

Как делать ТЗ на отчет

Для постановки задачи на разработку программного решения, в т.ч. отчета, нужно состыковать области компетенций аналитика и разработчика. Для этого нужен общий язык: постановщику задач важно понимать как делаются отчеты, а разработчику требуется знать схему данных, предметную область и сценарии использования его решения.

В данной статье нашей задачей будет описание технологии разработки отчета для аналитика на примере [службы отчетов](#) Microsoft SQL Server.

Основные понятия

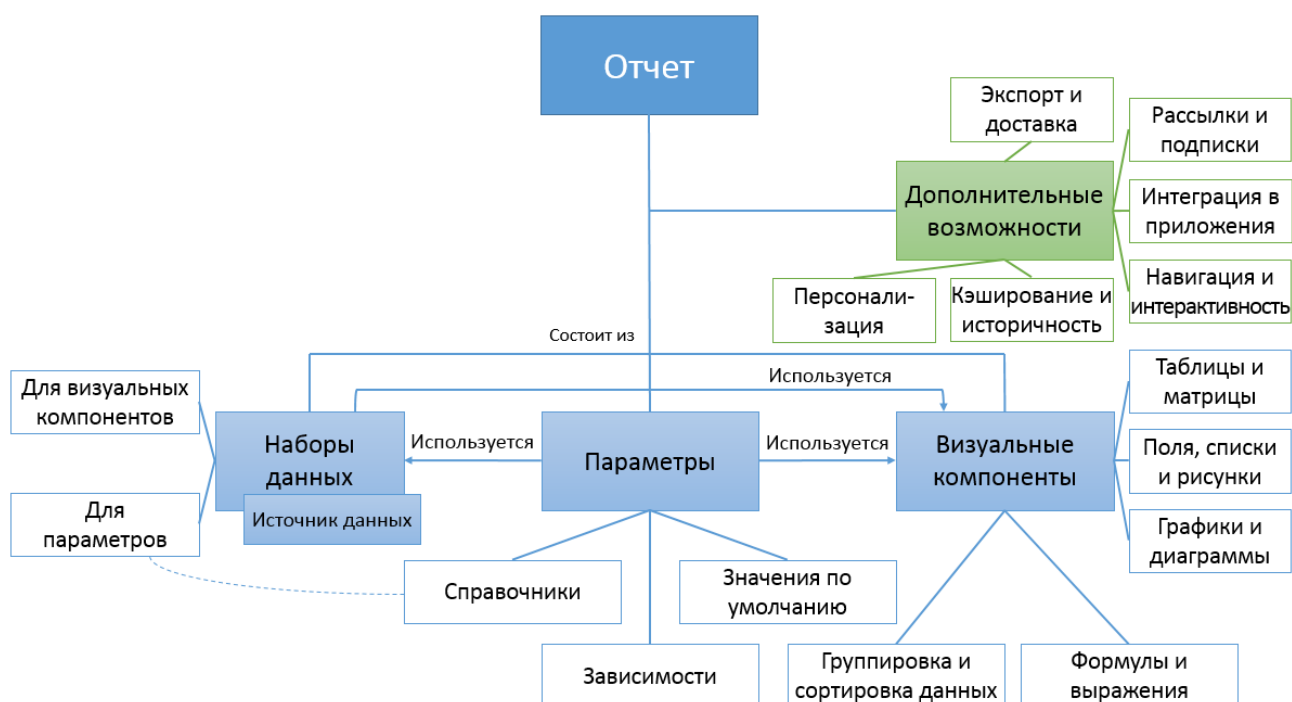
Отчет имеет несколько аспектов:

- *Источники данных* — база данных или веб-сервис, может быть нестандартное программное решение;
- *Наборы данных* — данные, получаемые из источников, могут иметь параметры для фильтрации при их чтении из источника данных;
- *Визуальный макет* — включает в себя визуальные компоненты (таблицы, графики и пр.), а также параметры фильтрации в самом отчете, без обращения к источнику данных, и интерактивность (возможность для пользователя переходить между визуальными представлениями и настраивать их для решения своих задач).

