессы;
- **модель "как должно быть"** ("to-be") - отражает необходимые изменения бизнес-процессов с учетом внедрения ИС.

На этапе анализа необходимо привлекать к работе группы тестирования для решения следующих задач:

- получения сравнительных характеристик предполагаемых к использованию аппаратных платформ, операционных систем, СУБД, иного окружения;
- разработки плана работ по обеспечению надежности информационной системы и ее тестирования.

Привлечение *тестировщиков* на ранних этапах разработки является целесообразным для любых проектов. Если проектное решение оказалось неудачным и это обнаружено слишком поздно (на этапе разработки или, что еще хуже, на этапе внедрения в эксплуатацию), то исправление ошибки проектирования обходится очень дорого. Чем раньше группы тестирования выявляют ошибки в информационной системе, тем ниже *стоимость* сопровождения системы. Время на тестирование системы и на исправление обнаруженных ошибок следует предусматривать не только на этапе разработки, но и на этапе проектирования.

Для *автоматизации тестирования* следует использовать системы отслеживания ошибок (*bug tracking*). Это позволяет иметь единое хранилище ошибок, отслеживать их повторное появление, контролировать скорость и эффективность исправления ошибок, видеть наиболее нестабильные компоненты системы, а также поддерживать *связь* между группой разработчиков и группой тестирования (уведомления об изменениях *по e-mail* и т.п.). Чем больше проект, тем сильнее потребность в *bug tracking*.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Результаты *обследования* представляют объективную основу для формирования *технического задания* на информационную систему.

***Техническое задание*** - это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки *автоматизированной системы управления*.

При разработке *технического задания* необходимо решить следующие задачи:

- установить общую цель создания ИС, определить состав подсистем и функциональных задач;
- разработать и обосновать требования, предъявляемые к подсистемам;
- разработать и обосновать требования, предъявляемые к информационной базе, математическому и программному обеспечению, комплексу технических средств (включая средства связи и передачи данных);
- установить общие требования к проектируемой системе;
- определить перечень задач создания системы и исполнителей;
- определить этапы создания системы и сроки их выполнения;
- провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения.

# Состав и содержание технического задания (ГОСТ 34.602- 89)

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



# Состав и содержание технического задания (ГОСТ 34.602- 89)

№ п\п	Раздел	Содержание
4	Требования к системе	<p>Требования к системе в целом:</p> <ul style="list-style-type: none"><li>• требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, <i>степень централизации</i>, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы)</li><li>• требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки)</li><li>• показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров)</li><li>• требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации</li></ul> <p>Требования к функциям (по подсистемам) :</p> <ul style="list-style-type: none"><li>• перечень подлежащих автоматизации задач</li><li>• временной регламент реализации каждой функции</li><li>• требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов</li><li>• перечень и критерии отказов</li></ul> <p>Требования к видам обеспечения:</p> <ul style="list-style-type: none"><li>• математическому (состав и область применения мат. моделей и методов, типовых и разрабатываемых алгоритмов)</li><li>• информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые <i>классификаторы</i>, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам)</li><li>• лингвистическому (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода- вывода)</li><li>• программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ)</li><li>• техническому</li><li>• метрологическому</li><li>• организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала)</li><li>• методическому (состав нормативно-технической документации)</li></ul>

# Состав и содержание технического задания (ГОСТ 34.602- 89)

№ п\п	Раздел	Содержание
5	Состав и содержание работ по созданию системы	<ul style="list-style-type: none"><li>•перечень стадий и этапов работ</li><li>•сроки исполнения</li><li>•состав организаций — исполнителей работ</li><li>•вид и порядок экспертизы технической документации</li><li>•программа обеспечения надежности</li><li>•программа метрологического обеспечения</li></ul>
6	Порядок контроля и приемки системы	<ul style="list-style-type: none"><li>•виды, состав, объем и методы испытаний системы</li><li>•общие требования к приемке работ по стадиям</li><li>•статус приемной комиссии</li></ul>
7	Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие	<ul style="list-style-type: none"><li>•преобразование <i>входной информации</i> к машиночитаемому виду</li><li>•изменения в объекте автоматизации</li><li>•сроки и порядок комплектования и обучения персонала</li></ul>
8	Требования к документированию	<ul style="list-style-type: none"><li>•перечень подлежащих разработке документов</li><li>•перечень документов на машинных носителях</li></ul>
9	Источники разработки	документы и информационные материалы, на основании которых разрабатывается ТЗ и система

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

**Эскизный проект** предусматривает разработку предварительных проектных решений по системе и ее частям.

Выполнение стадии эскизного проектирования не является строго обязательной. Если основные проектные решения определены ранее или достаточно очевидны для конкретной ИС и объекта автоматизации, то эта стадия может быть исключена из общей последовательности работ.

