

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Информационные системы и технологии корпоративного управления
(направленность (профиль))

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему «Исследование технологий создания отчетных документов из
информационных систем, работающих через web интерфейс»

Студент А.С. Ивненко
(И.О. Фамилия) (личная подпись)

Научный
руководитель А.Б. Кузьмичев
(И.О. Фамилия) (личная подпись)

Руководитель программы д.т.н., доцент, С.В. Мкртычев
(ученая степень, звание, И.О. Фамилия) (личная подпись)
« ____ » _____ 20 ____ г.

Допустить к защите
Заведующий кафедрой к.т.н., доцент, А.В. Очеповский
(ученая степень, звание, И.О. Фамилия) (личная подпись)
« ____ » _____ 20 ____ г.

Тольятти 2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1 АНАЛИЗ СУЩЕСТВУЮЩЕЙ ПРОБЛЕМЫ СОЗДАНИЯ ОТЧЕТОВ	6
1.1 Анализ проблем использования генераторов отчетов	6
1.2 Анализ процесса разработки/изменения отчета	12
1.3 Определение масштаба (локализация) выявленной проблемы создания отчетов	16
ГЛАВА 2 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ И ТЕХНОЛОГИЙ СОЗДАНИЯ ОТЧЕТНЫХ ДОКУМЕНТОВ	18
2.1 Аналитический обзор встроенных report-инструментов	19
2.2 Анализ генераторов отчетов, интегрируемых в информационные системы	25
2.3 Анализ низкоуровневых подходов к созданию генераторов отчетов	32
2.4 Анализ альтернативных решений проблемы, описанных в общедоступных источниках	34
ГЛАВА 3 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ТЕХНИЧЕСКОГО РЕШЕНИЯ НА ОСНОВЕ ОПТИМИЗИРОВАННОГО ПРОЦЕССА РАЗРАБОТКИ ОТЧЕТОВ	38
3.1 Предварительный бизнес-анализ и проектирование процесса разработки отчета	38
3.2 Разработка вариантов использования ПО создания отчетов	47
3.3 Формулирование требований к ПО создания отчетов	49
3.4 Реализация программного решения для разработки отчетов	53
ГЛАВА 4 АНАЛИЗ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ И ПОЛУЧЕННОГО ПРОГРАММНОГО РЕШЕНИЯ РАЗРАБОТКИ ОТЧЕТОВ	70
4.1 Анализ результатов применения оптимизированного процесса по разработке отчетов	70
4.2 Анализ результатов и перспектив реализованного программного решения для разработки отчетов	73
4.3 Анализ результативности проведенного исследования	83
ЗАКЛЮЧЕНИЕ	90
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	92

ВВЕДЕНИЕ

Актуальность темы исследования: важной частью функционала многих информационных систем является процесс формирования отчетных документов. Процесс подразумевает использование документа-шаблона для заполнения его данными. В ручном режиме процесс заполнения документа занимает много рабочего времени. Поэтому многие компании стремятся к его автоматизации, особенно в ситуации повторяющейся или систематической необходимости в актуализации формы отчета.

Универсальные инструменты подготовки отчетных документов подразумевают продолжительный процесс разработки отчетов с привлечением квалификационных работников (программистов). Однако истинной потребностью бизнес-пользователей является возможность *самостоятельной* оперативной разработки и/или доработки шаблонов отчетов.

Отыскание оптимального собственного технологического решения, удовлетворяющего противоречивым требованиям, является актуальной темой по причине нетривиальности и составляет суть данной исследовательской работы.

Научная проблема: констрадиктация научной проблемы проявляется в том, что для решения реальных практических задач применение универсальных report-инструментов видится необоснованным, а разработка собственного адаптированного инструмента сопряжено с решением изобретательских и оптимизационных задач, основанных на комплексном анализе процесса подготовки отчетных документов.

Исследуемая в диссертации проблема находится структурно на уровень выше обычного обсуждения требований, а значит, обычно не возникает цели ее исследовать и решить. Во многих организациях может вообще отсутствовать принципиальная возможность преодоления рассматриваемой ситуации. Кроме этого, представляется экономически нецелесообразным подходом осуществлять повсеместное самостоятельное проведение таких исследовательско-практических работ в каждой отдельной организации.

Финитизация данной научной проблемы заключается в предполагаемом экспоненциальном росте эффективности труда специалистов отдельных предметных областей, сталкивающихся в своей работе с частой необходимостью формирования отчетных документов, основанных на массивах данных. Достижение данного эффекта предсказуемо и является очевидным результатом обретения функциональной независимости предметного специалиста, который при позитивном исходе исследования проблемы и ее решения, получит ранее недоступные возможности.

Цель исследования: анализ доступных технологий, процессов, их технологической совместимости и вариантов комбинации, а также разработка технического решения, позволяющего автоматизировать, упростить и удешевить процесс создание нового отчетного документа или редактирования существующего. Определить инновационный процесс работы с созданным инструментом, роли и границу ответственности участников.

Область исследования: прикладные технологии и алгоритмы создания отчетных документов. Программное обеспечение смежных областей, библиотеки (API) генерации отчета. Организационные предпосылки бизнес-среды, влияющие на проявление проблемы.

Гипотеза: Процесс создания нового отчетного документа или редактирование существующего будет удешевлен и упрощен, если будут решены следующие задачи:

- Проанализированы существующие подходы и технологии создания отчетных документов, а также смежные технологии.
- Изучены авторитетные источники в исследуемой области знания.
- Определены основные требования к искомому техническому решению.
- Сформулированы варианты решения.
- Выбран оптимальный вариант решения проблемы, реализовано соответствующее ему программное обеспечение.

Задачи исследования: выявить проблемы создания, модернизации и поддержания отчетов в актуальном состоянии.

Проанализировать существующие подходы и решения, установить признаки их несоответствия потребностям заинтересованных лиц.

Практическая значимость проявится:

- при анализе целесообразности использования нового подхода в разрезе экономической эффективности;
- при оптимизации деятельности предметных специалистов, нуждающихся в самостоятельном режиме использования report-инструментария;
- в устранении обычной организационной инертности, связанной с привлечением программистов (негативно влияет очередность выполнения задач, необходимость предварительного формулирования задачи и контроля корректности исполнения).

Данное исследование принесет пользу в отыскании обоснованно-простого решения исследуемой проблемы, что позволит:

- заинтересовать сообщество профессионалов и энтузиастов развивать данный опыт и направление знаний в целях исключения необоснованной сложности при разработке информационных систем;
- мотивировать IT-предпринимателей активнее заниматься изыскательской деятельностью с последующей разработкой и продвижением новых нишевых программных продуктов для широкого круга пользователей.

Объем и структура диссертации: диссертационное исследование состоит из введения, 4 глав, заключения. Работа изложена на 95 страницах, содержит 30 рисунков, 23 таблиц.

ГЛАВА 1 АНАЛИЗ СУЩЕСТВУЮЩЕЙ ПРОБЛЕМЫ СОЗДАНИЯ ОТЧЕТОВ

1.1 Анализ проблем использования генераторов отчетов

В настоящее время востребовано, но не найдено решение в виде report-инструмента, лишенного практических недостатков с точки зрения эксплуатационной доступности для различных предметных специалистов и характеризующегося применимостью в информационных системах различного уровня сложности и ориентированностью на различные группы потребителей.

Например, в настоящее время в корпоративном секторе рынка IT-решений имеется ситуация, характеризующаяся превышением потребности компании в создании/изменении отчетных форм над возможностями (ресурсами) приданных IT-подразделений.

Проблема имеет несколько проявлений, но на первый план выходит потребность систематической разработки отчетов, не подкрепленная реальными организационными возможностями в виде доступности ресурсов (исполнителей). Выбранные инструменты разработки отчетов полностью исключают независимую разработку самими бизнес-акторами в то время, как постоянно растут потребности в генерации разных отчетных документов. Однако истинные проблемы могут быть не видны в первом приближении, и требуется применение системного анализа проблем исследуемых бизнес-процессов.

Анализ проблем и их коренных причин проведем в соответствии с соответствием с прогрессивными практиками lean-development: А3-анализ, диаграмма Исикавы (рыбий скелет)[27] и “5 почему”. Поиск коренных причин с помощью Диаграмма Исикавы представлен на рисунке 1.



Рисунок 1 – Поиск коренных причин с помощью Диаграмма Исикавы

Проблема заключается в том, что менеджеры процессов испытывают регулярную потребность в разработке отчетных документов, однако систематически сталкиваются с невозможностью динамичного исполнения поручений ИТ-специалистами, а самостоятельно разработать или доработать очередной отчет не представляется возможным.

Это важно, т.к. в современной бизнес-среде для достижения конкурентных преимуществ требуется интенсивная и качественная аналитическая работа на любом уровне. Каждый менеджер сталкивается с необходимостью получать либо представлять информацию в разных аналитических разрезах.

Каждый соприкоснувшийся должен проникнуться проблемой и захотеть участвовать в улучшениях. Тогда каждый менеджер процесса профессионально заинтересован и реально заботится как о личной эффективности, так и о результирующей. А вместо целевой аналитической (и другой) деятельности каждый из них сталкивается с рутинными мероприятиями по преодолению проявлений описываемой проблемы. Несмотря на кажущуюся неочевидность ее причин, при направленном анализе они обнаруживаются явно. Главное то, что должен быть концептуальный способ решить проблему.

В настоящий момент это работает таким образом, что менеджер процесса в процессе работы сталкивается с потребностью разработать отчет для того, чтоб потом сформировать его.

1. Менеджер понимает, что по ряду причин не может этого сделать самостоятельно. Формулирует задачу, в т.ч. форму отчета.
2. Обращается к ИТ-специалисту (программисту) с задачей по разработке/изменению отчета.
3. Программист анализирует требования к задаче, взаимодействует с Менеджером в целях ее уточнения.
4. Программист разрабатывает дизайн отчета, настраивает источник данных, пишет программный код.

5. Программист тестирует работоспособность, исправляет ошибки, сдает выполненную задачу постановщику.

6. Менеджер принимает работу, выявляет возможные ошибки, а в случае выявления ошибок возвращается на доработку до полного устранения ошибок.

7. Программист в случае возврата непринятой задачи устраняет ошибки и повторно сдает задачу постановщику.

8. Менеджер проверяет и окончательно принимает задачу.

Проблема с точки зрения Lean-Development проявляется в том, что шаги с 3 по 8 не создают дополнительной ценности продукту и являются непроизводительными потерями:

– Этап 3: обращение к ИТ-специалисту осложняется тем, что менеджер чаще всего не владеет навыками четкого описания требований;

– Этап 4: программист, не являющийся экспертом в предметной области, может долго разбираться с задачей и даже понять неправильно;

– Этап 5: программист повторяет в специальном дизайнера форму отчета (на основе предложенного Менеджером файла в популярном формате), выполняет прочие технические действия, которые не являются для потребителя самоценными;

– Этап 6: сложный инструментарий в том или ином виде требует затрат на тестирование и исправления;

– Этап 7: необходимость двухсторонней проверки результата заказчиком вызваны тем, что исполнитель и постановщик - разные актор;

– Этап 8: возможно зацикливание проверок и доработок.

Проблема носит измеряемый характер, т.к. затраты ресурса (времени, денег) можно измерить. Судя по примерной оценке, полезное использование ресурса составляет 1/3 от фактических затрат.

В результате мероприятий по решению изложенных проблем получим повышение профессиональной продуктивности Менеджера. Сокращение затрат

времени до уровня, не превышающего суммарные затраты на шаги 1, 2, 8. Потому, что шаги с 3-7 не являются самоценными для заинтересованного в разработке отчета лица. Представляется, что шаги 1, 2, 8 являются минимально важными для получения разработанного отчета.

По итогам ожидается изменение реальных метрик:

- Кратное сокращение затрат времени, а значит и связанное с этим количество денежных средств на оплату труда.
- Высвобождение время может повлечь условное кратное увеличение продуктивности.

Таким образом, проблемы обусловлены следующими первопричинами:

- выбором (предопределенным) используемого в организации report-инструментария;
- отсутствием у менеджеров специальных ИТ-компетенций, позволяющих объективно судить об обоснованности выбора report-инструмента;
- отсутствием известных альтернативных report-инструментов, способных решить проблему;
- корпоративными регламентами доступа к данным, которые никто не берется оспаривать;
- обычным отсутствием нацеленности на максимальную оптимизацию таких атомарных процессов в организации, как разработка нового отчета или изменение имеющегося.

По результатам анализа стало очевидно то, что каждая из идентифицированных проблем связана с наличием в процессе разработки отчетов этапов 3-7, а значит, радикальным решением проблем было бы отсечение лишних этапов. То есть, область решения ключевой проблемы исследования находится исключительно в области оптимизации процесса, вывод о чем подкрепляется анализом истинных потребностей, сведенных в таблице 1.

Таблица 1 – Анализ типовых бизнес-потребностей по созданию отчетов

№	Бизнес-потребность	Анализ
1.	Возможность полностью самостоятельной разработки отчетов бизнес-акторами.	<p>Бизнес-акторы ограничены в своей практике подготовки отчетов различными факторами:</p> <ul style="list-style-type: none"> - недостаточными ресурсами труда программистов; - технологической сложностью классических инструментов; - организационными рабочими регламентами взаимодействия между подразделениями.
2.	Требуется сокращение суммарных затрат времени на разработку каждого отчета.	<p>Современные условия труда характеризуются высокой интенсивностью, экспрессией, потребностью в качественном анализе и креативностью. Такие условия обусловили потребность в демонстрации состава и результатов анализа, что лучше всего достигается с применением отчетных документов. Кроме этого сохраняются классические наработанные приемы использования отчетных документов: заполнение форм учетных документов, списки, графики, планы и т.д.</p>
3.	Требуется отсутствие функциональных ограничений.	<p>В зависимости от специфики конкретной организации и сферы деятельности бизнес-актора требуются отчеты любой сложности, которая может быть обусловлена:</p> <ul style="list-style-type: none"> - обилием аналитических разрезов; - количеством источников данных; - гаммой востребованных представлений и интерпретаций отображаемой информации; - дизайнерскими и стилистическими потребностями - и т.д.

Таким образом, был проведен предварительный анализ и идентификация главной проблемы и её первопричин, что позволит провести направленную исследовательскую работу.

1.2 Анализ процесса разработки/изменения отчета

В целях проведения анализа текущего процесса разработки нового отчета или доработки существующего отчета построим диаграмму прецедентов, которая позволит визуальнo представить основную и альтернативные последовательности. Диаграмма представлена на рисунке 2.

