**1. Что должно быть в ТЗ?**

Технические задания разрабатываются под конкретный проект и, как правило, уникальны. Тем не менее есть пункты, которые в том или ином виде присутствуют во всех технических заданиях на разработку ПО

**a. Термины, сокращения и определения**

Использующиеся в тексте термины приводятся в начале документа. Это могут быть как IT-понятия – названия элементов, сред и языков программирования, технические определения, – так и слова и обозначения из той сферы, для которой предназначается IT-решение.

**b. Назначение продукта**

В этом блоке расписываются назначение IT-решения, цели его создания и целевая аудитория (К основным целям могут относиться увеличение клиентской базы, положительный имидж компании, увеличение производительности труда, уменьшение ручных операций и т.д.)

Пункт также содержит информацию о задачах проекта - **это может быть разработка с нуля и под ключ, участие в отдельном этапе создания (например, работа над программным обеспечением для готовой аппаратной части) или усовершенствование существующего продукта. Каждая из задач требует своего подхода и вводных данных, что нужно учитываться при создании помощника**

**c. Требования к продукту**

Самая весомая и развернутая часть ТЗ. Как правило, блок требований содержит следующие подразделы: общие требования, функциональные требования, нефункциональные требования, требования к разработке, требования к организации и качеству работ, требования к безопасности.

Помимо основных пунктов, **ТЗ содержит индивидуальные для конкретного проекта блоки**, например описание ролей для разграничения доступа, требования к интерфейсу, требования к размеру и внешнему виду устройства. Уточнения этих деталей должно происходить из контекста решаемой задачи и интервью с пользователем

**Общие требования** определяют последовательность процесса разработки. Например, так выглядят общие требования к проекту в ТЗ на разработку ПАК для управления оборудованием:

* Разработка контроллера для управления устройствами заказчика.Контроллер должен быть установлен на каждом устройстве.
* Создание пульта управления устройствами, используя одноплатный компьютер на базе Linux.
* Разработка приложения для удаленного управления.
* Интегрирование пульта управления и контроллера в единую систему управления.
* Тестирование.
* Возможные доработки и модификации системы.

**Функциональные требования** касаются функций и поведения IT-решения. Пример:

·       Система должна посылать уведомления о сбое в работе оборудования.

·       Система должна контролировать датчики.

·       Система должна передавать данные по радиоканалу.

·       Система должна иметь аварийную сигнализацию.

·       Система должна управлять всеми функциями устройства.

**Нефункциональные требования** определяют такие критерии, как производительность, масштабируемость, ремонтопригодность, безопасность продукта и многое другое.

**Требования к разработке** могут быть представлены несколькими пунктами, где подробно описываются этапы работ и используемые компоненты и инструменты. Например, в ТЗ на разработку программно-аппаратного комплекса может быть пункт, описывающий требования к разработке устройства, протоколов связи (MQTT, TCP и т.п.), пользовательского приложения для управления устройством.

**Требования к организации и качеству работ** определяют то, как будет организована работа над проектом, коммуникация между заказчиком и исполнителем, а также основные моменты, касающиеся качества системы, устройства или программного продукта – время непрерывной работы, поведение системы в аварийной ситуации и пр.

**Требования к безопасности** могут содержать требования о защите кода, разграничении доступа, прав и т.д.

**2. Ошибки при составлении ТЗ**

**a. Отсутствие словаря терминов**

Например, в техническом задании упоминаются аббревиатуры спортивных событий EPL, MLB, NBA и др, с которыми могут быть не знакомы, например, разработчики, но для качественного выполнения им это нужно. Конечно, есть гугл, но лучше в начале ТЗ дать расшифровку этих аббревиатур. Также расшифровку можно приводить при первом упоминании в тексте в круглых скобках, но не рекомендуется этим злоупотреблять

**b. Расплывчатые и непонятные формулировки**

**c. Перегруженное деталями описание**

Не стоит давать определения и пояснения каждому термину, встречающемуся в ТЗ. Если это будет важно, то читатель воспользуется иными источниками информации. В противном случае получится излишне полное ТЗ

**d. Путаница в функциональных и нефункциональных требованиях**

Пример:

**В спецификации от заказчика:** Система должна поддерживать температуру воды в нагревательной емкости не выше 50°С.

**Как должно быть:** Функциональные требования: Система должна поддерживать устанавливаемую пользователем температуру воды. Нефункциональные требования: Температура воды в нагревательном оборудовании не должна превышать 50°С.