Отчеты по своему назначению бывают нескольких видов:

- *Электронный документ* — автоматизация создания документов по шаблону, например, служебное письмо, договор или счет для клиента;
- *Оперативный отчет* — представление фактов за небольшой период деятельности (дни и недели) в виде таблиц и графиков;
- *Сводный отчет* — аналогичный оперативному отчету, но за большой период деятельности (месяцы и годы) и с агрегацией (сжатием) данных по разным аспектам (заказчики, подразделения, временные периоды и пр.);
- *Аналитический отчет* (цифровая панель) — представление оценки деятельности за достаточно большой период деятельности (обычно месяцы и годы) с анализом факторов и прогнозированием, в основном с помощью диаграмм и графических компонентов;
- *Аналитическое приложение* — динамическое построение визуальных представлений на основе бизнес-правил. Отдельная большая тема, в данной статье не рассматривается.

Простые отчеты могут состоять из одного-двух визуальных компонентов (таблиц или графиков) и параметров для фильтрации и отображения данных. Соответственно, для каждого визуального компонента и параметров, использующих справочники, используются наборы данных, связанные с определенным источником данных. Визуальные компоненты могут использовать общий набор данных: например, график и таблица данных.



Сложные отчеты могут содержать множество стандартных (настраиваемых) и нестандартных (программируемых) визуальных компонентов и наборов данных из разных источников данных. Зачастую подготовка данных может быть самой сложной и трудоемкой работой при разработке отчетов.

Структура ТЗ

Техническое задание на разработку отчета включает в себя следующие разделы:

1. *Назначение отчета* — описываются роли пользователей, для которых разрабатывается отчет, их типовые задачи и ключевые вопросы, решаемые с помощью данного отчета;
2. *Варианты использования* — определяются способы использования отчета пользователями и технические условия;
3. *Источники и наборы данных, фильтрация и параметры;*
4. *Визуальный макет* — представление отчета для пользователя, сценарии использования (для интерактивных возможностей).

Формат технического задания согласовывается с разработчиком, поскольку этот документ разрабатывается для разработчика. Если аналитик и разработчик нашли общий язык, хорошо знают информационную систему заказчика, то формат можно предельно упрощать, т.е. многие «детали» оставлять на усмотрение разработчика или реализовать их стандартным способом.

Можно разработать отдельный документ, в котором определяются общие требования к отчетам, например, форматы представления разных типов данных, использование фонов и цветов, требования по производительности.

Назначение отчета

Раздел «Назначение отчета» важен для разработчика, поскольку позволяет ему лучше сфокусироваться на задаче и найти удобные для пользователя варианты

решений. Также благодаря этому разделу аналитик вовлекает разработчика в проект и обеспечивает понимание предметного контекста.

Отчеты могут разрабатываться в большом количестве, в средней компании их количество измеряется в десятках и сотнях. Если разработчик понимает контекст решения задач и знает полный набор отчетов, который нужно сделать, то это позволяет ему выработать общую инфраструктуру системы отчетов и общий подход. Это значительно экономит ресурсы и время на реализацию решений.

В таком случае аналитик может делать комплексное техническое задание на группу отчетов, решающих определенный класс задач, например, оперативный анализ продаж и т.п.

Описание ролей пользователей используется для настройки прав доступа при развертывании отчетов в информационной системе заказчика.

Варианты использования

Для понимания контекста использования отчета нужно указать, в каких рабочих процессах (пользовательских историях) он используется и какие предметные задачи решает.

Способы использования отчета:

- Как независимое решение в системе отчетов — у заказчика имеется система отчетности, работающая как отдельное приложение;
- Отчет, встроенный в приложение — требуется указать вид приложения (веб, настольное, мобильное) и способ интеграции по источнику данных и параметрам (данные предоставляет приложение или выполняются отдельные запросы к источникам данных);
- Доставка по электронной почте по расписанию — требуется контролировать конфиденциальность данных в таких отчетах, обычно используется для внешнего документооборота с контрагентами и клиентами.
- Доставка на рабочее место пользователя по расписанию — внутренняя регламентированная (стандартная) отчетность, также может использоваться для «тяжелых» отчетов.