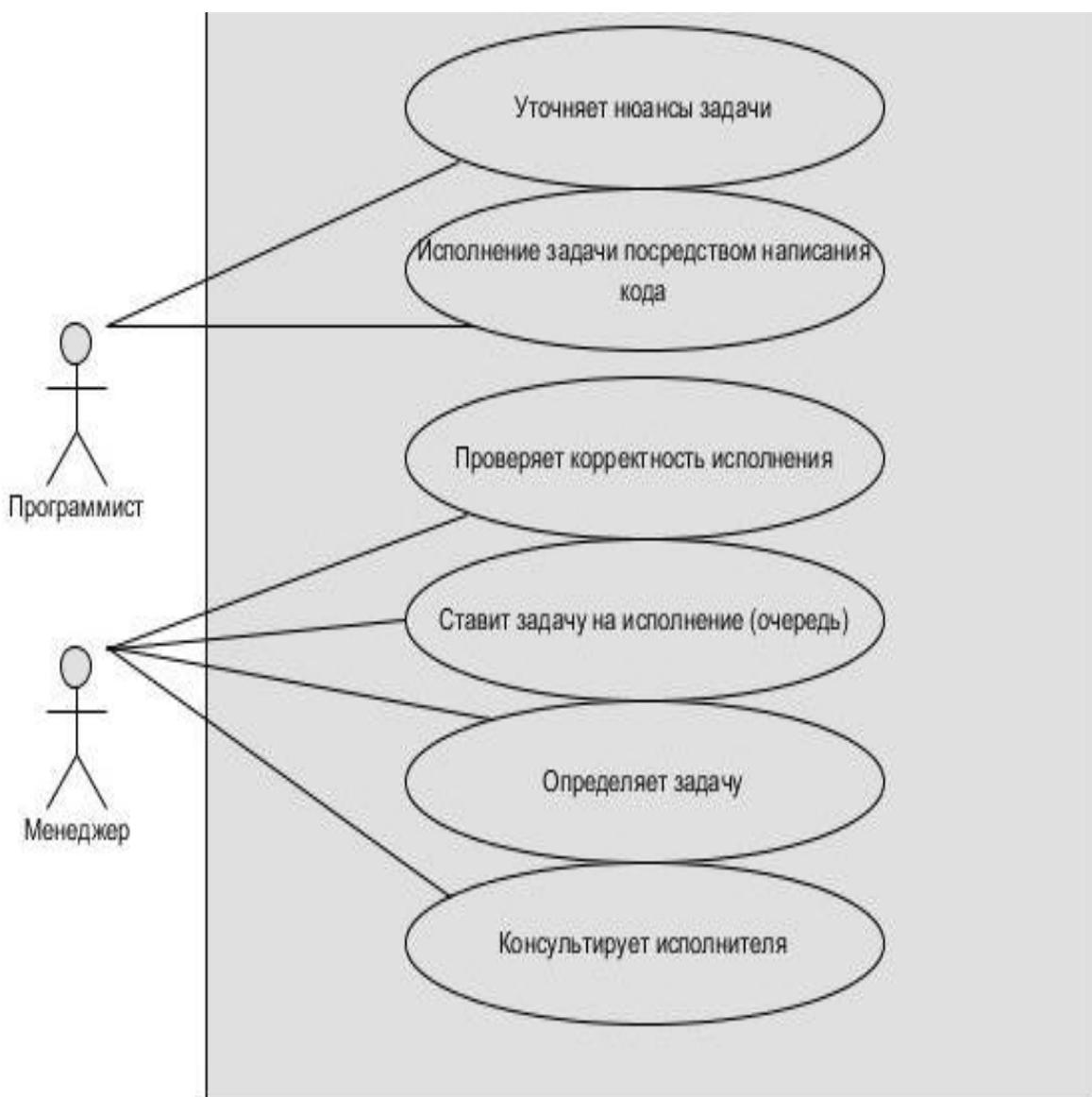


Рисунок 2 – Use-case диаграмма текущего процесса

Опишем в виде таблицы участников процесса – акторов.

Таблица 2 – Краткое описание акторов диаграммы

Актор	Краткое описание
Программист	Квалифицированный специалист выполняющий техническую роль по реализации отчетов.
Менеджер	Актор, обладающий знаниями в предметной области и бизнес-процессах, но не разбирающийся в ИТ.

Опишем детально подробности прецедентов в виде спецификации.

Таблица 3 – Прецедент: Определяет задачу

Прецедент: Определяет задачу
Id: 1
Краткое описание: определяет задачу путем формирования ТЗ на разработку отчета.
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: наличие потребности в новом отчете или в изменении готового.
Основной поток: Документирует требования и готовит макет отчета
Постусловия: ТЗ подготовлено
Альтернативные потоки: уточняет требования

Таблица 4 – Прецедент: Ставит задачу на исполнение (очередь)

Прецедент: Ставит задачу на исполнение (очередь)
Id: 2
Краткое описание: обращается регламентированным способом в ИТ-службу
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: Имеется готового ТЗ.
Основной поток: передает описание задачи и ТЗ

Постусловия: обращение зарегистрировано
Альтернативные потоки: если получен отказ в виду емкости задачи, то упрощает требования в ТЗ

Таблица 5 – Прецедент: Консультирует исполнителя

Прецедент: Консультирует исполнителя
Id: 3
Краткое описание: Проводит консультирование исполнителя.
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: получен список вопросов по ТЗ
Основной поток: вербально или письменно коммуницирует с исполнителем
Постусловия: на все вопросы даны исчерпывающие ответы
Альтернативные потоки: если хоть на 1 вопрос не дан исчерпывающий ответ, то ТЗ возвращается на доработку.

Таблица 6 – Прецедент: Проверяет корректность исполнения

Прецедент: Проверяет корректность исполнения
Id: 4
Краткое описание: принимает работу.
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: Программист выполнил поставленную задачу по разработке отчета и сообщил об этом заказчику
Основной поток: Проводит тестирование в соответствии с бизнес-критериями корректности. Сообщает о приемке задачи.
Постусловия: Результат выполнения задачи соответствует поставленному описанию задачи и ТЗ
Альтернативные потоки: Если выявляются ошибки и несоответствие ТЗ, то задача возвращается исполнителю на доработку.

Таблица 7 – Прецедент: Уточняет нюансы задачи

Прецедент: Уточняет нюансы задачи
Id: 5

Краткое описание: уточняет требования к отчету
Главные актеры: Программист
Второстепенные актеры: отсутствуют
Предусловия: Задача описана, но описание не в полном объеме понято исполнителем.
Основной поток: задает вопросы менеджеру, получает ответы.
Постусловия: Программист полностью понял задачу
Альтернативные потоки: если ответ хоть на один вопрос не получен, то возвращает задачу менеджеру на доработку.

Таблица 8 – Прецедент: Исполнение задачи посредством написания кода

Прецедент: Исполнение задачи посредством написания кода
Id: 6
Краткое описание: Программист реализует поставленную задачу
Главные актеры: Программист
Второстепенные актеры: отсутствуют
Предусловия: задача понятна исполнителю
Основной поток: разрабатывает дизайн, пишет программный код, тестирует результат.
Постусловия: Задача выполнена.
Альтернативные потоки: нет

Диаграмма прецедентов и спецификация к ней позволили выявить акторов процесса, определить необходимость коммуникаций между актерами, а также возможность заикливания бизнес-процесса из-за потенциального отсутствия единого языка между актерами в каждой рассматриваемой предметной области. Такое положение вещей означает наличие проблемы процесса, связанной с непредсказуемостью затрат времени на реализацию каждого конкретного отчета.

1.3 Определение масштаба (локализация) выявленной проблемы создания отчетов

В целях достижения возможности решения проблемы проведем ее локализацию, т.е. определим ее масштабы и сферу распространения.

Так, на примере компании «Квартплата 24» можно констатировать осознанность данной проблемы ввиду огромного количества клиентов, имеющих свои специфические требования к формированию отчетных документов. При этом ресурсы компании ограничены, что приводит к невозможности реализовывать привычными способами все пожелания клиентов по модернизации отчетов.

Также данная проблема актуальна и для Тольяттинского Государственного Университета в связи со следующим.

В России в настоящее время на любой ступени образования существуют определенные стандарты, которые носят название ФГОС (Федеральный образовательный стандарт). В соответствии с этими стандартами, образовательные организации должны осуществлять образовательную в строгом соответствии с положениями этих документов[11].

Контроль же над этим осуществляет вышестоящий орган – министерство образования, который с определенной периодичностью может потребовать разного рода отчеты. Некоторые из них могут формироваться на базе уже существующих в учреждении программ, но не все.

Источником проблемы является сложность классических инструментов разработки отчетов, т.к. в подавляющем большинстве случаев специалистам, ответственным за отчеты необходима помощь программистов. Программисты, используя специальный инструментарий, индивидуально создают шаблоны отчетов, настраивают для них источники данных, пишут программный код, если требуется дополнительная сложная логика, выходящая за рамки возможностей SQL.

Также не обошла эта потребность (формирования разного рода отчетной документации) и любое предприятие, организацию, любой бизнес, в которых

функционируют определенные процессы. Именно в целях улучшения статистической и оперативной управляемости процессов в организациях присутствует постоянная потребность в формировании отчетов. Учитывая современные условия деловой деятельности, необходимо отметить, что в постоянно меняющейся конкурентной среде возникают новые цели и задачи, модернизируются процессы, а значит, появляются новые метрики, ключевые показатели эффективности. Это неизбежно несет постоянные изменения в формы отчетных документов. Поэтому постоянное привлечение к работе квалифицированного программиста становится не целесообразным.

Однако следует отметить, что в большинстве случаев программисты используют на практике небольшой достаточно стандартный перечень типовых готовых решений при разработке отчетов. Общим их свойством является непригодность для динамической самостоятельной работы бизнес-акторов, т. к. целью создания такого инструментария являлось удовлетворение потребностей корпораций и крупных производств с устоявшимися бизнес-процессами.

Необходимо отметить, что в процессе профессиональной подготовки программисты зачастую не фокусируют свое внимание на уместности применения тех или иных технологий. Поэтому, при найме на работу таких сотрудников, организация рискует получить специалиста, размножающего решения, неадекватные задачам.

Таким образом, приходим к выводу, что проблема известна, но комплекс ее причин не является всегда осознанным в рамках конкретной организации. В такой ситуации важно подчеркнуть необходимость осознанного выбора применяемых технологий для разработки отчетов.

ГЛАВА 2 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ И ТЕХНОЛОГИЙ СОЗДАНИЯ ОТЧЕТНЫХ ДОКУМЕНТОВ

В целях осуществления осознанного выбора технологий, пригодных для реализации задач формирования отчетных документов, проведем исследование функционала общеизвестных инструментальных средств по подготовке отчетов.

Представляется интересным провести классификацию инструментов, определить полезные особенности, которые могут пригодиться для разработки собственного решения.

Очевидными классификационными группами для анализа являются:

- Встроенные report-инструменты – это функционал по подготовке отчетов, который является неотъемлемой частью некоторой основной информационной системы;
- Интегрируемые report-инструменты – это универсальные программные модули, которые могут быть встроены в информационные системы;
- Отдельные технологии, например низкоуровневые программные библиотеки, RESTful-API и так далее, характеризующие узкой специализацией, которые в определенной комбинации позволяют разработать низкоуровневое программное решение.

Анализ встроенных report-инструментов интересен исключительно оригинальными идеями реализации пользовательского интерфейса в части процесса настройки импорта данных в отчет. Для целей анализа интересны, например, платформенный инструментарий (1С), а также специализированные решения в составе конкретных web-сервисов (Мегаплан, Redmine, Zendesk).

При анализе интегрируемых report-инструментов есть вероятность обнаружить уже существующие эффективное программное решение для получения требуемого результата, означающего решение проблемы. Типичными представителями этой группы являются такие решения, как:

FastReport, Stimulsoft Reports.Web и т.д., которые характеризуются наличием визуального редактора шаблонов с web-интерфейсом.

Анализ отдельных технологий позволяет с большей вероятностью решить проблему, исключив наследование нежелательных особенностей известных высокоуровневых решений. Под отдельными технологиями подразумеваются web-api таких продуктов, как: Google Docs, Google Sheets, Google Drive, Microsoft Office-Online, myoffice.com и т.д.. Дополнительно к этой группе можно отнести такую библиотеку, как Apache POI.

2.1 Аналитический обзор встроенных report-инструментов

Проведём анализ функционала инструмента 1С. Конфигуратор.

1С. Конфигуратор встроенный инструмент разработки на платформе 1С в том числе для разработки отчетных документов [31].

В 1С реализовано 3 способа создания отчетов. Способы разработки отчетов (всего 3)[13]:

- С помощью написания кода. Существует возможность разработать отчет полностью исключительно с помощью написания программного кода.
- С помощью СКД (система компоновки данных) (рисунок 3)
- С помощью конструктора запроса с обработкой данных (рисунок 2)

Этапы разработки отчетов с применением перечисленных выше подходов представлены на соответствующих диаграммах. Основные отличия подходов обусловлены функциональностью того или иного инструмента.

Предварительно проведенный анализ функциональности различных инструментов показал их адаптированность к отчетам определенной сложности. Сама по себе функциональность встроенного инструментария в платформу 1С интересна и рассматривается лишь по причине зрелости интерфейса настройки отчета.

Перейдем к моделированию диаграмм, иллюстрирующих разные подходы к разработке отчетов.

На рисунке 3 представлен процесс разработки отчета с применением конструктора запроса (с обработкой данных).



Рисунок 3 – С помощью конструктора запроса с обработкой данных

Рисунок 4 иллюстрирует процесс разработки отчета с помощью СКД (система компоновки данных).



Рисунок 4 – С помощью СКД (система компоновки данных)

Важно отметить, что в 1С реализована возможность разработки отчета исключительно путем написания программного кода. Для этого подхода диаграмму процесса не имеет смысла моделировать в виду не показательности.

В результате анализа всех трех подходов создания отчетов в 1С приходим к выводу, что для целей решения нашей проблемы интересен подход настройки импорта данных путем выбора набора данных, что возможно, благодаря predetermined предметной области, что характерно для всех встроенных report-инструментов.

Далее проведем анализ функционала генератора отчетов в web-сервисе «Мегаплан».

Встроенный инструментальный web-сервиса «Мегаплан» позволяет не только самостоятельно создавать, но и видоизменять уже имеющиеся отчеты. Report-инструмент дает возможность не только сохранить табличный отчет в формате xlsx, но и просто просматривать его внутри web-приложения, а также автоматически генерировать отчет по расписанию с рассылкой на e-mail [29].

Характерными особенностью данного report-инструмента является предопределенность наборов данных с интуитивно понятными названиями, что является обычным для специализированных (встроенных) report-систем, заточенных под определенный контекст и предметную область.

