МДК.05.01 Проектирование и дизайн информационных систем

2 Лекция. Разработка ТЗ. Этапы создания. Разработка эскизного проекта. Разработка технического проекта.

**Цель лекции:** Сформировать понимание целей, структуры и содержания ключевых проектных документов на этапах предпроектного и технического проектирования ИС.  
**Ключевая мысль:** Качественное проектирование на 80% определяет успех всего проекта. "Семь раз отмерь (спроектируй), один раз отрежь (напиши код)".

**План лекции:**

1. **Введение: Место проектирования в жизненном цикле разработки ИС.**
2. **Техническое задание (ТЗ) как основа проекта:**
   * Цель, назначение и правовой статус ТЗ.
   * Структура ТЗ согласно ГОСТ 34.602-89.
   * Процесс и этапы разработки ТЗ.
3. **Эскизный проект (ЭП):**
   * Цели и задачи этапа эскизного проектирования.
   * Основные компоненты и документация эскизного проекта.
4. **Технический проект (ТП):**
   * Цели и задачи этапа технического проектирования.
   * Детальный разбор компонентов технического проекта. Отличие от ЭП.
5. **Сравнительная таблица: ТЗ, ЭП, ТП.**
6. **Заключение: Важность последовательного проектирования.**

**1. Введение: Жизненный цикл разработки ИС**

Создание сложной ИС — это не кодирование, а **проект**. Любой проект требует тщательного планирования и проектирования. Упрощенно жизненный цикл можно представить так:

1. **Предпроектная стадия** (анализ, формирование требований) -> **Техническое задание (ТЗ)**
2. **Проектирование** (создание архитектуры и моделей) -> **Эскизный проект (ЭП)** -> **Технический проект (ТП)**
3. **Разработка** (написание кода)
4. **Тестирование**
5. **Ввод в эксплуатацию**
6. **Сопровождение**

*Сегодня мы подробно разбираем первые два пункта.*

**2. Техническое задание (ТЗ)**

**Техническое задание (ТЗ)** — это основной документ, определяющий цели, требования и порядок создания информационной системы. Он утверждается заказчиком и является **основой для договора** и **критерием приемки** готовой системы.

**Этапы разработки ТЗ:**

1. **Выявление и анализ требований:** Интервью, опросы, анкетирование, анализ бизнес-процессов заказчика.
2. **Структурирование и документирование требований:** Формализация собранной информации.
3. **Согласование с заказчиком:** Обсуждение, уточнение, утверждение.

**Типовая структура ТЗ (на основе ГОСТ 34.602-89):**

1. **Общие сведения:**
   * Наименование системы и проекта.
   * Наименования заказчика и разработчика.
   * Основание для разработки (приказ, договор).
2. **Назначение и цели создания системы:**
   * Что будет делать система? Какую бизнес-проблему решает?
3. **Характеристики объекта автоматизации:**
   * Описание бизнес-процессов компании-заказчика.
4. **Требования к системе (Функциональные требования):**
   * **Самый важный раздел!** Подробное описание того, *что* должна делать система. Часто оформляется в виде **Use Case** (вариантов использования) или пользовательских историй (**User Stories**).
   * *Пример: "Система должна позволять пользователю регистрировать новый заказ, внося данные: ФИО клиента, товар, количество, дата отгрузки".*
5. **Нефункциональные требования:**
   * Требования к производительности, безопасности, надежности, удобству использования (usability), масштабируемости.
   * *Пример: "Время отклика системы при любом запросе не должно превышать 2 секунд при нагрузке до 100 concurrent пользователей".*
6. **Состав и сроки выполнения работ:**
   * Этапы разработки и их сроки.
7. **Порядок контроля и приемки системы:**
   * Как будут тестировать систему? Критерии приемки.
8. **Приложения:** Могут включать схемы бизнес-процессов, глоссарий терминов.

**3. Эскизный проект (ЭП)**

**Цель этапа:** Разработать и утвердить **архитектурные и проектные решения высокого уровня**. Ответить на вопрос: "Как в общих чертах мы будем реализовывать требования из ТЗ?".