Способы представления отчета:

- *Для печати* — официальные документы, которые необходимо фиксировать в бумажном виде, в этом случае от разработчика требуется четко задавать размеры и положение визуальных компонентов, чтобы они помещались на листе заданного формата;
- *Для просмотра* — такой отчет обычно просматривают на дисплее, может иметь адаптивный размер визуальных компонентов (в зависимости от данных), при печати может не помещаться по ширине на одну страницу.

Во многих случаях важно определить возможность и [форматы сохранения отчета](#) пользователем. Стандартные варианты – это выгрузка для чтения в Adobe Reader, либо для и редактирования в текстовых редакторах и электронных таблицах. Могут также использоваться технологические форматы, например, XML или текстовый файл с разделителями (CSV).

В службе отчетов также можно использовать серверные отчеты как интеграционные решения и источники данных для других приложений: веб-сервисы или [веб-каналы \(atom\)](#).

Для конфиденциальных данных, наоборот, возможность выгрузки и печати отчетов должна контролироваться и предоставляться только для пользователей с соответствующим уровнем доступа только с их рабочего места и в рабочее время. Любой доступ к данным через отчет должен протоколироваться. Это может потребовать разработки нестандартных решений с помощью программирования.

Источники и наборы данных

Для разработчика самый удобный источник данных – это база данных, позволяющая легко анализировать данные, тестировать разные варианты решений при их оптимизации и выполнять обработку данных любого уровня сложности. Такой вариант оптимальный для внутреннего источника (принадлежащего самому заказчику).

Если источник данных является внешним или есть ограничения по безопасности, то используются веб-сервисы или другие решения, но в этом случае у разработчика намного меньше пространство решений, производительность и гибкость решения будет зависеть в большей степени от внешних факторов.

Сложные отчеты могут иметь несколько источников данных. Если они разного типа и не согласованы между собой по идентификаторам и форматам данных, то может потребоваться разработка отдельного интеграционного решения или хранилища данных для набора отчетов.

Важно определить ответственность за анализ данных и выбор решения по их обработке. Если разработчик отчетов опытный и хорошо знает информационную систему заказчика, то можно и нужно делегировать ему право выбора технического решения по данным. В этом случае для аналитика описание данных упрощается.

Если задача по обработке является сложной, либо разработчик внешний, то в техническом задании требуется описание схемы данных (можно использовать имеющуюся документацию по базам данных). Также может потребоваться разработка отдельного технического задания по интеграционным решениям или хранилищу данных. В этом случае аналитику требуется поддержка архитектора и ведущего разработчика.

Результат запроса к источнику данных представляется как набор данных:

- *общий набор данных* — сохраняется в системе отчетов как отдельный невизуальный компонент и используется в нескольких отчетах;
- *внедренный набор данных* — сохраняется как компонент отчета и используется только в этом отчете.

Общие наборы данных используются при большом количестве отчетов. Также разработчики могут заранее создавать общие наборы данных, чтобы помочь пользователям самостоятельно создавать отчеты без использования SQL.

Сортировка и агрегация данных

Сортировка данных может выполняться как на стороне источника данных (может быть тяжелой операцией), так и в самом отчете. Рекомендуется делать ее в отчете, поскольку это снижает нагрузку на сервер базы данных и позволяет делать интерактивную сортировку, настраиваемую пользователем.

Агрегация данных также может выполняться в отчете, например, если в таблице используется многоуровневая группировка данных. Либо в диаграмме, которая автоматически выполняет требуемую агрегацию данных.

Одни и те же данные можно использовать как в оперативном отчете, который показывает детальную информацию о фактах, так и в аналитическом отчете, который на основе этих данных вычисляет показатели и итоги.