Настройка (создание) пользовательского отчета состоит из нескольких типовых этапов:

- Определяется название отчета, вводится его описание, а также определяется ограниченный контекст отчетов;
- Определяется список колонок методом добавления их из выпадающего списка доступных;
- Определяется группировка;
- Настраивается фильтрация;
- Уточняются права доступа к отдельным столбцам (опционально).

После этого происходит сохранение отчета.

Имеющиеся ограничения системы «Мегаплан» обоснованы ее спецификой и наработанными приемами работы. Примером такого ограничения является возможность формировать отчеты исключительно в табличном виде.

Интересной особенностью является возможность предварительного просмотра отчета в онлайн-режиме прямо на web-странице, где происходит его настройка. Данный функционал является потенциально полезным для решения нашей проблемы, так как позволит простому пользователю (бизнес-актору) в удобной форме контролировать результаты собственных действий по настройке отчета без привлечения квалифицированного программиста.

Рассмотрим модуль отчетов в системе баг-трекера Redmine.

В redmine существует два варианта формирования отчетов – исходя из данных, обрабатываемых в системе[39]:

- Выгрузка информации по задачам, соответствующим определенным критериям, на основе встроенного настраиваемого report-функционал.
- С помощью дополнительного интегрируемого функционала, благодаря возможности установки специальных плагинов.

Первый подход характеризуется возможностью создания элементарных табличных отчетов, основанных на контексте управления задачами. При этом существует характерная возможность выгрузки данных в формате xls и pdf, что является достаточным выбором относительно специфики системы.

Однако, наибольший интерес обычно вызывают возможности формирования отчетов, обретаемые благодаря плагинам. Основная идея таких расширений заключается в специфической интерпретации тех же самых данных в различных разрезах:

- Практики управления проектами Канбан и SCRUM;
- Классические диаграммы управления проектами;
- Отчеты в формате дашбордов;
- KPI сотрудников;
- Бонусы сотрудников, как элемент системы мотивации.

При этом доступные форматы выгрузки отчетов уже определяются спецификой тематики конкретного плагина.

Детальное описание возможностей по созданию и редактированию настроек отчета не имеет большого смысла по причине великого разнообразия плагинов как платных, так и коммерческих. Но важно отметить принадлежность такой системы построения отчетов к классификационной группе «специализированные», т.е. работающие в строго ограниченном

контексте данных, что в значительной степени определяет не только функционал, но и сам процесс по созданию отчетов.

Redmine имеет функционал по настройке условий выборки данных, который продемонстрирован на рисунке 5.

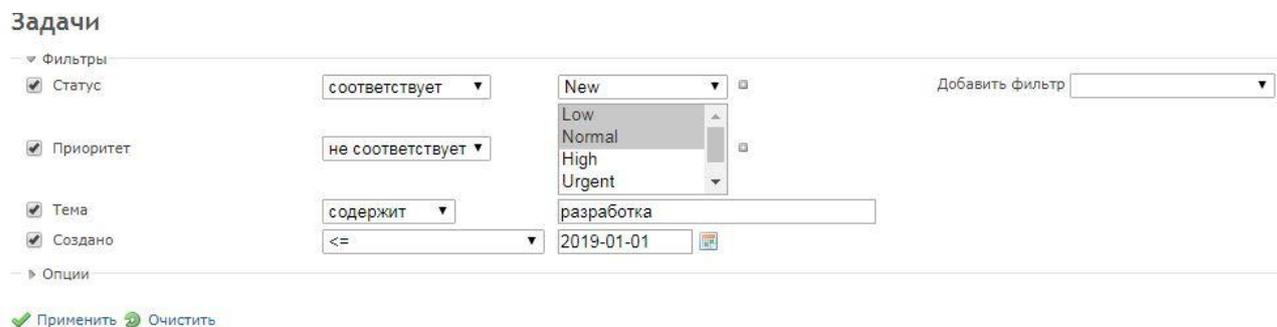


Рисунок 5 – Настройка условий в redmine

Рассмотрим алгоритм настройки поэтапно:

- Пользователь настраивает сортировку путем добавления фильтров из списка «Добавить фильтр». Разные строчки (фильтры) связаны условием И.
- Пользователь выбирает операцию сравнения из списка. Возможные варианты этого списка определяются и зависят от типа выбранного фильтра.
- Пользователь указывает в редактируемом поле значение аргумента сравнения для выбранного фильтра. Существует возможность указать несколько аргументов сравнения. Для этого пользователь должен нажимать на UI-кнопку «+», выбирая необходимое количество значений. Между выбранными аргументами сравнения в рамках одного фильтра предполагается логический оператор ИЛИ.

Представим данный алгоритм настройки в виде блок-схемы представленной на рисунке 6.

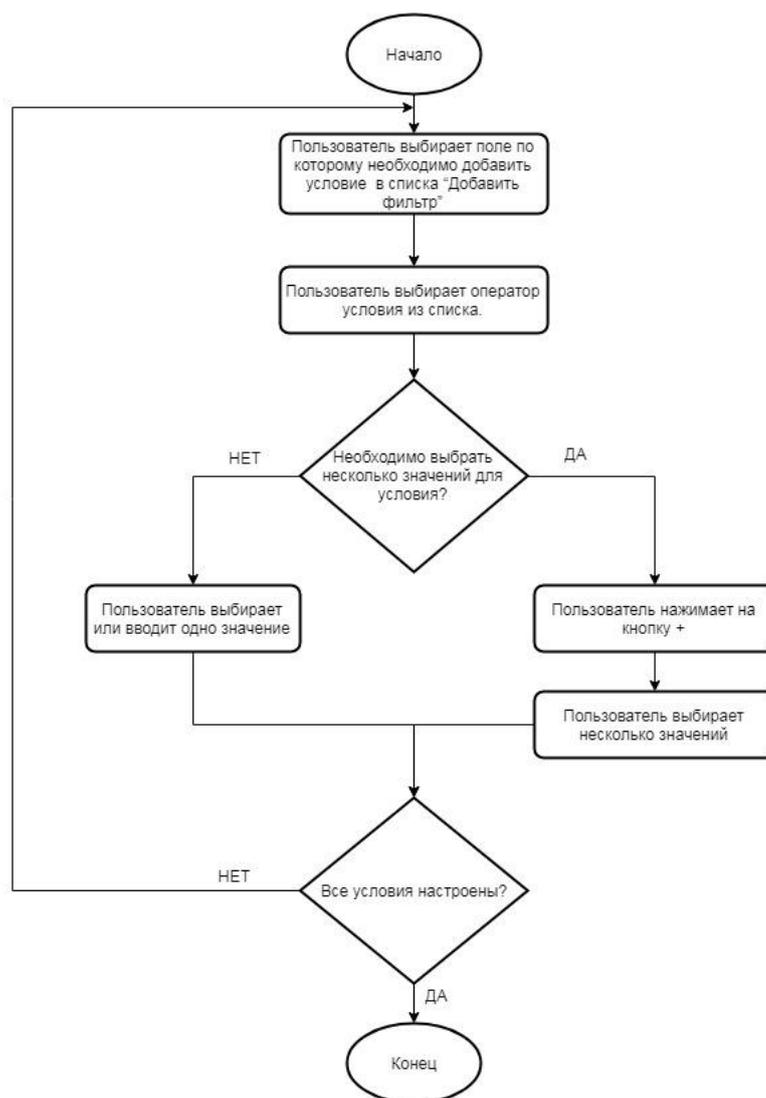


Рисунок 6 – Блок-схема алгоритма условий в redmine

Данный функционал условий в redmine реализован зрело. Поэтому данный пример реализации может быть с успехом использован для преодоления проблемы, решаемой в рамках настоящей работы.

2.2 Анализ генераторов отчетов, интегрируемых в информационные системы

В первую очередь рассмотрим интегрируемый (универсальный) инструмент по созданию отчетов JasperReports.

JasperReports является программной библиотекой для языка Java, которая применяется для генерации отчетов. Проанализируем укрупнено подход ее использования. Печатные документы генерируются на основании .jrxml файла

шаблона путем включения импортируемых из различных источников данных [18].

Для упрощения подготовки дизайна отчетов может быть использовано бесплатное программное обеспечение Jasper Studio, которое является визуальным редактором. Шаблон отчёта – это файл в XML-формате, в котором хранится структура отчета, его дизайн и логика.

Входные данные отчёта – это параметры и источник данных (data source). Параметры – это входные аргументы. По сути это переменные языка Java. Их значения неизменны во всём отчёте. Например, если нужно в заголовке отчёта на каждой странице выводить один и тот же текст, который генерируется перед показом отчёта, то можно передать его в отчет как параметр.

Источник данных – это коллекция записей, каждая из которых представляет набор полей (переменных языка Java).

Альтернативой Jasper Studio может служить программный продукт iReport – графический редактор отчётов(создание шаблонов под jasperReports).

Очевидными недостатками с точки зрения целей настоящей работы является:

- отсутствие возможности работать через Web-интерфейс;
- потребность в квалифицированном программисте;
- непригодность для самостоятельной перенастройки отчета

сотрудником, не являющимся программистом.

Инструментов лишенным недостатком отсутствия работы через Web-интерфейс является Stimulsoft Reports.Web.

Это группа программных продуктов, представляющих из себя генератор отчетов, предназначенный для разработки и построения отчетов в Web-среде [23].

Stimulsoft Reports.Web обеспечивает полный цикл разработки отчетов, начиная от создания макетов отчетов и заканчивая их отображением в браузере. Это полнофункциональный генератор отчетов, который позволяет редактировать отчеты прямо в Web-браузере.

Stimulsoft Reports.Web предлагает к использованию два полнофункциональных дизайнера отчетов, которые работают прямо в окне Web-браузера. Оба дизайнера отчетов выполнены в виде ASP.NET и ASP.NET MVC компонентов. Работают компоненты по клиент-серверной технологии. На стороне ASP.NET, работу сервера выполняет компонент размещенный на web странице. На стороне клиента, работает HTML5 модуль дизайнера отчетов. Оба дизайнера отчетов использует Ribbon-интерфейс, содержат большой инструментарий для работы с отчетами, имеют компактный размер для быстрой загрузки в браузере. HTML5 версия специально разработана для построения отчетов, как на мобильных устройствах, так и на персональных компьютерах.

Генератор отчетов поддерживает большое количество видов данных. Во-первых, полная поддержка бизнес-объектов – любой уровень зависимостей, сортировка, группировка, фильтрация. Во-вторых, поддерживаются большинство типов данных .NET. , все стандартные виды ADO.NET – DataSet, DataTable, DataView, DataRelation. Все данные выводятся, учитывая иерархию данных в DataSet. Все данные типизированы. Имеется возможность получать данные напрямую из MS SQL, Oracle, ODBC, OleDb, FirebirdSQL, PostgreSQL, SQLCE, SQLite и т.д. Также можно без проблем получать данные из XML-файлов.

Возможна установка web-модуля. Библиотеку для разработчиков можно использовать при создании java-проектов. Имеется возможность указать огромное количество разновидностей источников с данными и создать шаблон.

Ключевыми недостатками данного решения для целей данной работы являются:

- непригодность для самостоятельной перенастройки отчета сотрудником, не являющимся программистом;
- это платное дорогостоящее решение, доступное только организациям.

Также существует инструмент по созданию отчетов Crystal Reports. Это известное «классическое» решение для сред разработки программного обеспечения (IDE). Имеет базовый функционал работы через web-интерфейс. [16]

Недостатки:

- имеется только платные версии десктопного приложения и web-решение распространяется по платной подписке;
- непригодность для самостоятельной перенастройки отчета сотрудником, не являющимся программистом.

Аналогичными характеристиками обладает популярный инструмент создания отчетов Fast Reports [19].

Важной особенностью является наличие визуального онлайн-дизайнера, работа с которым осуществляется в браузере. FastReport Online Designer представляет собой web-версию дизайнера отчетов FastReport для платформы .Net.

Онлайн-дизайнер отчетов является RIA (Rich Internet application) приложением, что позволяет запускать его с любого устройства, на котором есть современный интернет-браузер. Онлайн-дизайнер будет работать в последних версиях популярных браузеров (Chrome, Firefox, Opera, Safari, IE), в отличие от десктоп-версии дизайнера, которая работает только в операционной системе Windows.

Но, несмотря на все кроссплатформенные преимущества, онлайн-версия уступает десктопной по удобству и функциональности.

FastReport Online Designer позиционируется как редактор .Net отчетов, которые уже были созданы и размещены по какому-либо UUID на сервере. Онлайн-дизайнер «общается» с сервером через оговоренный заранее API, в который входит 3 запроса.

Тест демонстрационной версии подтвердил достаточные функциональные возможности работы с приложением через WEB-интерфейс представлен на рисунке 7.