Содержание эскизного проекта задается в ТЗ на систему. Как правило, 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.

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

По результатам проделанной работы оформляется, согласовывается и утверждается документация в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию системы.

На основе технического задания (и эскизного проекта ) разрабатывается технический проект ИС.

**Технический проект** системы - это техническая документация, содержащая общесистемные проектные решения, алгоритмы решения задач, а также оценку экономической эффективности автоматизированной системы управления и перечень мероприятий по подготовке объекта к внедрению.

На этом этапе осуществляется комплекс научно-исследовательских и экспериментальных работ для выбора основных проектных решений и расчет экономической эффективности системы.

# Состав и содержание технического проекта

№ п\п	Раздел	Содержание
1	Пояснительная записка	<ul style="list-style-type: none"><li>•основания для разработки системы</li><li>•перечень организаций разработчиков</li><li>•краткая характеристика объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами</li><li>•краткие сведения об основных проектных решениях по функциональной и обеспечивающим частям системы</li></ul>
2	Функциональная и организационная структура системы	<ul style="list-style-type: none"><li>•обоснование выделяемых подсистем, их перечень и назначение</li><li>•перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания</li><li>•схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы</li></ul>
3	Постановка задач и алгоритмы решения	<ul style="list-style-type: none"><li>•организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются)</li><li>•экономико-математическая модель задачи (структурная и развернутая форма представления)</li><li>•входная оперативная информация ( характеристика показателей, диапазон изменения, формы представления)</li><li>•нормативно-справочная информация ( НСИ) (содержание и формы представления)</li><li>•информация, хранимая для связи с другими задачами</li><li>•информация, накапливаемая для последующих решений данной задачи</li><li>•информация по внесению изменений ( система внесения изменений и перечень информации, подвергающейся изменениям)</li><li>•алгоритм решения задачи ( последовательность этапов расчета, схема, расчетные формулы)</li><li>•контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливаемой и хранимой информацией, формы выходных документов, заполненные по результатам решения экономико-технической задачи и в соответствии с разработанным алгоритмом расчета)</li></ul>

# Состав и содержание технического проекта

№ п\п	Раздел	Содержание
4	Организация информационной базы	<ul style="list-style-type: none"><li>•источники поступления информации и способы ее передачи</li><li>•совокупность показателей, используемых в системе</li><li>•состав документов, сроки и периодичность их поступления</li><li>•основные проектные решения по организации фонда НСИ</li><li>•состав НСИ, включая перечень реквизитов, их определение, диапазон изменения и перечень документов НСИ</li><li>•перечень массивов НСИ, их объем, порядок и частота корректировки информации</li><li>•структура фонда НСИ с описанием связи между его элементами; требования к технологии создания и ведения фонда</li><li>•методы хранения, поиска, внесения изменений и контроля</li><li>•определение объемов и потоков информации НСИ</li><li>•контрольный пример по внесению изменений в НСИ</li><li>•предложения по унификации документации</li></ul>
5	Альбом форм документов	
6	Система математического обеспечения	<ul style="list-style-type: none"><li>•обоснование структуры математического обеспечения</li><li>•обоснование выбора системы программирования</li><li>•перечень стандартных программ</li></ul>

# Состав и содержание технического проекта

№ п\п	Раздел	Содержание
7	Принцип построения комплекса технических средств	<ul style="list-style-type: none"><li>• описание и обоснование схемы технологического процесса обработки данных</li><li>• обоснование и выбор структуры комплекса технических средств и его функциональных групп</li><li>• обоснование требований к разработке нестандартного оборудования</li><li>• комплекс мероприятий по обеспечению надежности функционирования технических средств</li></ul>
8	Расчет экономической эффективности и системы	<ul style="list-style-type: none"><li>• сводная смета затрат, связанных с эксплуатацией систем</li><li>• расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства (объединения), снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений</li></ul>
9	Мероприятия по подготовке объекта к внедрению системы	<ul style="list-style-type: none"><li>• перечень организационных мероприятий по совершенствованию бизнес-процессов</li><li>• перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц</li></ul>
10	Ведомость	

# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

В завершение стадии технического проектирования производится разработка документации на поставку серийно выпускаемых изделий для комплектования ИС, а также определяются технические требования и составляются ТЗ на разработку изделий, не изготовляемых серийно.

На стадии "**рабочая документация**" осуществляется создание программного продукта и разработка всей сопровождающей документации. Документация должна содержать все необходимые и достаточные сведения для обеспечения выполнения *работ по вводу ИС в действие* и ее эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы. Разработанная документация должна быть соответствующим образом оформлена, согласована и утверждена.

Для ИС, которые являются разновидностью автоматизированных систем, устанавливают следующие основные виды испытаний: предварительные, *опытная эксплуатация* и приемочные. При необходимости допускается дополнительно проведение других видов испытаний системы и ее частей.