В плане производительности важно выбирать требуемый уровень детализации данных, т.е. использовать для аналитического отчета агрегированные (сжатые) на стороне источника данные, содержащие только необходимые показатели.

Фильтрация данных и параметры

Важным аспектом является фильтрация данных — это важно для получения требуемых данных. Лишние данные или отсутствие необходимых данных является критической ошибкой. Особенно это важно для оптимизации производительности решения.

Условия выборки данных могут быть постоянные для данного отчета, либо зависеть от [параметров отчета](#). Параметры могут быть скрытые, управляемые приложением, в котором используется отчет. Либо могут быть доступные для пользователя.

Кроме фильтрации данных на стороне источника данных параметры могут использоваться для фильтрации данных в самом отчете, без повторного запроса данных к источнику. Также с помощью параметров можно настраивать визуальное представление отчета: скрывать или показывать визуальные компоненты, менять формат представления данных и пр.

Параметр может быть обязательным либо не обязательным, имеет определенный тип данных (строка, число, дата и т.д.) и может иметь значение по умолчанию. Конкретный список типов данных зависит от используемой технологии разработки отчетов.

Для параметра можно указать список значений или связать его с набором данных, например, со справочником, и предоставлять пользователю выбор значения из него. Для этого нужно определить отдельный источник данных и создать набор данных для параметра, а также указать, какое поле данных используется как значение (используется в запросах и формулах), а какое поле – как наименование (показывается пользователю).

Параметры можно делать зависимыми от значения других параметров, например: сначала выбрать период продаж (простые параметры типа «Дата»), а затем за указанный период сформировать справочник сотрудников, создававших заказы на продажу за этот период.

Параметр со списком значений или справочником может быть *множественным* — позволять выбирать несколько значений.

Значение по умолчанию можно задавать в самом отчете как константу, либо устанавливать программным способом, с помощью SQL-запроса или формулы.

Для параметров некоторых типов данных (число, дата) можно задавать ограничение по диапазону значений.

Набор данных может содержать параметры для фильтрации данных пользователем или приложением. Если общий набор данных содержит параметры, то при его добавлении в отчет также будут добавлены эти параметры.

Визуальный макет

Обязательной частью постановки задачи для разработки отчета является визуальный макет, который вначале согласовывается с заказчиком и затем позволяет разработчику понять основные требования к отчету.

Аналитик может использовать электронную таблицу как метафору для собственного понимания и как инструмент разработки макета, поскольку общая логика отчетов с ней совпадает.

Макет страницы

Страница состоит из следующих разделов (в скобках указаны варианты наименований):

- *Заголовок (верхний колонтитул) отчета* — показывается в начале первой страницы отчета. Используется для вывода заголовка отчета и выбранных параметров.
- *Итог (нижний колонтитул) отчета* — показывается в конце последней страницы отчета. Используется для вывода общих итогов по отчету.
- *Верхний колонтитул страницы* — показывается в начале каждой страницы, может быть требование, чтобы на первой странице не показывался. Используется для вывода дополнительной информации, например, номера страницы, даты и время печати, сотрудник, создавший отчет, и прочее.
- *Нижний колонтитул страницы* — показывается в конце каждой страницы. Используется для вывода дополнительной информации и итогов по текущей странице.
- *Тело отчета* — основная часть, в которой выводятся визуальные компоненты, отображающие данные.

Можно разбивать страницы между разделами отчета. В некоторых визуальных компонентах (таблица, матрица или список) также могут определяться разделы с верхним и нижним колонтитулами для групп данных, разделителями страниц между этими группами.

Во многих технологиях разработки отчетов можно делать [вложенные отчеты](#), т.е. один отчет можно использовать как компонент для другого отчета. При этом в основной отчет добавляется только тело вложенного отчета.

Визуальные компоненты

Визуальный компонент ([область данных](#) в службе отчетов) отображает для пользователя набор данных. Также как и наборы данных, визуальные компоненты могут быть общими (использоваться в разных отчетах) либо внедренными, используемыми только в данном отчете.