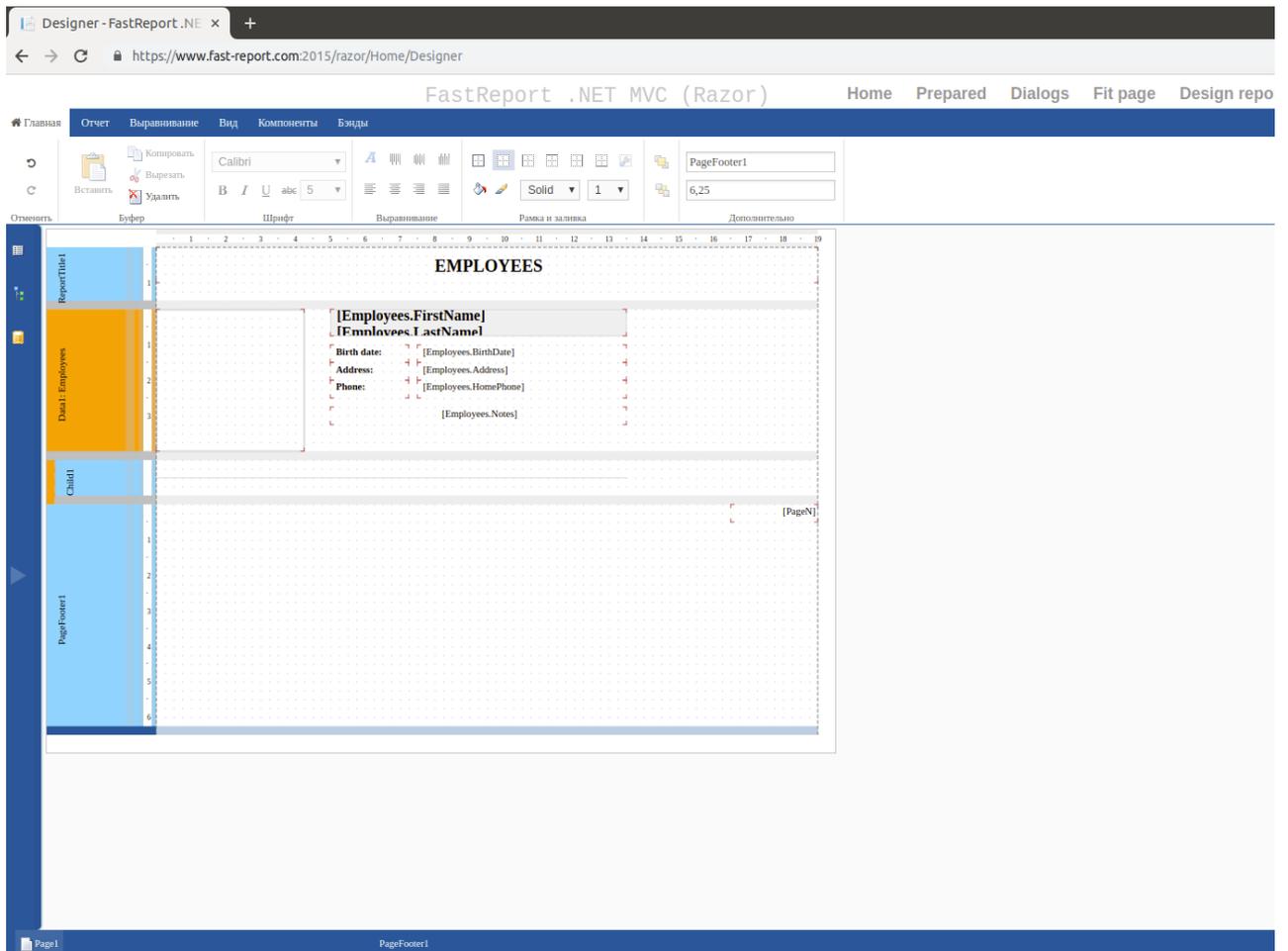


Рисунок 7 – fast report online

Недостатками данного программного продукта являются:

- высокая цена;
- непригодность для самостоятельной перенастройки отчета сотрудником, не являющимся программистом, несмотря на стилизацию web-интерфейса под текстовый онлайн-редактор.

Исследование было бы не полным без рассмотрения инструмента создания отчетов YARG, который представляет собой open-source генератор отчетов, специализированное средство разработки отчетов для CUBA.platform [34].

YARG-генератор отчетов позволяет создавать шаблоны в привычных и доступных всем средах (Microsoft Office и LibreOffice/OpenOffice) и описывать

источники данных в интерфейсе приложения, используя модель данных приложения, SQL, JPQL или исполняемые скрипты.

С помощью генератора отчетов можно:

- Быстро создавать шаблоны отчетов непосредственно из приложения с помощью пошагового мастера;
- Формировать отчёты в DOC, DOCX, ODT, XLS, XLSX, HTML и произвольных текстовых форматах;
- Создавать XLS(X) отчёты сложной структуры – многоуровневые, с агрегацией данных или перекрестными таблицами;
- Использовать графики и формулы в XLS(X) отчетах;
- Конвертировать отчёты из офисных форматов или HTML в PDF.

Не web-интерфейс, а значит, по очевидным причинам не подходит для нашей работы, несмотря на заманчивые возможности.

В целях формирования понимания ключевых этапов работы с встраиваемыми report-инструментами целесообразно графически представить схемы процессов.

Важно отметить концептуальное сходство, а значит и наличие схожих проблем между процессом использование инструментов, разработанных ключевыми игроками ИТ-рынка в сравнении с попытками энтузиастов создать альтернативные решения с целью «отвязать» менеджера от зависимости по отношению к ИТ-специалисту.

Глубокий анализ этих процессов с декомпозицией ключевых этапов в рамках данной работы не имеет смысла, т.к. даже поверхностный анализ дает представление о том, что это не приближает исследователя к решению обнаруженной проблемы.

Ключевым наблюдением является то, что участников процесса не менее двух. Возможны еще более сложные с организационной точки зрения схемы процессов по созданию отчетных документов, например, с привлечением третьего актора – бизнес-аналитика.

На рисунке 8 приводятся диаграмма, иллюстрирующая работу без привлечения посредника в лице аналитика.

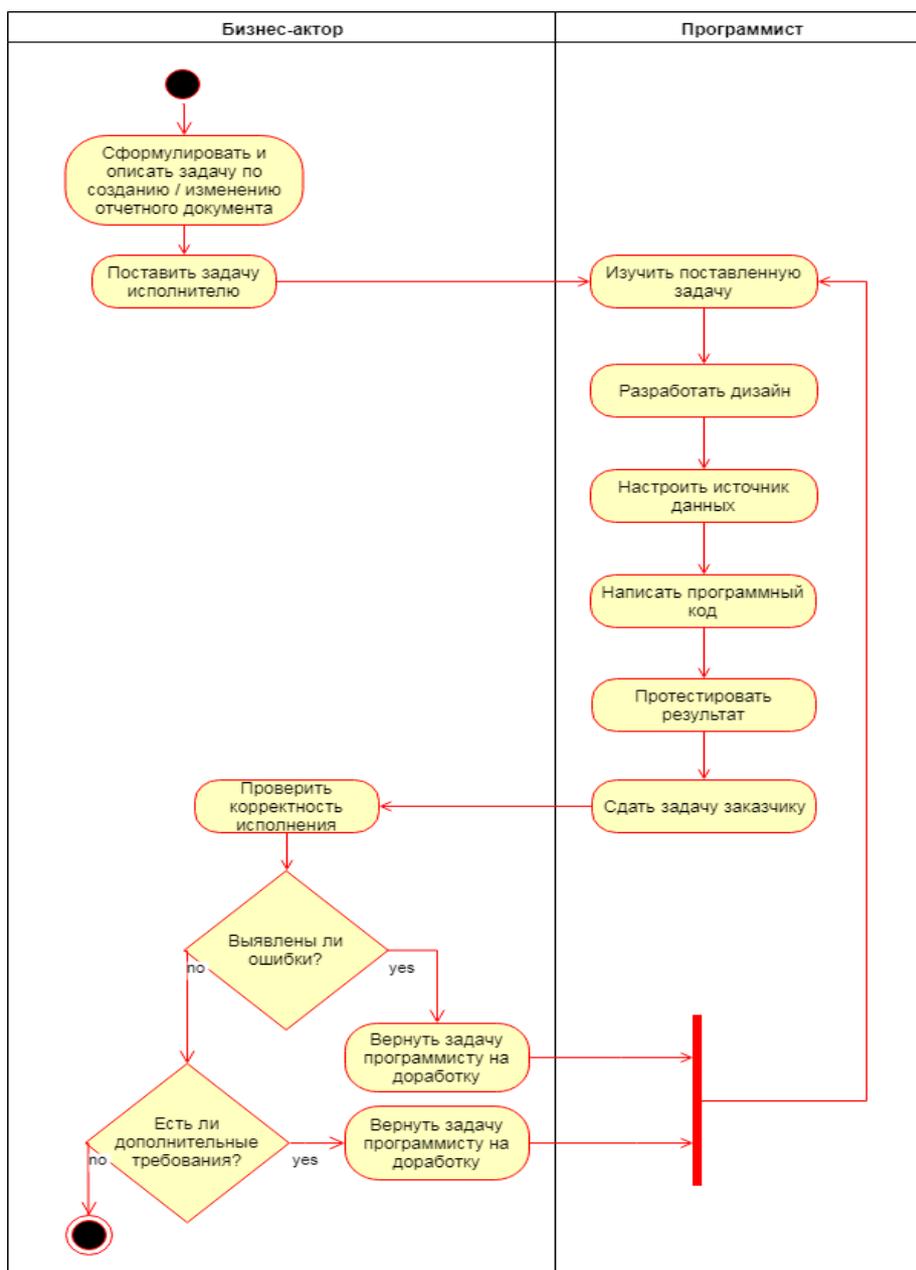


Рисунок 8 – процесс, основанный на технологиях JasperSoft или FastReport

Подведем итоги анализа встраиваемых технологии. В качестве вывода можно отметить необходимость поиска альтернативных технических решений, т.к. анализ показал ориентацию проанализированных продуктов исключительно на пользователей-программистов.

2.3 Анализ низкоуровневых подходов к созданию генераторов отчетов

Проанализируем отдельные технологии, а именно низкоуровневые программные библиотеки и RESTful-API, которые характеризуются узкой специализацией и позволяют разработать программное решение для генерации отчетов.

Рассмотрим известную библиотеку Apache POI на языке Java, назначение которой заключается в чтении и записи документов Microsoft Office.

Обрабатываемыми форматами являются файлы таких приложений как Excel, PowerPoint и Word. POI позволяет программно генерировать новые файлы или изменять существующие документы [24].

Библиотека позволяет обрабатывать вложенные (embedded) объекты, такие как документы, картинки и т.д.

В части функционала, касаемо excel-файлов необходимо отметить следующие интересные возможности:

- обработка большого объема записей - более 100 тысяч записей;
- библиотека POI поддерживает 140 функций из 300 имеющихся в Excel 2010.

Касаемо word-файлов также присутствуют следующие возможности:

- конвертация doc-файлов в HTML, XML, PDF (с использованием Apache FOP).
- возможности применения удобных инструментов парсинга содержимого файла, которые могут быть применены для определения спецметок, обозначающих место вставки в шаблон какой-либо информации.

Однако, помимо возможностей низкоуровневого доступа к файлу шаблона важно найти еще и текстовый онлайн-редактор, который сможет выполнить роль удобного редактора шаблонов.

Известные производители программного обеспечения, такие как Google, Microsoft предоставляют бесплатные онлайн-сервисы по созданию, редактированию офисных документов, а именно файлов в форматах doc, docx, xls,xlsx. Кроме этого, данные сервисы предоставляют возможности по

взаимодействию с экосистемой данных сервисов, а также обращаться непосредственно к файлам и хранилищу через REST-API.

Проведем дополнительный анализ онлайн-редакторов, которые могут быть использованы в качестве средства визуального редактирования шаблонов.

Данный анализ сведен в таблицу 9.

Таблица 9 – Анализ возможности использования текстовых онлайн-редакторов в качестве визуальных редакторов шаблонов

Наименование продукта	Форматы файлов шаблонов	Выходные форматы документов	Наличие API	Стоимость
Microsoft Office-Online	xlsx, docx, pdf, dot, xlt	Xls, doc, xlsx, docx, ODF, pdf	Есть	Бесплатная и платная версии.
Google Documents	xlsx, docx, pdf, dot, xlt	Xls, doc, xlsx, docx, ODF, pdf	Есть (возможность использования любых популярных языков)	Бесплатная и платная версии.
myoffice.com	xlsx, docx, pdf	xlsx, docx, pdf	Есть	Индивидуально – только для корпоративного сегмента рынка.
onlyoffice.com	xlsx, docx, pdf, dot, xlt и собственный формат шаблонов	xlsx, docx, pdf	Есть	Условно бесплатная.

Из таблицы 9 видно, что такие решения как myoffice.com и onlyoffice.com нам не подходят по причине отсутствия поддержки старых форматов файлов (doc, xls), а также из-за отсутствия бесплатных версий.

Приоритетным выбором из перечисленных в таблице сервисов являются решения от Microsoft и Google, однако сервис Google предоставляет широкий выбор api-библиотек для различных языков программирования.

Таким образом, был проведен анализ низкоуровневых технологий, и лидером сравнения стали облачные сервисы Google, предоставляющие комплекс необходимых инструментов:

- низкоуровневый доступ к содержимому шаблонов;
- онлайн-редактор шаблонов.

2.4 Анализ альтернативных решений проблемы, описанных в общедоступных источниках

В ходе изучения материалов распространенных в интернете о работах аналогичной направленности были найдены и проанализированы следующие оригинальные технические решения:

- Двухфазная технологии формирования (разработки) отчетных документов на основе шаблонов в формате файлов офисных редакторов [3];
- Самодельный идеальный конфигуратор отчетов на базе системы «ERP-Платформа» [28];
- Интеграция средств управления отчетными документами со средствами ведения справочника и OLAP-инструментария [12];

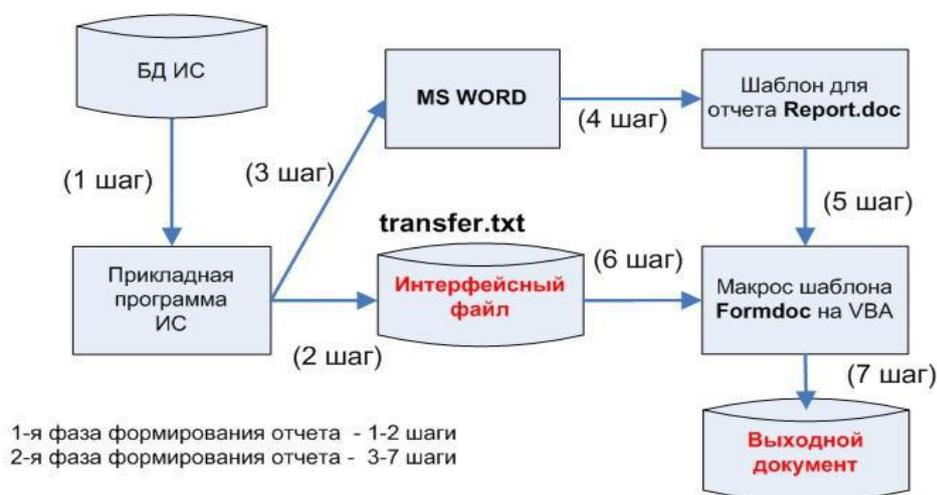


Рисунок 9 – Двухфазная технология формирования отчетов

Более детально по шагам этот процесс выглядит так:

1-й шаг – Формирование выборки из БД ИС на основе конкретных запросов сформированных пользователем в прикладной программе. Такие запросы выполняются специализированными функциями и чаще всего основаны на SQL операторах, формируемых в программе.

2-й шаг – На основе полученной выборки формируется простой текстовый файл (у нас – интерфейсный файл), который содержит в упорядоченном виде информацию выборки. Упорядоченность необходима для дальнейшей однозначной интерпретации данных.

3-й шаг – Выполняется автоматизированный запуск универсального текстового процессора (в нашем случае MS WORD), с параметрами, указывающими какой из шаблонов, заранее подготовленных пользователем шаблонов, нужно использовать в данном конкретном случае.

4-й шаг – Текстовый процессор загружает заданный шаблон (в нашем случае – Report.doc) и вызывает программу (макрос) на Visual Basic for Application (VBA) [2,3], прикрепленную к данному шаблону.