**Как написать ТЗ на разработку IT-продукта. Что должно быть в ТЗ для разработчика**

**1. Что предшествует составлению ТЗ на разработку?**

Как правило до написания технического задания формируют бизнес и функциональные требования, они и станут основой ТЗ программного продукта

**a. Бизнес-требования**

**Бизнес-требования** — это задачи, которые должен решать IT-продукт, с какой целью этот продукт создается и как он поможет в достижении бизнес-показателей.

Например, бизнес-требованием можно назвать:

* Настроить фильтрацию каталога по ключевым словам: бренд, цвет, вид товара, производитель, страна производителя

**b. Функциональные требования**

**Функциональные требования (ФТ)** — это набор требований, которые должны быть реализованы, иными словами функционал, которым должна обладать система, без подробного описания.

Например, к функциональным требованиям можно отнести:

* Организовать сортировку результатов поиска по релевантности. Первыми должны идти те товары, у которых все слова поискового запроса находятся в одной строке. Далее идут товары у которых все слова встречаются в разных свойствах (при этом учитывается «вес свойства» см. ниже). Далее идут товары у которых встречается меньшее количество слов из поискового запроса
* Сделать возможность поиска по свойствам: бренд, суббренд, вид товара, название производителя, страна производитель
* Организовать сортировку результатов поиска в соответствии с «весом» свойства (название, бренд, суббренд, вид товара, состав, название производителя, страна производитель)

**2. Что должно быть в ТЗ для разработки IT-продукта?**

**a.     Информация о разрабатываемом IT-продукте (назначение и цели, из чего состоит IT-продукт и как он взаимодействует с системой)**

**b.    Глоссарий**

**c.     Требования к IT-продукту**

·       Бизнес-требования

·       Функциональные требования

·       Технические требования:

o   характеристики (нагрузка, которую должен выдержать сервис, производительность и т.д.)

o   требования к технологиям

o   требования к серверам

o   требования к скорости работы сервиса

o   обязательная интеграция со сторонними сервисами

o   требования к безопасности

o   и прочее. Таких требований может быть очень много, все зависит от того, какой продукт вы разрабатываете

**d.     Пользовательские сценарии**

Бывает полезно добавлять в ТЗ Use case — это ответ системы на действия пользователя. Пользовательские сценарии помогают всем участникам системы, в том числе тимлиду:

* Определить появление возможных проблем во время разработки
* Четко определить и предсказать поведение системы
* Упрощает приемку задачи, дает совпасть ожиданию и реальности
* Дает однозначность: упрощает жизнь для разработчика, тестировщика, постановщика

**ТЗ для IT проекта**

Помимо пунктов, которые упоминались в двух предыдущих статьях, добавляется «Взаимодействие с другими системами», если проект будет интегрирован с другими системами. В этом случае необходимо описать требования к взаимодействию и интерфейсам с внешними системами, используемые протоколы обмена данными и возможные ограничения

Помимо этого, есть пункты «Этапы разработки», «Управление рисками», «Тестирование и валидация» и «Документация. Для их написания требуются знания о составе и возможностях команды разработки, используемых технологиях и других деталях, поэтому их создание может быть проблематичным для предполагаемого сервиса

**Самый подробный план ТЗ**

Можно перед началом работы с пользователем выдать ему перечень с чекбоксами, чтобы он сам определил, что ему нужно.