[Общие визуальные компоненты](#) позволяют реализовать стандартное представление данных в разных отчетах и обеспечить для пользователей создание отчетов в режиме конструктора.

Также могут использоваться простые визуальные компоненты — элементы, отображающие значения, такие как [текстовые поля](#), линии, фигуры и изображения.

Текст или рисунок могут читать данные из источника или представлять данные, встроенные в сам отчет.

Для задания значений данных и свойств визуальных компонентов можно использовать [выражения](#) — настраивать правила визуализации или рассчитать производное значение на основе исходных данных. Например, задать цвет фона в зависимости от значения или посчитать рентабельность на основе расходов и доходов за определенный период.

Визуальные компоненты можно динамически показывать или скрывать, в зависимости от параметров, данных или состояния других визуальных компонентов.

Типы визуальных компонентов:

- *Информационный* — детальное отображение данных в виде структурированного набора значений;
- *Аналитический* — обобщенное графическое отображение данных.

Информационные визуальные компоненты:

- [Таблица](#) (сетка) — отображает множество структурированных данных, имеет фиксированное количество колонок и переменное количество строк (в зависимости от данных), например, список продаж с набором атрибутов определенного типа данных (наименование заказчика, номер и дата заказа, наименование и количество проданного товара и т.д.). Ограничение: фиксированное количество колонок, данные представляются только как строки.
- [Матрица](#) — отображает свернутое множество данных (аналог PIVOT в SQL), имеет переменное количество колонок и строк, например, по строкам можно вывести заказчиков, а по колонкам год, и в ячейках вывести сумму продаж. Ограничение: не позволяет задавать несколько фиксированных колонок.
- [Табликс](#) (гибрид таблицы и матрицы) — сочетает фиксированные и переменные колонки, получается таблица с переменным числом колонок. Ограничение: данные представляются только как строки.
- [Список](#) — произвольное расположение полей. Ограничение: сложность создания визуальной структуры данных.

В этих компонентах можно настроить сортировку и сделать ее интерактивной — позволить пользователю менять порядок данных при просмотре отчета. Также можно задавать [группировку данных](#) с итогами на каждом уровне данных, например, сумма продаж по регионам, заказчикам и заказам.

Для матрицы или табликса можно также делать группировку данных по колонкам, например, сделать уровни – год, квартал и месяц.

Группировка данных может быть интерактивной, т.е. пользователь при просмотре отчета может переключать уровень детализации данных.

В информационные компоненты можно добавлять аналитические компоненты — [спарклайны и гистограммы](#).

Аналитические визуальные компоненты обеспечивают наглядное представление числовых данных:

- **Диаграмма** — представление с помощью графических компонентов (точки, линии, прямоугольники, регионы), связанные с аналитическими измерениями (справочниками) и имеющие одну или несколько числовых шкал;
- **Датчик (индикатор)** — представление показателя с оценкой текущего значения относительно планового значения, с возможностью отобразить шкалу с минимумом и максимумом. Могут быть **вложены в таблицы или матрицы**;
- **Карта** — представление показателей, связанных с географическими данными (страна, регион, город и т.д.).

Помимо стандартных визуальных компонентов, можно разрабатывать собственные либо приобретать готовые с требуемыми возможностями.

Пример макета для простого отчета

Важно не усложнять техническую документацию и делать ее максимально понятной и удобной для использования. Для простых отчетов можно использовать упрощенный формат постановки. Данные при этом могут быть описаны прямо в визуальном макете, пример:

Обязательные параметры:

- Начало периода - по умолчанию начало текущего месяца (только дата, без времени);
- Окончание периода - по умолчанию текущая дата (только дата, без времени).

Необязательные параметры:

- Регион - выбор из справочника территорий, упорядоченный по наименованию, показываются только регионы, в которых были продажи за указанный период;
- Продавец - выбор из справочника сотрудников, упорядоченный по полному имени (ФИО), показываются только сотрудники, которые регистрировали заказы за указанный период и выбранным регионам.