5-й шаг – Программа макроса выполняется, причем она считывает данные из интерфейсного файла (6-й шаг) и вставляет эти данные в формируемый выходной документ (7-й шаг). Вставка в документ данных выполняется последовательно на основе закладок (Bookmark), которые предусмотрены в шаблоне документа. Далее документу может быть присвоено название файла, передаваемое из прикладной программы, а работа с ним может быть продолжена, как с любым документом, сделанным в универсальном текстовом редакторе.

Данное программное решение предопределяет процесс разработки отчета. Представим его в виде диаграммы на рисунке 10.

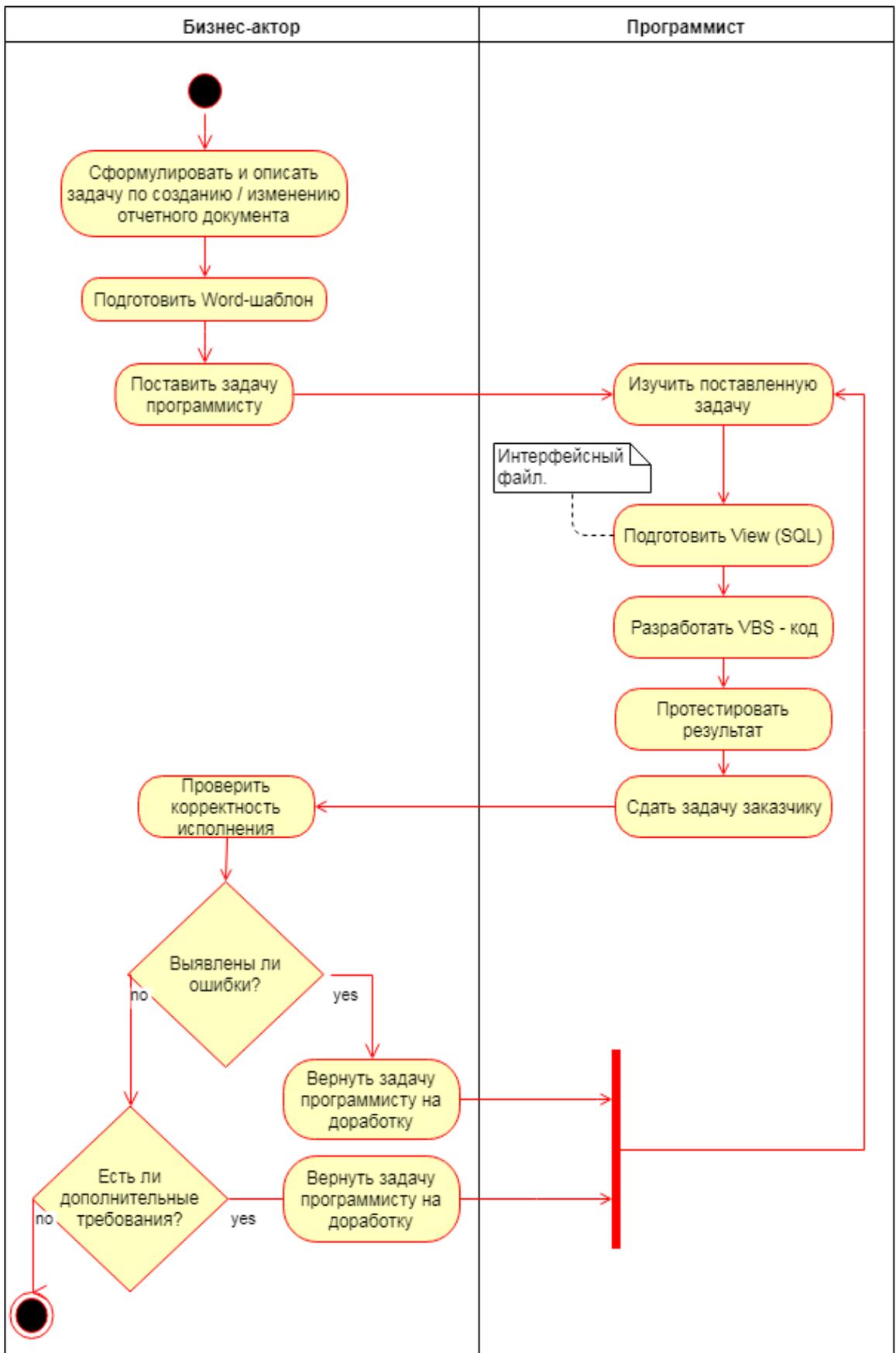


Рисунок 10 – Процесс при двухэтапной технологии формирования отчета с помощью word-шаблона

Интересной особенностью рассмотренного двухфазного подхода является самостоятельная разработка шаблона отчета бизнес-актором, при этом сам шаблон используется не в качестве макета будущего файла отчета, а в качестве конечного файла наполняемого данными. Однако, главным недостатком этого решения является то, что оно не позволяет полностью отказаться от привлечения к работе второго актора – программиста.

В данной главе был произведен анализ технологий и решений. В результате анализа можно отметить, что описанная в данной главе классификация решений и технологий показывает:

- Встроенные генераторы отчетов невозможно применить для решения задач вне информационной системы, для которой они разработаны;
- Интегрируемые инструменты генерации отчетов в виду своей универсальности не позволяют исключить труд программиста из процесса создания отчетов;
- Независимые попытки создания нишевых технических решений, направленных на преодоление исследуемой проблемы позволили упростить и сократить процесс разработки отчета, но полностью не исключили необходимость привлечения программиста к участию в процессе;
- Низкоуровневые технологии, решения и инструменты не дают готового решения проблемы, а лишь позволяют их скомбинировать, руководствуясь какой-либо идеей решения проблемы.

Таким образом, в следующей главе будет предложен собственный организационный подход и разработано соответствующее ему программное обеспечение. Это возможно благодаря комбинации удачных решений: редактор шаблонов реализован в виде онлайн-сервиса (FastReport), предпросмотр результата заполнения шаблона (Мегаплан), интерфейс настройки условий отбора (Redmine, 1С), самостоятельная подготовка бизнес-актором шаблона отчета с использованием офисных приложений (двухфазная технология генерации отчетов [3]).

ГЛАВА 3 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ТЕХНИЧЕСКОГО РЕШЕНИЯ НА ОСНОВЕ ОПТИМИЗИРОВАННОГО ПРОЦЕССА РАЗРАБОТКИ ОТЧЕТОВ

В основе разработки собственного организационно-технического подхода должен лежать измененный процесс подготовки нового отчетного документа или видоизменение уже созданного.

Таким образом, первостепенной задачей анализа является выработка оптимизированного процесса. Этого можно достичь путем отсеечения необоснованных этапов, а также уточнения состава минимально необходимых.

3.1 Предварительный бизнес-анализ и проектирование процесса разработки отчета

Ключевой проблемой исходного процесса видится вынужденная несамостоятельная работа, а именно, необходимость профессионально-социального взаимодействия с исполнителем по вопросу реализации каждой отдельной задачи.

С точки зрения оптимальной организации работы представляется необходимым устранить конкуренцию менеджеров за очередность исполнения задач. Эта причина исходит также из несамостоятельности, а точнее из необходимости обращаться к ограниченному ресурсу - компетентному программисту. Ситуация усугубляется обычно тем, что отсутствует возможность данный ограниченный ресурс увеличить из-за неготовности компании к расширению штата. За рамками конструктивного описания процесса можно оставить возможные проблемы конструктивных мероприятий связанных с межличностными отношениями в коллективе. С точки зрения руководства необходимость расширения штатов не является очевидной по причине отсутствия прямой корреляции между пользой от отчетной деятельности и финансовым результатом в виде прибыли. Между тем при старом организации процесса можно отметить значительность затрат на

разработку отчетов, пропорционально их количеству. Дополнительным контр аргументом по отношению к идеи расширения штата является классическая, постоянна действующая задача оптимизации затрат.

С точки зрения организационных первопричин разбираемой причины можно отметить затраты времени из-за бюрократических процедур между подразделениями, а также возможную итеррационность обращения в рамках решения одной конкретной задачи.

С точки зрения технических причин можно отметить predetermined выбор report-инструмента, который обусловлен профессиональными привычками, либо ложившейся практикой заимствования готовых решений.

При этом важно отметить, что доступные на рынке инструменты нацелены на стабильные во времени формы отчетов, так как разработчики встраиваемого инструментария нацелены исключительно на крупные ниши рынка, а альтернативные report-инструменты не развиты либо отсутствуют.

С точки зрения кадрового ресурса важно обратить внимание на следующие процессуальные аспекты такие как: узкая специализация менеджера из-за отсутствия профессиональной подготовки, а также узки фокус возможных внутренних программ, не способных качественно приблизить компетенции обычных пользователей к профессиональным навыкам программиста.

В результате такого анализа процессных проблем становится очевидным ключевой критерий проектируемого инновационного процесса, а именно самостоятельность или одноакторность. В первом приближении предлагаемый процесс состоит структурно из нескольких простых этапов характеризуемых строгой самостоятельностью и независимостью их исполнению ключевым актором – менеджером.

При этом предлагаемый процесс концептуально не содержит описанных выше непроизводительных затрат связанных с необходимостью профессионального взаимодействия с исполнителем. Таким образом, ключевой актер самостоятельно ставит задачу, сам оформляет надлежащим образом законченный шаблон, проводит настройки вставки в него данных, а также

тестирует корректность заполнения шаблона в соответствии со своими собственными представлениями.



Рисунок 11 – Процесс в соответствии с предлагаемыми решениями

Описанные на рисунке 11 элементы бизнес-процесса (этапы) требуют декомпозиции, а также подробного описания. Более детальное описание представлено в виде activity-диаграммы на рисунке 12. На которой представлено взаимодействие между единственным бизнес-актором, офисным приложением и разрабатываемым web-приложением по созданию отчетов.

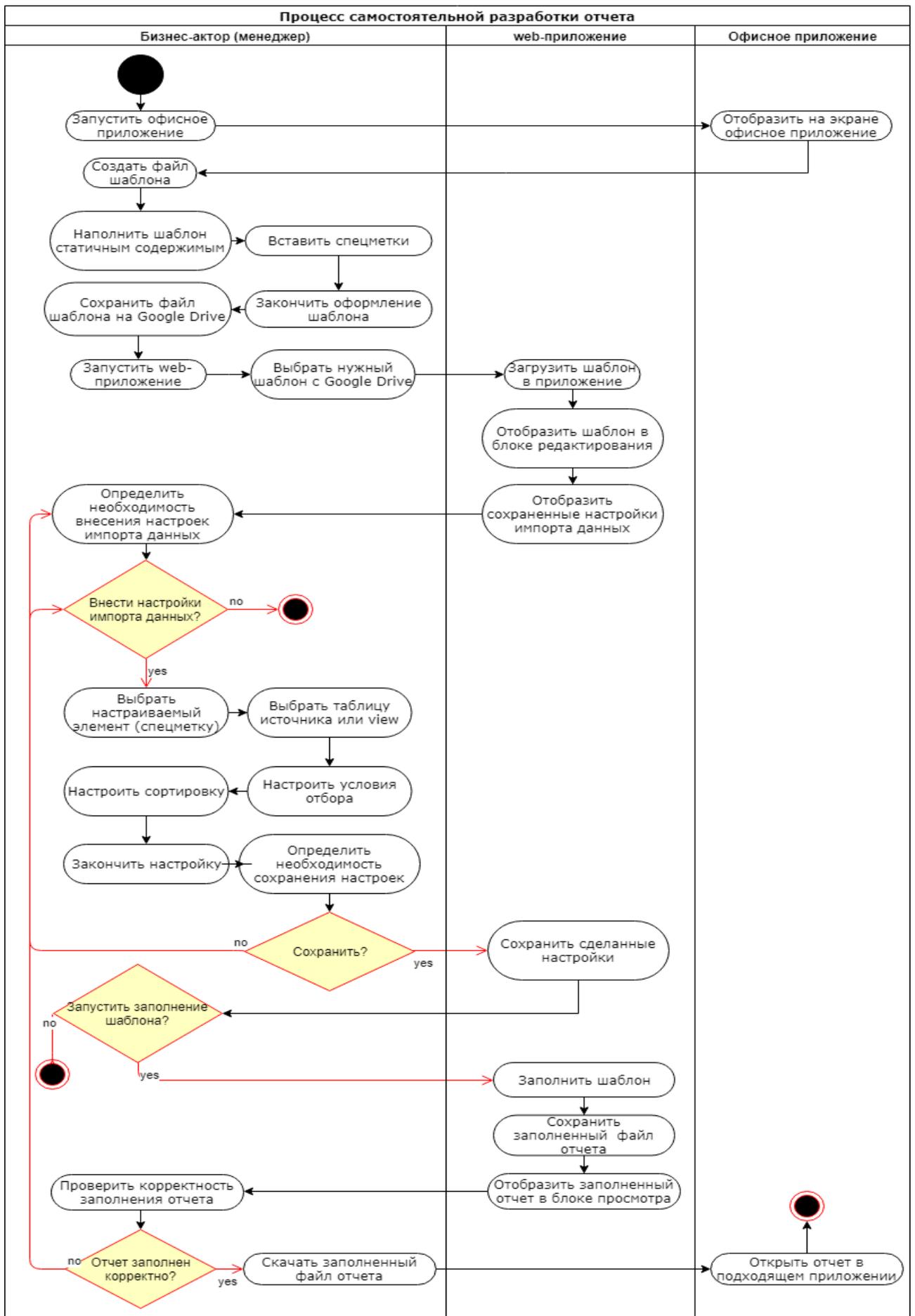


Рисунок 12 – Детализация процесса самостоятельной разработки отчета

Опишем подробнее виденье оптимизированного процесса работы по подготовке отчета.

Пользователь, являющийся хозяином бизнес процесса, начинает работу с подготовки шаблона документа с помощью популярного офисного приложения-редактора (excel, word и прочее). При этом он определяет не только дизайн (представление), но и содержимое (контент) будущего отчета (в понимании англ.терм. Report). Идея использования офисных приложений обоснована функциональностью, а также доступностью с точки зрения требуемых минимальных компетенций.

На следующем этапе пользователь проставляет метки в тех структурных элементах будущего шаблона, куда будут динамически подставляться какие-либо данные из вне. При этом, данному этапу не свойственно определение связей между заполняемыми специализированными полями и источниками данных.

Далее пользователь, используя удобный web-интерфейс, должен определить выше упомянутую связь, загрузив шаблон в разрабатываемый web-сервис. Идеология этого этапа является развитием известных интерфейсных достижений офисных программ связанных с настройкой источников данных. Например, в программном продукте MS Excel реализован интерфейс подключения различных источников данных, который является с функциональной точки зрения минимально достаточным. Вместе с тем является очевидной необходимостью адаптации разрабатываемого аналогичного функционала под нужды и профессиональные компетенции нашего целевого пользователя.