На начальном этапе будем фокусироваться только на первых ключевых частях, которые должны быть в каждом документе с техническим заданием, а именно Введение, Назначение и цели продукта, пользовательские группы, обзор содержания и возможное взаимодействие с другими компонентами

1. **Введение**

* **Общие сведения**: Краткое описание проекта, его название и цель. Здесь также можно указать основные задачи и проблемы, которые решает проект
* **Основания для разработки**: Обоснование необходимости проекта и его актуальность. Это может включать анализ рынка, потребности пользователей или бизнес-цели заказчика

1. **Назначение и Цели Проекта**

* **Пользовательские цели**: Описание того, как пользователи будут использовать ПО. Например, для автоматизации рабочих процессов, управления данными или предоставления услуг
* **Бизнес-цели**: Описание того, как ПО будет способствовать достижению бизнес-целей заказчика. Это может включать увеличение эффективности, снижение затрат или улучшение качества обслуживания

1. **Пользовательские Группы**

* **Целевая аудитория**: Описание групп пользователей, для которых предназначено ПО. Это может включать сотрудников компании, клиентов или партнеров
* **Роли и ответственность**: Определение ролей и уровней доступа для разных групп пользователей. Например, администратор, менеджер или обычный пользователь

1. **Обзор Содержания**

* **Функциональные требования**: Описание функций и возможностей ПО. Например, регистрация пользователей, управление заказами или анализ данных
* **Пользовательские сценарии**: Описание того, как пользователи будут взаимодействовать с ПО. Например, как пользователь будет проходить регистрацию или как он будет использовать основные функции

1. **Взаимодействие с Другими Компонентами**

* **Интеграция с другими системами**: Описание того, как ПО будет интегрироваться с другими системами или сервисами. Например, интеграция с CRM-системой или платежными шлюзами
* **Протоколы обмена данными**: Указание протоколов, используемых для обмена данными между системами. Например, API, SOAP или REST

1. **Обзор Интерфейса**

* **Требования к дизайну**: Описание требований к внешнему виду и структуре интерфейса. Это может включать требования к цветовой схеме, шрифтовым стилям и расположению элементов
* **Фирменный стиль**: Указание на использование фирменного стиля заказчика. Это может включать логотипы, цветовые схемы и другие элементы брендинга

1. **Безопасность**

* **Системы безопасности**: Описание систем защиты данных и доступа. Например, использование SSL/TLS для шифрования данных или двухфакторная аутентификация
* **Шифрование и аутентификация**: Требования к шифрованию и аутентификации пользователей. Например, использование SHA-256 для хеширования паролей или OAuth для аутентификации через социальные сети

1. **Разработка**

* **Технологический стек**: Указание на используемые технологии и инструменты. Например, языки программирования (Java, Python), фреймворки (React, Angular) или базы данных (MySQL, PostgreSQL)
* **Языки программирования**: Определение языков, которые будут использоваться в разработке. Это может включать как основные языки, так и вспомогательные (например, SQL для работы с базами данных)

1. **Системное Окружение**

* **Технические характеристики**: Описание требований к серверам и устройствам, на которых будет работать ПО. Например, требования к оперативной памяти, процессору или операционной системе
* **Среда выполнения**: Указание на операционные системы и среды, поддерживаемые ПО. Например, Windows, macOS, Linux или мобильные операционные системы (iOS, Android)

**10. Требования к Качеству и Надежности**

* **Время непрерывной работы**: Требования к стабильности и доступности системы. Например, требование к работе без сбоев в течение определенного времени
* **Поведение в аварийных ситуациях**: Описание того, как система должна вести себя в случае ошибок или сбоев. Например, автоматическое восстановление после сбоев или уведомление администратора

**11. Этапы и Сроки Разработки**

* **План этапов**: Описание основных этапов разработки и их сроков. Например, этапы проектирования, разработки, тестирования и внедрения
* **Критерии приемки**: Установление критериев для приемки работ на каждом этапе. Например, успешное прохождение тестов или демонстрация функциональности

**12. Контроль и Приемка**

* **Процесс контроля**: Описание того, как будет осуществляться контроль за выполнением работ. Например, регулярные встречи с командой разработки или использование инструментов для отслеживания прогресса
* **Процедура приемки**: Указание на процедуру приемки готового ПО. Это может включать тестирование, проверку документации и подписание акта о приемке

**13. Требования к Документации**

* **Типы документации**: Описание видов документации, которые должны быть предоставлены вместе с ПО. Например, техническая документация, руководство пользователя или документация API
* **Формат документации**: Указание на форматы и стандарты документации. Например, PDF, HTML или Markdown

**14. Приложения**

* **Дополнительные материалы**: Приложение дополнительных материалов, таких как прототипы, схемы архитектуры, технические спецификации или другие документы, которые могут быть полезны для разработки