Заголовок отчета

Продажи по заказчикам с {Параметр.Начало периода} по {Параметр.Окончание периода}

Подзаголовок (строка отображается, если выбрано значение в соответствующем параметре)

Регион: {Параметр.Регион.Наименование}

Продавец: {Параметр.Сотрудник.Полное имя}

Таблица данных

Заказчик	Заказ	Товар	К-во	Сумма
{Заказчик.Наименование}			{Сумма по заказчику}	{Сумма по заказчику}
	{Заказ.Номер} от {Заказ.Дата регистрации}		{Сумма по заказу}	{Сумма по заказу}
		{Товар.Наименование}	{Номенклатура заказа.Количество}	{Номенклатура заказа.Количество} * {Номенклатура заказа.Цена}
Итого			{Общая сумма}	{Общая сумма}

Нижний колонтитул

{Отчет.Номер страницы} из {Отчет.Количество страниц}

{Отчет.Дата и время создания}

Если разработчик хорошо знает источник данных, то по данной постановке он сможет разработать требуемый отчет. Также может потребоваться добавить описание дополнительных требований заказчика, например, выделять цветом сумму продаж, если она превышает определенный порог.

Навигация и интерактивность

Для аналитических отчетов либо отчетов с большим объемом данных полезными функциями являются: возможность для пользователя настраивать нужное представление, быстро переходить между разделами отчета, переходить между отчетами с автоматической передачей параметров по контексту данных, либо запускать выполнение программной функции.

Для навигации по большому отчету можно использовать [закладки и гиперссылки](#) как на веб-страницах или делать [схему \(карту\) документа](#) — боковую навигационную панель, связанную с таблицей либо матрицей, и автоматически отображающую уровни группировки в связанном визуальном компоненте в виде визуальной иерархии («дерева»).

В самой таблице либо матрице при использовании группировки данных можно скрывать нижележащие уровни с помощью [переключателей видимости](#). Аналогично можно скрывать и показывать разделы отчета.

Для анализа данных полезной является [интерактивная сортировка](#), чтобы пользователь мог представить данные в нужном порядке по выбранному показателю или значению.

[Гиперссылки](#) могут содержать собственное значение, либо можно их привязывать к данным, и по ним пользователь может открывать другие отчеты, веб-страницы, либо запускать действия с помощью вызова методов веб-сервисов.

Навигация между отчетами обычно строится от общего (сводного или аналитического отчета) к частному (электронному документу либо оперативному отчету). Но при необходимости можно строить любые связи, в т.ч. и с внешними информационными системами.

Дополнительные возможности

Современные инструменты разработки отчетов становятся мощными конструкторами веб-приложений, позволяющими быстро создавать и [публиковать](#) отчеты. При необходимости можно создавать (программировать) собственные компоненты, расширяющие стандартные возможности.