Сопоставление данных с отмеченными специализированными полями может производиться, например, путем отображения статического содержимого шаблона и диалогов настройки predetermined источников данных, что представляет несомненное удобство для пользователя, так как позволяет ему производить точечную пошаговую настройку отчета, не отвлекаясь от бизнес-контекста.

Определенную положительную роль в обеспечении удобства настройки подключенного по умолчанию источника данных может сыграть минимальная документация. Состав этой документации может определять перечень таблиц и составляющих их столбцов.

Предварительно пользователю необходимо выбрать вариант импорта данных: вставить ячейку, строку, столбец, или таблицу.

На интерфейсе пользователя аналогичное удобство может быть реализовано в виде списка доступных таблиц, а при наведении мышью на конкретный элемент списка может показываться его описание. Выбранная таким образом таблица предопределяет возможность выбора необходимого количества составляющих её столбцов.

Отбор необходимого количества строк реализовать в виде удобного, генерируемого «на лету», список интерфейсных элементов, которые позволяют повторить в GUI функционал, аналогичный SQL (перечисление условий после служебного слова where).

После организации связей между специализированными полями в шаблоне и конкретными данными требуется возможность сохранить созданные связи в информационной системе либо в самом файле шаблона.

Запуск заполнения документа на основании разработанного шаблона осуществляется по инициативе пользователя. Генерация целевого заполненного документа (отчета) должна проходить в запрашиваемом формате.

Более подробного описания требует этап №2. В этих целях приведем его расшифровку ниже в форме диаграммы, представленной на рисунке 13. Представление этого этапа в виде диаграммы обосновано его наполненностью основными новаторскими идеями по сравнению с остальными.

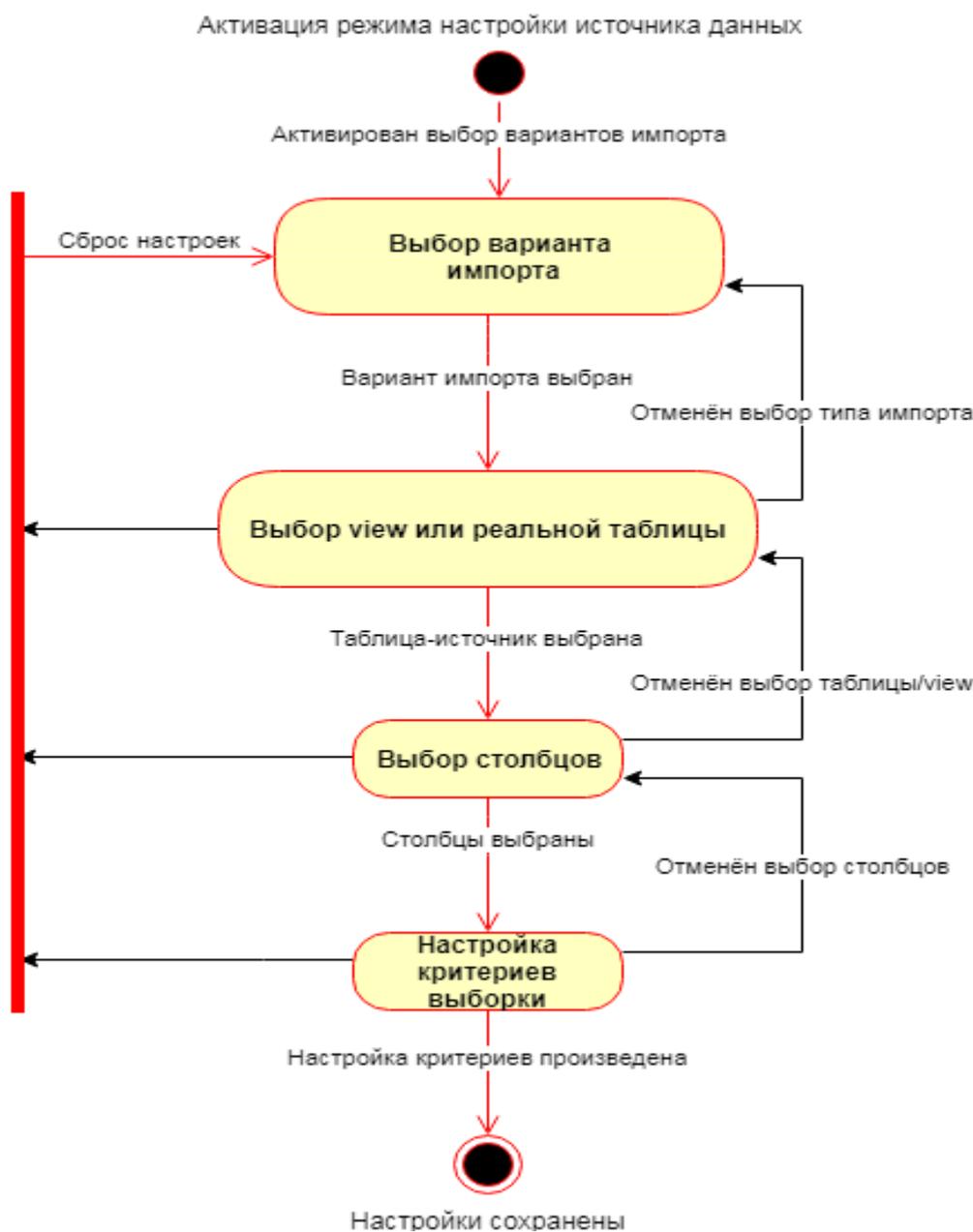


Рисунок 13 – Диаграмма процесса настройки источника данных

Таблица 10 – Спецификация к диаграмме настройки источника данных

Исходное состояние	Триггер (событие)	Условие перехода	Эффект	Конечное состояние
Активация режима настроек и источника данных	Активирован выбор вариантов импорта	Безусловно	- показана подсказка текущего этапа; - активны элементы настройки текущего этапа.	Выбор варианта импорта

Исходное состояние	Триггер (событие)	Условие перехода	Эффект	Конечное состояние
Выбор варианта импорта	Вариант импорта выбран	Безусловно	<ul style="list-style-type: none"> - отображается надпись о выбранном варианте импорта (например:»Импортировать ячейку»); - показана подсказка текущего этапа; - активны элементы настройки текущего этапа; - активны кнопки возврата. 	Выбор view или реальной таблицы
Выбор view или реальной таблицы	Таблица-источник выбрана	Осуществлен выбор свойственны й исходному состоянию и нажата кнопка «Далее»»	<ul style="list-style-type: none"> - отображается надпись о выбранном view или таблице; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа. 	Выбор столбцов
Выбор столбцов	Столбцы выбраны	Осуществлен выбор свойственны й исходному состоянию и нажата кнопка «Далее»»	<ul style="list-style-type: none"> - отображается надпись о выбранных столбцах; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа. 	Настройка критериев выборки
Настройка критериев выборки	Настройка критериев произведена	Произведена настройка критериев выборки и нажата кнопка «Сохранить»»	- отображается надпись «Сохранено».	Настройка и сохранены

Исходное состояние	Триггер (событие)	Условие перехода	Эффект	Конечное состояние
Выбор view или реальной таблицы	Отменён выбор типа импорта	Нажата кнопка «Назад»	<ul style="list-style-type: none"> - исчезает надпись о выбранном варианте импорта; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа; - деактивированы кнопки возврата. 	Выбор варианта импорта
Выбор столбцов	Отменён выбор таблицы/view	Нажата кнопка «Назад»	<ul style="list-style-type: none"> - исчезает надпись о выбранном view или таблице; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа. 	Выбор view или реальной таблицы
Настройка критериев выборки	Отменён выбор столбцов	Нажата кнопка «Назад»	<ul style="list-style-type: none"> - исчезает надпись о выбранных столбцах; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа. 	Выбор столбцов
Выбор view или реальной таблицы	Сброс настроек	Нажата кнопка «В начало»	<ul style="list-style-type: none"> - исчезают надписи о всех выбранных настройках; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа; - деактивированы кнопки возврата. 	Выбор варианта импорта

Исходное состояние	Триггер (событие)	Условие перехода	Эффект	Конечное состояние
Выбор столбцов	Сброс настроек	Нажата кнопка «В начало»	- исчезают надписи о всех выбранных настройках; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа; - деактивированы кнопки возврата.	Выбор варианта импорта
Настройка критериев выборки	Сброс настроек	Нажата кнопка «В начало»	- исчезают надписи о всех выбранных настройках; - показана подсказка текущего этапа; - активны элементы настройки текущего этапа; - деактивированы кнопки возврата.	Выбор варианта импорта

Вышеописанные диаграммы позволяют реализовать прототип интерфейса предполагаемого инновационного приложения.

В данном разделе раскрыты основные идеи проектирования иного (новаторского) подхода к организации процесса разработки отчета, важного для достижения целей настоящей работы, а также для создания предполагаемого web-приложения в частности.

3.2 Разработка вариантов использования ПО создания отчетов

В целях проведения анализа разрабатываемого report-инструмента, необходимо отметить, что в результате преобразования процесса разработки отчета был исключен второстепенный актер – программист. Это позволило сократить процесс до трех прецедентов и одного актора – менеджера. Осуществив выделения единственного актора, опишем прецеденты использования и представим их в виде спецификации – таблицы 11-13.

Таблица 11 – Прецедент: определяет задачу

Прецедент: Определяет задачу
Id: 1
Краткое описание: Менеджер создает шаблон отчетного документа.
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: Наличие офисного пакета программ, которые будут использованы для подготовки эскиза/прототипа шаблона.
Основной поток: Менеджер определяет назначение, состав, логику использования шаблона, а также параметры, по которым можно судить о корректности реализации. Оформляет шаблон.
Постусловия: Прототип шаблона составлен в нем расставлены метки-поля которые буду в дальнейшем связываться с данными.

Таблица 12 – Прецедент: реализует задачу

Прецедент: Реализует задачу
Id: 2
Краткое описание: Менеджер настраивает импорт данных.
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: Загруженный в систему предварительный шаблон с помеченными областями для заполнения данными.
Основной поток: После загрузки исходного файла-шаблона в систему, актер осуществляет настройку связи между данными и специальными помеченными областями в шаблоне.
Постусловия: Сохранение в систему файл хранящий информацию о связи между специализированными полями и конкретными наборами данных.

Таблица 13 – Прецедент: проверяет корректность исполнения

Прецедент: Проверяет корректность исполнения
Id: 3

Краткое описание: Менеджер проверяет корректность заполнения отчета
Главные актеры: Менеджер
Второстепенные актеры: отсутствуют
Предусловия: Отчет заполнен данными
Основной поток: Осуществляет проверку типовых параметров, по которым можно судить о корректности реализации задуманного.
Постусловия: Получение полностью законченного выходного документа-отчета.

Таким образом, мы рассмотрели разрабатываемый процесс с помощью диаграммы прецедентов, что является ключевым этапом на этапе анализа требований к разрабатываемому программному обеспечению. Спецификация к диаграмму раскрывает отдельные прецеденты с точки зрения функциональной модели.

3.3 Формулирование требований к ПО создания отчетов

Учитывая явную обоснованную потребность организаций в возможности самостоятельной подготовки отчетных, в т.ч. шаблонизированных отчетных документов, требуется разработать report-инструмент в соответствии с новой, еще нигде не реализованной концепцией. Для этого требуется ее разработать с учетом очевидных пожеланий заинтересованных лиц.

Обобщенный портрет заинтересованных лиц не определяется четко. Однако, важно понимать, что под ними подразумеваются бизнес-акторы, которые независимо от отрасли испытывают схожие потребности в ведении самостоятельной аналитической и отчетной деятельности, для чего и нужен адаптированный report-инструмент.

Вопрос разработки концепции всегда характеризуется идеологией решения конкретной проблемы, в качестве которой может быть взята основная проблема исследования, описываемая в настоящей работе. Что же касается предстоящей реализации, то она всегда базируется на комбинации известных

решений с изящным дополнением в виде эвристических решений отдельных подзадач.

Определим основной тезис концептуальной модели искомого программного решения: требуется идеальный report-инструмент (для типичного представителя бизнес-процесса) с исчезающе малым количеством низкоуровневых технологий типа SQL-запросов, отображаемых на интерфейсе пользователя.

Идеальный report-инструмент – оптимальная комбинация простоты и максимальной функциональности.

В качестве анти-примера можно упомянуть функционал 1С: многофункциональный, но обычный бизнес-актор, не обладающий компетенциями программиста 1С, не сможет и не будет его использовать. Другой крайностью является встроенный инструментарий платформы Битрикс: интуитивный, слишком простой, однако не отвечает требованиям минимальной функциональности.

Идеальный репорт-инструмент должен быть доступен целевому пользователю, однако должен позволять создавать отчеты любого уровня сложности.

Определим базовые функциональные требования к такому инструменту:

1. Элементарная когнитивная доступность. Инструмент разрабатывается для простого менеджера, далекого от использования SQL, позволяет сделать отчет по выбранным заранее подготовленным (View) данным.

2. Многофункциональность. Важно иметь возможность формировать отчеты по всей системе, по любым пользовательским данным.

3. Возможность дополнений состава данных. В целях достижения гибкости при эксплуатации и доработке основной ИС важно обеспечить возможность растущего состава данных, т.е., если было добавлено новое поле в

таблицу, или очередная таблица, или подключается новая база данных, то все изменения должны стать доступными при работе с report-инструментом.

4. Поддержка многоуровневых (вложенных) запросов. Учитывая сложность структуры реальных баз данных, представляется необходимым реализовать возможность использования вложенных запросов без использования наиболее сложных конструкций SQL. Данное требование вызвано потребностью одновременного оперирования несколькими таблицами и справочниками с фактическим применением операций реляционной алгебры, но не явно, а адаптировано к функциональному портрету заинтересованных лиц – бизнес-акторов. Глубина вложенности запросов требуется как минимум на 2 уровня вложенности.

5. Оперирование сложными структурами данных, в т.ч. гетерогенными. На практике часто возникает потребность в формировании сложного отчета, который основан на сложной интерпретации данных, собранных из многих таблиц. В данной ситуации бывает не обойтись простой выборкой данных, а требуется процедурная обработка уровня PL/SQL. Предполагается, что для этого нужно реализовать изящное интерфейсное решение, во многом заменяющее средства работы с PL/SQL популярных ERP-Платформ.

6. Форматы итоговых документов (файлов). Требуется иметь возможность получить отчет в общедоступных форматах таких, как PDF, xlsx, docx.

