

Спецификация требований

[Документ в целом состоит из двух основных разделов: введения и самих требований. Но структуру разделов и подразделов можно менять, в зависимости от того как удобнее и принято в конкретном проекте. Например, можно выделить отдельно функциональные требования в один раздел, требования к интерфейсам — в другой, а остальные нефункциональные требования — в третий, как сделано в этом шаблоне. Аналогично, можно раздел введения на несколько.]

Введение

[Здесь кратко описывается назначение документа (не системы), описывается соглашения, принятые в документе, используемые нотации и так далее. Можно также описать для какой аудитории предназначен документ и каким типам читателей какие разделы могут быть особенно полезны].

Описание системы

Назначение

[Назначение системы, для бизнес-задачи она решает]

Система предназначена для автоматизации процесса обработки заявки в колл-центре оператора.

Границы системы и бизнес-требования

[Краткое описывает, что система будет делать — что будет в границе её зоны ответственности — определяет, какие бизнес-цели система предназначена реализовывать. При необходимости — можно отдельно указать, что находится вне границ системы.]

Система должна обеспечить:

- Увеличение производительности оператора колл-центра на 20%.
- Уменьшить среднее время обработки звонка до 3 минут.

[Диаграмма сценариев]

Краткое описание контекста использования

[Кратко перечисляет какие API и пользовательские интерфейсы система предоставляет и использует, какое аппаратное обеспечение нужно для её работы. Детальное описание требований к программным и пользовательским интерфейсам будет дальше.]

Система взаимодействует с внешними системами:

- Получает данные пользователей из CRM-системы посредством её SOAP API
- Получает данные заказов, товаров и остатков из ERP-системы посредством файлового обмена из общего FTP-сервера.
- Предоставляет информацию о звонках пользователя CRM- и ERP-системами посредством REST API.

[Контекстная диаграмма]

Заинтересованные лица

[Описание заинтересованных лиц и их роль в проекте.]

Типы пользователей

[Кратко перечисляет типы пользователей, требования которых система будет реализовывать, с их характеристиками.]

Основные функции

[Кратко перечисляет основные функции системы с точки зрения пользователей — фактически, пользовательские требования. Здесь можно использовать информацию из пользовательских историй или перечислить пользовательские сценарии использования]

Ограничения

[Основные ограничения, накладываемые на систему или её разработку внешней средой: технические ограничения, ограничения регуляторов, лицензионные ограничения и т. д.]

- Пользователи будут использовать систему, используя канал связи низкой пропускной способности.
- При разработке системы запрещается использовать open-source-компоненты.

- Система должна хранить данные пользовательских действий в течение 30 дней по требованиям федерального закона.

Определения

[Глоссарий терминов и расшифровка аббревиатур. Эти термины должны использоваться дальше по всему документу.]

Требования

[Основной раздел документ, содержащий системные требования. Требования (кроме сценариев) стоит формулировать в стандартной форме: [Условие] [субъект] должен [действие] [объект] [ограничение] [значение ограничения]]

[Также в раздел можно включать подразделы, описывающие архитектурные решения, если проект находится уже на стадии проектирования:]

- архитектурные принципы основные компоненты
- потоки данных между компонентами
- распределение функций между компонентами]

Сценарии использования

[Подразделы со сценариями использования. Их можно группировать, например по основным акторам. Раздел можно объединить с функциональными требованиями, где функциональные требования будут располагаться в разделах соответствующих сценариев.]

Функциональные

[Список функциональных системных требований их стоит сгруппировать, например по одному из признаков: типам пользователей, из требований которых вытекают системные требования к функциям исходным пользовательским требованиям (фичам) сценариям использования режимам работы]

Логическая модель данных

[Включает описание основных сущностей, их атрибутов и связей — как минимум на концептуальном уровне.]

[Если уже началось проектирование — то и логическую структуру БД. Здесь пригодятся UML Class Diagram или ERD.]

Программные интерфейсы

[Описание API, которой система будет предоставлять. Стоит расписать принципы, по которым строить API (REST/RPC/SOAP, файловый обмен и т. д.), методы, их входящие и возвращаемые параметры.]

[Можно ссылаться на объекты логической структуры данных. Если проектирование ещё не началось — можно описывать смысл методов, без всех подробностей.]

Пользовательские интерфейсы

[Структура и описание экранов пользовательского интерфейса.]

[Очень полезно добавлять макеты и снабжать их кратким описанием экранов.]

Удобство использования

[Должны быть конкретными, если их невозможно сформулировать — раздел не пишут.]

Пользовательский интерфейс мобильного приложения должен соответствовать требованиям рекомендаций Apple к iOS-приложениям

Цвета и шрифты пользовательского интерфейса веб-сайта должны соответствовать требованиям стандарта по инклюзивному дизайну WCAG 2.0

Атрибуты качества

[Остальные нефункциональные требования, описывающие качества работы системы: производительность, надёжность, доступность, информационная безопасность, поддерживаемость и другие. Нужно стараться указывать измеримые характеристики.]

Система должна обрабатывать 95% запросов от пользователей не дольше 1 секунду.

Система должна обеспечивать доступность на уровне 99,9% времени работы.