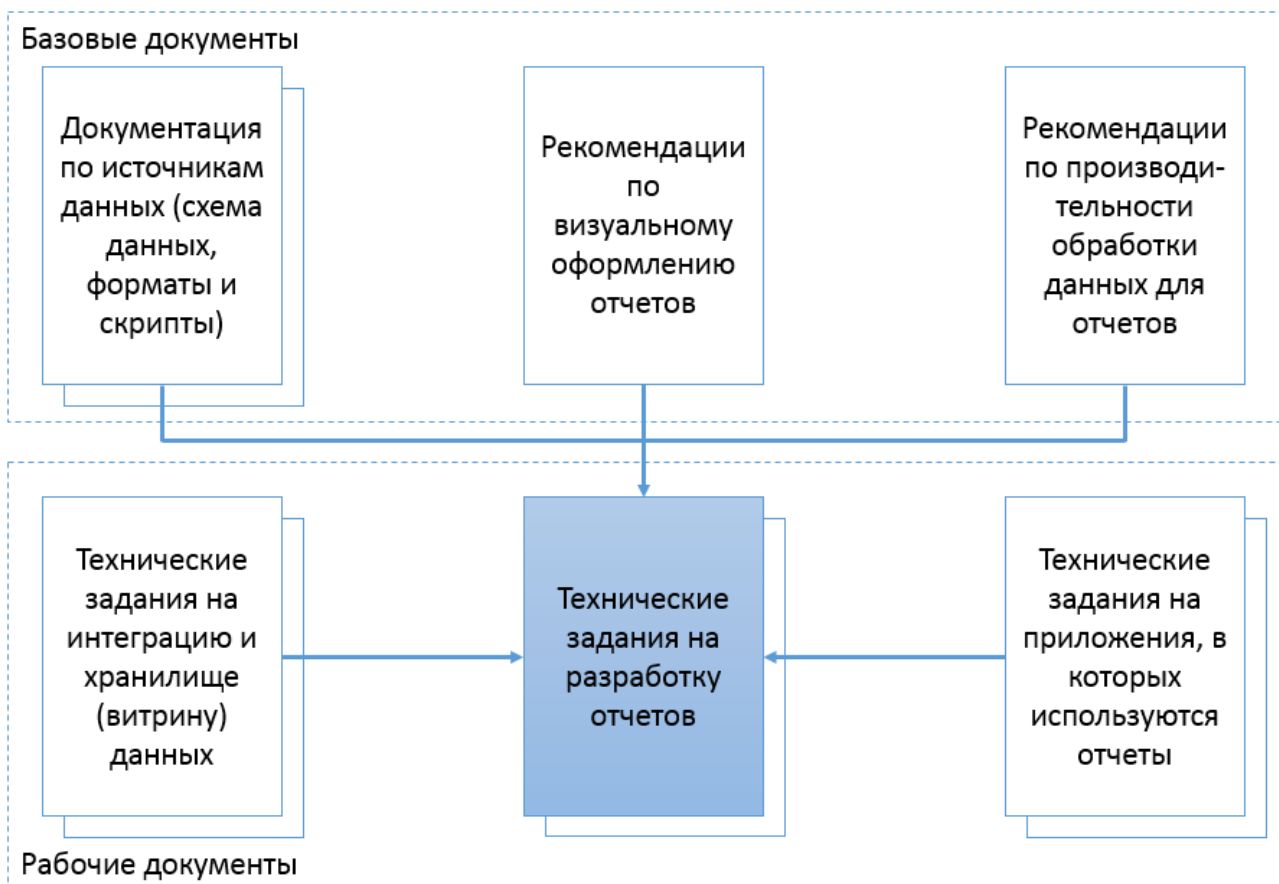
В зависимости от используемой технологии разработки отчетов может быть различный список дополнительных возможностей:

- [Кеширование отчетов](#) — используется для оптимизации выполнения «тяжелых» отчетов, позволяя выполнять их обработку заранее, в момент минимальной нагрузки на систему, и быстро получать готовый отчет для пользователя.
- [Историчность \(моментальный снимок\)](#) — сохранение отчетов на определенный момент времени для обеспечения возможности проверок и сравнения данных.
- [Доставка отчетов по расписанию](#) — отправка отчетов в определенный момент времени по электронной почте или на рабочие места сотрудников. Возможность сотрудников [подписаться](#) на важные для них события (изменения данных) для получения соответствующего отчета по этим событиям.
- [Персонализация](#) — возможность для пользователя настроить для себя личную папку и сохранить в ней важные для себя отчеты.

Система документации

Если отчет разрабатывается в рамках проекта, то может потребоваться разработка базовой документации, в которой описываются источники данных. Общие рекомендации по визуальному оформлению и производительности обработки данных также можно оформить в виде отдельных документов.

Это требует сотрудничества с архитектором, системным аналитиком, разработчиком и графическим дизайнером, которые разрабатывают эту документацию.



Если для отчетов требуется разработка интеграционных решений, хранилища или витрины данных, то также требуется разработать технические задания на эти решения.

Если отчеты используются в приложении, то сначала разрабатывается техническое задание на приложение, а задание на разработку отчета является его частью.

*Сергей Минюров
Москва, май 2018*