Нефункциональные требования:

1. Report-инструмент должен ощутимо упростить работу заинтересованных лиц. Необходимо найти обоснованный баланс между простотой и функциональностью. Обычно расширенный функционал более сложен в эксплуатации, что часто приводит к необходимости дополнительного обучения персонала. В данном случае этого нужно избежать, реализовав нечто инновационное.

Базовые потребности должны удовлетворяться элементарными действиями. Углубленная работа с точки зрения требований к отчету, предположительно, требует дополнительных действий на интерфейсе (должен быть гибким): масштабируемая сложность. Модульная архитектура отчета могла бы заменить недружелюбные для простого пользователя технологии типа PL/SQL. Тогда все многообразие отчетов может быть реализовано в доступном для пользователя варианте – с низким порогом вхождения в работу с report-инструментом.

2. Report-инструмент должен обладать гибкостью и отличной адаптацией к изменениям инфраструктуры основной системы и состава данных. Требуется минимизировать либо вовсе устранить необходимость доработок самого report-инструмента при дополнении системы новым функционалом, а значит и новыми информационными потоками и ограниченными контекстами.

В популярных платформенных инструментах, например в 1С это является обусловленным единственным решением, что налагает определенные ограничения на функционал, в т.ч. и на приемы работы по подготовке отчетных документов при смене конфигураций.

3. Report-инструмент должен отлично выполнить свою основную задачу, а именно: предоставить полнофункциональные технологически зрелые возможности простому пользователю. Второстепенный функционал приветствуется только в случае качественного улучшения потребительских свойств разрабатываемого report-инструмента.

Данные идеологически продиктованные специфические требования повторяют изящный подход, применяемый при разработке узкоспециализированных, заточенных на конкретную функцию, утилит. Так, в unix-подобных системах каждая конкретная утилита умеет решать конкретную задачу. Идеология швейцарского ножа для требуемого решения не уместна. Однако не отрицается модульный подход или, на примере все тех же утилит, результат выполнения одной может быть передан для использования другой.

Обычно идеологический подход, продиктованный спецификой разрабатываемого решения, органично распространяется на всю экосистему решения. Так, управление правами доступа (и их применением) к отчетам должен заниматься специализированный модуль.

В случае разработки портируемого решения, которое будет встраиваться в различные программные инфраструктуры, важно учесть технологическую сложность разработки универсального решения в части управления правами доступа к различным наборам данных, отдельных таблиц и даже составных частей таблиц.

Также не имеет смысла реализовывать какой-то свой редактор для PL/SQL-запросов на пользовательском интерфейсе report-инструмента. Это излишне утяжелит интерфейс и деморализует простых пользователей. Целесообразнее для этих целей использовать любой редактор процедур, который позволяет писать и отлаживать сложные запросы.

Аналогичные требования и к самому report-инструменту – он должен блестяще выполнять свои прямые задачи и обладать возможностями интеграции с другими модулями системы.

Данный подход к построению системы позволяет отладить узкоспециализированные функции и не расходовать дорогостоящее время квалифицированных разработчиков (программистов) на реализацию в значительной степени похожих функций в различных частях системы.

3.4 Реализация программного решения для разработки отчетов

На основании проведенного в главе 2.3 анализа низкоуровневых решений и сложившегося понимания концепции разрабатываемого программного обеспечения опишем диаграмму, иллюстрирующую основные архитектурные решения, принятые в отношении будущего ПО.

В основе архитектурного решения лежат возможности, предоставляемые сервисами Google.

Web-приложение осуществляет обработку запросов web-клиента с использованием компонента, выполняющего роль controller`а (в соответствии с паттерном MVC).

Для общения с Google-сервисами создадим три сервиса, назовем их Drive Service, Docs Service, Sheets Service. JS Application является клиентом данного web-сервера. Также на странице приложения имеется iframe-тэг, внутри которого встроена страница с Google-приложением.

Данная диаграмма представлена на рисунке 14.

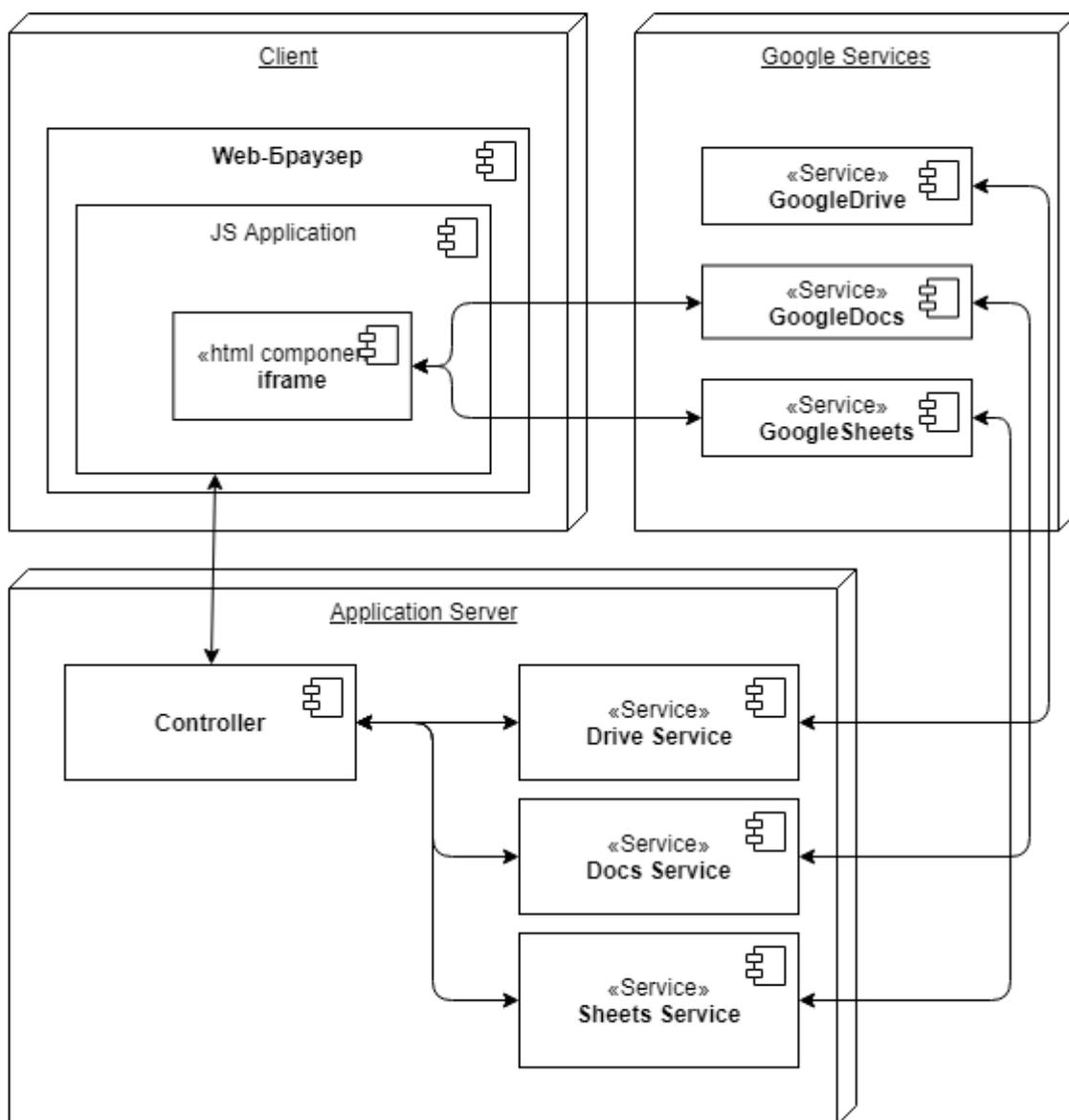


Рисунок 14. Диаграмма компонентов web-приложения

Одним из сложных для понимания этапов работы с приложением является процесс загрузки файла-шаблона. Это связано с наличием не

тривиального действий системы по обнаружению в шаблоне специальной разметки, наличия сохраненных настроек для загруженного экземпляра отчета, а также отображения минимально необходимой информации на UI с учетом состояния отдельных его элементов.

Для лучшего понимания данного этапа процесс был представлен в виде блок-схемы на рисунке 15.

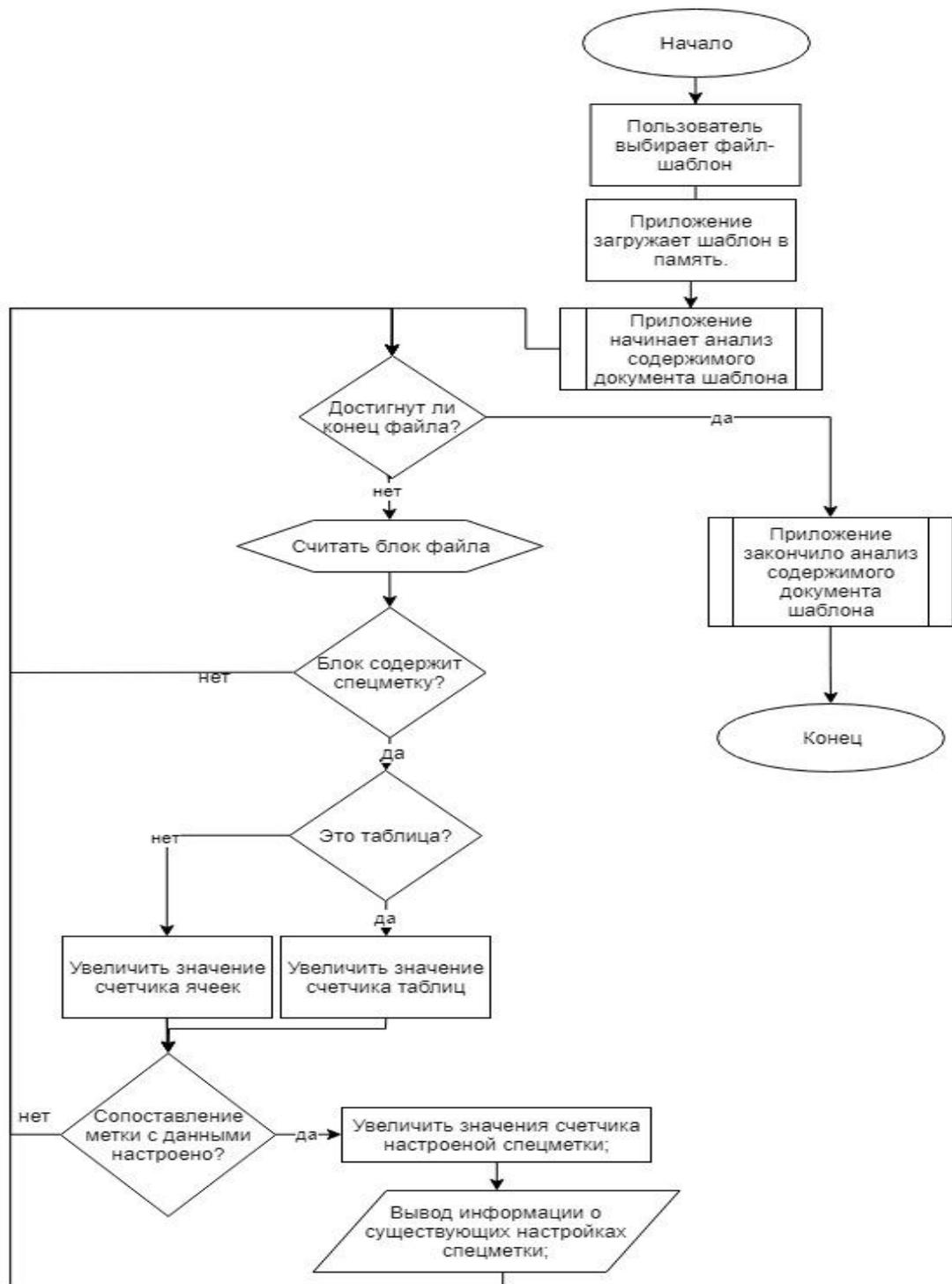


Рисунок 15. Блок-схема процесса загрузки файла-шаблона в приложение

Для целей хранения технологической информации разрабатываемого приложения требуется миниатюрное хранилище, состоящее всего из двух таблиц:

1. `tTranslator` – хранит информацию о человекочитаемых наименованиях полей и таблиц из базы данных.
2. `tMappingSettings` – хранит информацию о сопоставлении специализированных полей из заполняемого шаблона с конкретными таблицами и полями из базы данных. Также хранится информации о настройке условия выборки, сортировки данных.

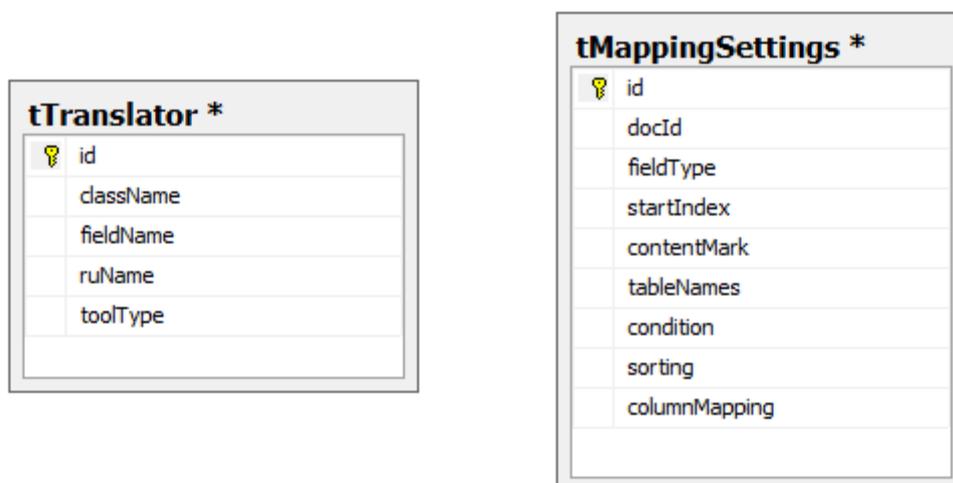


Рисунок 16. Структура таблиц базы данных приложения

В целях отразить назначение полей вышеописанных таблиц приведём их описание ниже.

Таблица 16 – Описание полей таблицы `tTranslator`.

Наименование поля	Описание
<code>className</code>	Имя таблицы – источника импорта
<code>fieldname</code>	Имя поля
<code>ruName</code>	Человекочитаемое описание
<code>toolType</code>	Тип поля

Таблица 14 – Описание полей таблицы `tMappingSettings`.