В зависимости от взаимосвязей частей ИС и объекта автоматизации испытания могут быть автономные или комплексные. Автономные испытания охватывают части системы. Их проводят *по мере готовности* частей системы к сдаче в *опытную эксплуатацию*. Комплексные испытания проводят для групп взаимосвязанных частей или для системы в целом.



# КАНОНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

- Для планирования проведения всех видов испытаний разрабатывается документ "*Программа и методика испытаний*". Разработчик документа устанавливается в договоре или ТЗ. В качестве приложения в документ могут включаться тесты или контрольные примеры.
- ***Предварительные испытания*** проводят для определения работоспособности системы и решения вопроса о возможности ее приемки *вопытную эксплуатацию*. *Предварительные испытания* следует выполнять после проведения разработчиком отладки и тестирования поставляемых программных и технических средств системы и представления им соответствующих документов об их готовности к испытаниям, а также после ознакомления персонала ИС с эксплуатационной документацией.
- ***Опытную эксплуатацию*** системы проводят с целью определения фактических значений количественных и качественных характеристик системы и готовности персонала к работе в условиях ее функционирования, а также определения фактической эффективности и корректировки, при необходимости, документации.
- ***Приемочные испытания*** проводят для определения соответствия системы *техническому заданию*, оценки качества *опытной эксплуатации* решения вопроса о возможности приемки системы в постоянную эксплуатацию.

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

Методы *типового проектирования ИС* достаточно подробно рассмотрены в литературе.

***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 имеющиеся на рынке  *типовые проектные решения*, которые настраиваются на особенности конкретного предприятия.

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

**Типовое проектное решение (ТПР)** - это тиражируемое (пригодное к многократному использованию) проектное решение.

Принятая классификация *ТПР* основана на уровне декомпозиции системы. Выделяются следующие классы *ТПР*:

- элементные *ТПР* - типовые решения по задаче или по отдельному виду обеспечения задачи (информационному, программному, техническому, математическому, организационному);
- подсистемные *ТПР* - в качестве элементов типизации выступают отдельные подсистемы, разработанные с учетом *функциональной полноты* и минимизации внешних информационных связей;
- объектные *ТПР* - типовые отраслевые проекты, которые включают полный набор функциональных и обеспечивающих подсистем ИС.
- Каждое типовое решение предполагает наличие, кроме собственно *функциональных элементов* (программных или аппаратных), документации с детальным описанием *ТПР* и процедур настройки в соответствии с требованиями разрабатываемой системы.

# Достоинства и недостатки ТПР

Класс ТПР Реализация ТПР	Достоинства	Недостатки
<p>Элементные ТПР</p> <p>Библиотеки методо-ориентированных программ</p>	<ul style="list-style-type: none"> <li>•обеспечивается применение модульного подхода к проектированию и документированию ИС</li> </ul>	<ul style="list-style-type: none"> <li>•большие затраты времени на сопряжение разнородных элементов вследствие информационной, программной и технической несовместимости</li> <li>•большие затраты времени на доработку ТПР отдельных элементов</li> </ul>
<p>Подсистемные ТПР</p> <p>Пакеты прикладных программ</p>	<ul style="list-style-type: none"> <li>•достигается высокая степень интеграции элементов ИС</li> <li>•позволяют осуществлять: модульное проектирование; параметрическую настройку программных компонентов на различные объекты управления</li> <li>•обеспечивают: сокращение затрат на проектирование и программирование взаимосвязанных компонентов; хорошее документирование отображаемых процессов обработки информации</li> </ul>	<ul style="list-style-type: none"> <li>•адаптивность ТПР недостаточна с позиции непрерывного инжиниринга деловых процессов</li> <li>•возникают проблемы в комплексировании разных функциональных подсистем, особенно в случае использования решений нескольких производителей программного обеспечения</li> </ul>
<p>Объектные ТПР</p> <p>Отраслевые проекты ИС</p>	<ul style="list-style-type: none"> <li>•комплексирование всех компонентов ИС за счет методологического единства и информационной, программной и технической совместимости</li> <li>•открытость архитектуры — позволяет устанавливать ТПР на разных программно-технических платформах</li> <li>•масштабируемость — допускает конфигурацию ИС для переменного числа рабочих мест</li> </ul>	<ul style="list-style-type: none"> <li>•проблемы привязки типового проекта к конкретному объекту управления, что вызывает в некоторых случаях даже необходимость изменения организационно-экономической структуры объекта автоматизации</li> </ul>

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

Для реализации *типового проектирования* используются два подхода:

- *параметрически-ориентированное;*
- *модельно-ориентированное проектирование.*

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

**Параметрически-ориентированное проектирование** включает следующие этапы: определение критериев оценки пригодности пакетов прикладных программ (*ППП*) для решения поставленных задач, анализ и оценка доступных *ППП* по сформулированным критериям, выбор и закупка наиболее подходящего пакета, настройка параметров (доработка) закупленного *ППП*.