**Результатом этапа является пакет документов "Эскизный проект", который обычно включает:**

* **Архитектура системы:** Выбор типа архитектуры (монолит, микросервисы, клиент-сервер).
* **Схема компонентов системы:** Крупные блоки (модули) системы и их взаимодействие.
* **Схема базы данных:** Предварительная ER-диаграмма (сущности и связи).
* **Выбор технологий и сред:** Стек технологий (языки программирования, СУБД, фреймворки, серверы).
* **Интерфейсы взаимодействия:** Как система будет взаимодействовать с другими системами (API, протоколы обмена).
* **Предварительный план внедрения.**

**Эскизный проект согласуется с заказчиком, чтобы убедиться, что разработчик правильно понял ТЗ и предлагаемые решения устраивают заказчика.**

**4. Технический проект (ТП)**

**Цель этапа:** Разработать **детальные проектные решения**, на основе которых можно **непосредственно писать код**. Это углубление и детализация эскизного проекта.

**Результатом этапа является пакет документов "Технический проект", который включает:**

* **Детальные схемы компонентов и модулей:** Состав каждого модуля, его функции.
* **Подробная схема базы данных:** Полная нормализованная ER-диаграмма, даталогическая и физическая модель данных (названия таблиц, столбцов, типы данных, индексы).
* **Алгоритмы:** Описание ключевых бизнес-алгоритмов (например, блок-схема расчета сложной отчетности).
* **Детальные спецификации интерфейсов:** Форматы запросов и ответов API (часто в формате OpenAPI/Swagger).
* **Структура классов и диаграммы последовательностей (UML):** Детальное проектирование на уровне объектов.
* **Макеты пользовательского интерфейса (UI):** Детальные прототипы или макеты всех экранов системы.
* **Технологические решения:** Конфигурация серверов, сетевые настройки.

**Технический проект — это последний этап, где всё продумывается на бумаге (в моделях), прежде чем начать дорогостоящий процесс написания кода.**

**5. Сравнительная таблица: ТЗ, ЭП, ТП**

| Критерий | Техническое задание (ТЗ) | Эскизный проект (ЭП) | Технический проект (ТП) |
| --- | --- | --- | --- |
| **Основной вопрос** | **ЧТО** сделать? (Требования) | **КАК** в общих чертах? (Архитектура) | **КАК** в деталях? (Реализация) |
| **Уровень детализации** | Высокоуровневое описание | Архитектурный уровень | Детальный, низкоуровневый |
| **Целевая аудитория** | Заказчик, руководство, аналитики | Заказчик, архитекторы, team leads | Разработчики, тестировщики, DevOps |
| **Основное содержание** | Функциональные и нефункциональные требования | Выбор технологий, схема компонентов, логика данных | Схемы БД, API-спекы, алгоритмы, макеты UI |
| **Правовой статус** | **Основа для договора и приемки** | Основа для согласования архитектуры | **Инструкция для разработчиков** |

**6. Заключение**

* Последовательная разработка **ТЗ -> ЭП -> ТП** позволяет выявить и исправить ошибки и недопонимание на самых ранних, **наименее затратных** этапах.
* Исправление ошибки на этапе проектирования в десятки раз дешевле, чем исправление в коде или после внедрения.
* Эти документы служат единственно верным источником истины для всех участников проекта: от заказчика до разработчика.
* Качество проектной документации напрямую влияет на скорость разработки, стоимость владения и итоговое качество информационной системы.

**Контрольные вопросы для аудитории:**

1. Почему ТЗ является основой для договора между заказчиком и разработчиком?
2. В чем ключевое различие между функциональными и нефункциональными требованиями? Приведите примеры.
3. Какой главный вопрос решается на этапе эскизного проектирования?
4. Почему технический проект называют "инструкцией для разработчиков"?
5. Что дороже: исправить ошибку в ТЗ или в готовом коде? Почему?