Критерии оценки *ППП* делятся на следующие группы:

- назначение и возможности пакета;
- отличительные признаки и свойства пакета;
- требования к техническим и программным средствам;
- документация пакета;
- факторы финансового порядка;
- особенности установки пакета;
- особенности эксплуатации пакета;
- помощь поставщика по внедрению и поддержанию пакета;
- оценка качества пакета и опыт его использования;
- перспективы развития пакета.

Внутри каждой группы критериев выделяется некоторое *подмножество* частных показателей, детализирующих каждый из десяти выделенных аспектов анализа выбираемых *ППП*. Достаточно полный перечень показателей можно найти в литературе.

Числовые значения показателей для конкретных *ППП* устанавливаются экспертами по выбранной шкале оценок (например, 10-бальной). На их основе формируются групповые оценки и комплексная оценка пакета (путем вычисления средневзвешенных значений). Нормированные взвешивающие *коэффициенты* также получают экспертным путем.

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

**Модельно-ориентированное проектирование** заключается в адаптации состава и характеристик типовой ИС в соответствии с моделью объекта автоматизации.

Технология проектирования в этом случае должна обеспечивать единые средства для работы как с моделью типовой ИС, так и с моделью конкретного предприятия.

*Типовая ИС* в специальной базе метаданных - репозитории - содержит модель объекта автоматизации, на основе которой осуществляется *конфигурирование* программного обеспечения. Таким образом, *модельно-ориентированное проектирование ИС* предполагает, прежде всего, построение модели объекта автоматизации с использованием специального программного инструментария (например, *SAP Business Engineering Workbench (BEW)*, *BAAN Enterprise Modeler*). Возможно также создание системы на базе *типовой модели ИС* из репозитория, который поставляется вместе с программным продуктом и расширяется по мере накопления опыта проектирования информационных систем для различных отраслей и типов производства.

*Репозиторий* содержит базовую (ссылочную) модель ИС, типовые (референтные) модели определенных классов ИС, модели конкретных ИС предприятий.



# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

- **Базовая модель ИС** в репозитории содержит описание бизнес-функций, бизнес-процессов, бизнес-объектов, бизнес-правил, организационной структуры, которые поддерживаются программными модулями типовой ИС.
- **Типовые модели** описывают конфигурации информационной системы для определенных отраслей или типов производства.
- Модель конкретного предприятия строится либо путем выбора фрагментов основной или *типовой модели* в соответствии со специфическими особенностями предприятия (BAAN *Enterprise Modeler*), либо путем автоматизированной адаптации этих моделей в результате экспертного опроса (*SAP Business Engineering Workbench*).
- Построенная модель предприятия в виде метаописания хранится в репозитории и при необходимости может быть откорректирована. На основе этой модели автоматически осуществляется *конфигурирование* и настройка информационной системы.
- Бизнес-правила определяют условия корректности совместного применения различных компонентов ИС и используются для поддержания целостности создаваемой системы.
- Модель бизнес-функций представляет собой иерархическую декомпозицию функциональной деятельности предприятия (подробное описание см. в разделе "*Анализ и моделирование функциональной области внедрения ИС*").
- Модель бизнес-процессов отражает выполнение *работ* для функций самого нижнего уровня модели бизнес-функций (подробное описание см. в разделе "*Спецификация функциональных требований к ИС*"). Для отображения процессов используется модель управления событиями (EPC - Event-driven Process Chain). Именно модель бизнес-процессов позволяет выполнить настройку программных модулей - приложений информационной системы в соответствии с характерными особенностями конкретного предприятия.
- Модели бизнес-объектов используются для интеграции приложений, поддерживающих *исполнение* различных бизнес-процессов (подробное описание см. в разделе "*Этапы проектирования ИС с применением UML*").

# ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

Модель организационной структуры предприятия представляет собой традиционную иерархическую структуру подчинения подразделений и персонала (подробное описание см. в разделе "*Анализ и моделирование* функциональной области внедрения ИС").

Внедрение типовой информационной системы начинается с анализа требований к конкретной ИС, которые выявляются на основе результатов предпроектного *обследования* объекта автоматизации (см. раздел "*Анализ и моделирование* функциональной области внедрения ИС"). Для оценки соответствия этим требованиям программных продуктов может использоваться описанная выше методика оценки ППП. После выбора программного продукта на базе имеющихся в нем референтных моделей строится предварительная модель ИС, в которой отражаются все особенности реализации ИС для конкретного предприятия. Предварительная модель является основой для выбора *типовой модели* системы и определения перечня компонентов, которые будут реализованы с использованием других программных средств или потребуют разработки с помощью имеющихся в составе типовой ИС инструментальных средств (например, АВАР в SAP, *Tools* в BAAN).

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.