Наименование поля	Описание
-------------------	----------

docId	Id файла шаблона из GoogleDrive
fieldType	Тип спецметки
startIndex	Индекс начала спецметки в файле
contentMark	Название спецметки
tableNames	С какой таблицей из таблицы сопоставлено
condition	Условие выборки
sorting	Настройка сортировка
columnMapping	Сопоставление таблицы шаблона с полями из базы

Теперь определим функциональный состав и интерфейсные решения разрабатываемого программного обеспечения.

Видится удобным организовать отчеты в древовидный список, т.к. такое представление имеющихся наработок позволяет использовать оптимальные запросы и быстро реализовывать очередной вариант отчета. Такой подход представляется целесообразным в виду того, что значительная часть отчетов являются иной интерпретацией уже имеющегося.

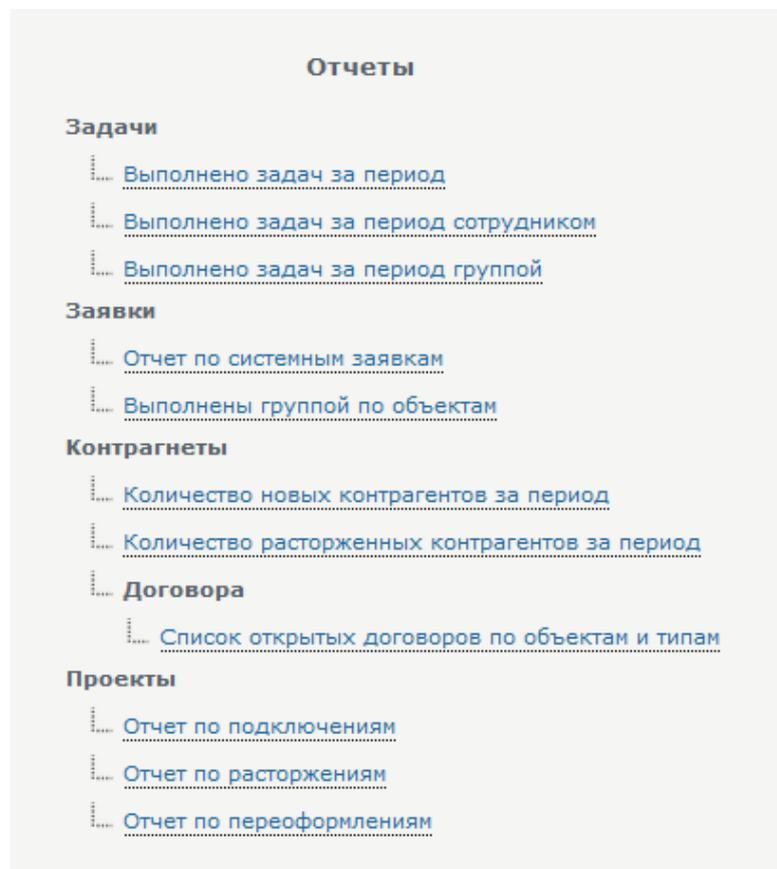


Рисунок 17 – Древовидный список отчетов

Предполагаемый бизнес-пользователь может создать свою ветку, включая вложенности без ограничений, а также перемещать свой отчет в иные ветки, делать их доступными другим пользователям. При указании отчета должна появляться подсказка с описанием отчета.

Управление деревом отчетов представляет отдельный прикладной интерес, т.к. позволяет в понятной форме организовать не только сами отчеты, но и настроенные для них источники данных.

Особую роль в индивидуальном отображении для каждого пользователя дерева отчетов играют права доступа. Допустимость абсолютной доступности всей информации организации для различных бизнес-акторов трудно представить в современных условиях по многим причинам, в том числе из-за классификации некоторой информации как защищаемой, являющейся критичной к утечкам с точки зрения аспектов экономической безопасности, а также требований законодательства в области защиты персональных данных.

При этом, важно понимать допустимость существования общих для всех отчетов, а также тех, которые могут быть доступны отдельным сотрудникам или группам сотрудников. Разграничение доступа может касаться целой ветки report-документов.

Администрирование прав доступа к отчетам, а равно и к связанным источникам данных должно быть делегировано ИТ-специалисту либо иному уполномоченному лицу.

Важно выстроить структуру, т.е. информационные связи в системе, а значит определить наличие зависимостей конкретных полей с другими таблицами, создавая тем самым предпосылки к понятному для бизнес-актора виду, исключая отображение ключей (key) вместо значений (value) из базы. Эту настройку требуется произвести единожды, дополнив необходимыми комментариями, чем и ограничивается роль разработчика в части поддержания функциональности report-подсистемы.

Имя поля	Тип данных	Автозап.	Отч.	Ун.сп.	Связи
Автор	INTEGER		<input type="checkbox"/>		
Дата_добавления	TIMESTAMP		<input type="checkbox"/>		
ID	INTEGER	38 [++1]	<input checked="" type="checkbox"/>		
ID_Контрагент	INTEGER		<input checked="" type="checkbox"/>		КОНТРАГЕНТЫ --> [3. ID]
ID_адреса	INTEGER		<input type="checkbox"/>		
ID_договора	INTEGER		<input checked="" type="checkbox"/>		КОНТРАГЕНТЫ_Договоры --> [3. ID]
Тип_заявки	INTEGER		<input checked="" type="checkbox"/>	№ 42	
Вид_работ	INTEGER		<input checked="" type="checkbox"/>	№ 43	

Рисунок 18 – Настройка связанных таблиц

Настройка каждой отдельной таблицы включает специфические процедуры, связанные с отображением полей таблицы. Галочкой отмечаются поля, которые будут видны пользователю при использовании report-инструмента и доступны для использования при построении отчетных документов. Такая функциональность востребована в связи с нецелесообразностью отображения служебных полей обычному пользователю, который не оценит «мертвую» информацию, считая ее лишней для бизнес-целей.

Однако, если конкретное поле является идентификатором записи в справочнике, то важно указать информацию из справочника. Аналогично нужно решить вопрос с отображением значения из связанной по ключу таблицы.

Данный функционал настройки отображения наборов данных в зависимости от прочих условий может оказаться за границами решаемой задачи. Однако, концептуальная связь с функциональностью основного пользовательского интерфейса очевидна и не требует пояснений.

Работу по настройке связей по ключу делает разработчик – квалифицированный программист.

При добавлении очередного поля таблицы, информация в котором имеет определенную бизнес-ценность, в соответствующих выпадающих списках

добавляется новое поле, доступное для формирования отчетов (с учетом специфики разделения доступа к информации).

Таким образом, после такой специфической настройки репорт-инструмент будет знать связи, а значит, при работе будет выводить пользователю поля, доступные для использования:

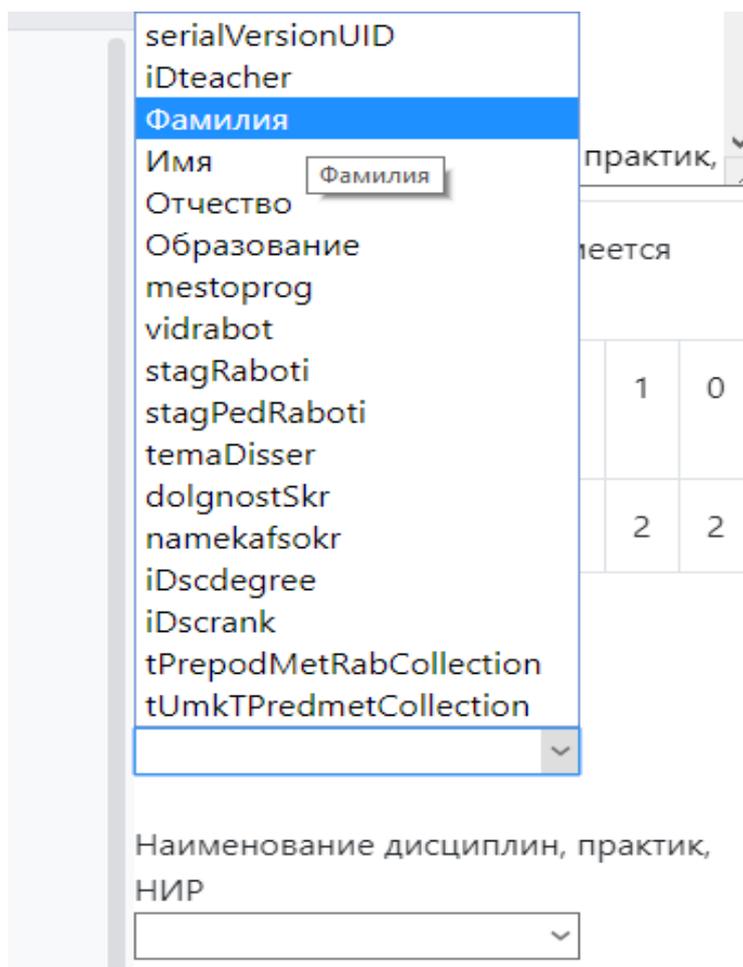


Рисунок 19 – Отображение полей таблицы при настройке источника данных

Определяющую роль в обеспечении адаптированности реальных данных для понимания простого пользователя является подготовка эталонных натуральных таблиц с исключенными служебными полями, а также искусственных таблиц – View, которые являются результатами выполнения запросов, прежде всего сложных вложенных.

View целесообразно использовать как своеобразный уровень абстракции, отделяющий бизнес-пользователей от конкретной реализации модели данных, что создает реальные удобства и неоспоримые преимущества в сравнении с альтернативными подходами (в т.ч. имеющими коммерческую реализацию).

При пополнении базы данных очередной таблицей, ее данные, структура и связи должны быть учтены и адаптированы для использования. Представляется, что подход предварительно денормализованной адаптации данных является прогрессивным по сравнению с тупиковой альтернативой – индивидуальной разработкой отчетных документов усилиями программиста.

От ИТ-специалиста, сопровождающего информационную систему, требуется на предварительном этапе произвести минимальные настройки – отметить новые поля, которые должны быть доступны бизнес-актору при самостоятельной разработке отчетных документов с использованием разрабатываемого report-инструмента.

Для таких пользователей, все должно выглядеть так, что все новые данные автоматически стали доступны.

Представляется, что описанные выше функциональные потребности были бы не реализуемы без глобального справочника значений, использование которого необходимо в качестве дополнительной абстракции, дающей возможность отображать на интерфейсе report-инструмента значения (value), которые связаны с ключом (key) в настраиваемой таблице. Для настройки импорта значений из глобального справочника выделен целый столбец при операциях с полями нужной таблицы. Для обозначения данного справочника введем аббревиатуру ГС (глобальный справочник значений).

Главное, это то, что ГС – элегантный технологичный способ значительно снизить трудозатраты на разработку, исключив необходимость привлечения программиста к реализации каждого отдельного report-документа.

Теоретические аспекты создания справочников определяют, что большинство справочников содержат четыре поля:

1. Идентификатор параметра

2. Наименование параметра
3. Численное значение параметра
4. Текстовое (строковое) значение параметра

Например. Справочник «Дисциплина» может содержать 4 основных поля:

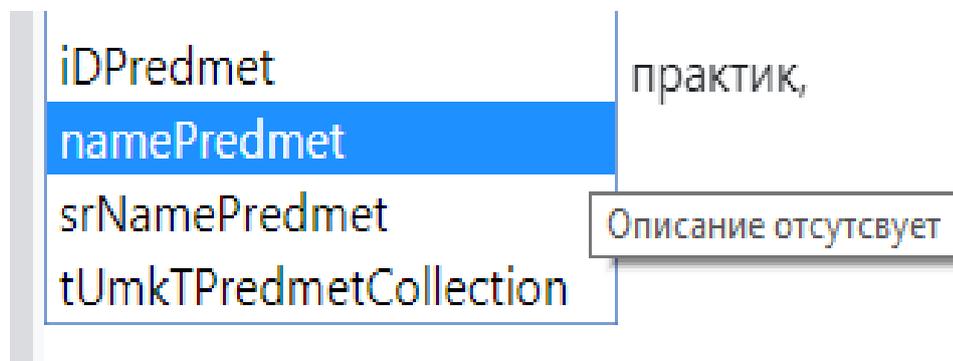


Рисунок 20 – Настройка глобального справочника значений

Другой пример – справочник «Преподаватель»:

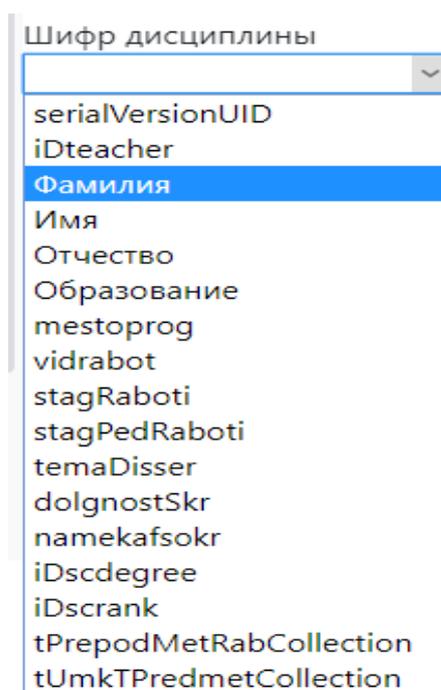


Рисунок 21 – Настройка глобального справочника значений

Важно отметить, что речь идет про справочники, используемые в качестве абстракции, т.е. сгенерированные для кэширования отображения значений вместо ключей.

Учитывая специфику применения ГС, возможна его реализация с использованием key-value базы данных, которая характеризуется несопоставимо большей скоростью выборки значения по ключу по сравнению с

инными неспециализированными базами данных. Однако, важно отметить, что локализация такого справочника на стороне основной информационной системы, а развертывание такого дополнительного оснащения обосновано только в высоконагруженных системах. В других более простых случаях уместна реализация справочника (ГС) в качестве составной части уже развернутой базы данных. В том и другом случае формирование глобального справочника значений осуществляется программистом единожды – на предварительном этапе подготовки report-инструментария к работе.

Удобство использования глобального справочника определяется следующими свойствами записей в нем:

1. они состоят из идентичного набора полей единого типа.
2. возможно использовать один редактор, одну процедуру для запроса и т.п.

При необходимости использования справочника не требуется реализация дополнительных интерфейсов, ссылок, создание запросов.

В общем, ГС удобен тем, что просто решает задачу абстракции данных.

После описания функций и назначения глобального справочника рассмотрим функционал report-инструмента в части настройки источника данных с точки зрения бизнес-актора.

Report-инструмент должен предоставлять следующие функциональные возможности:

1. Позволять выбирать таблицу, поля таблицы, осуществлять сортировку и группировку.
2. Определять условия отбора.
3. Предпросмотр результата отбора.

Основной интерфейс report-инструмента представлен на рисунке 21.

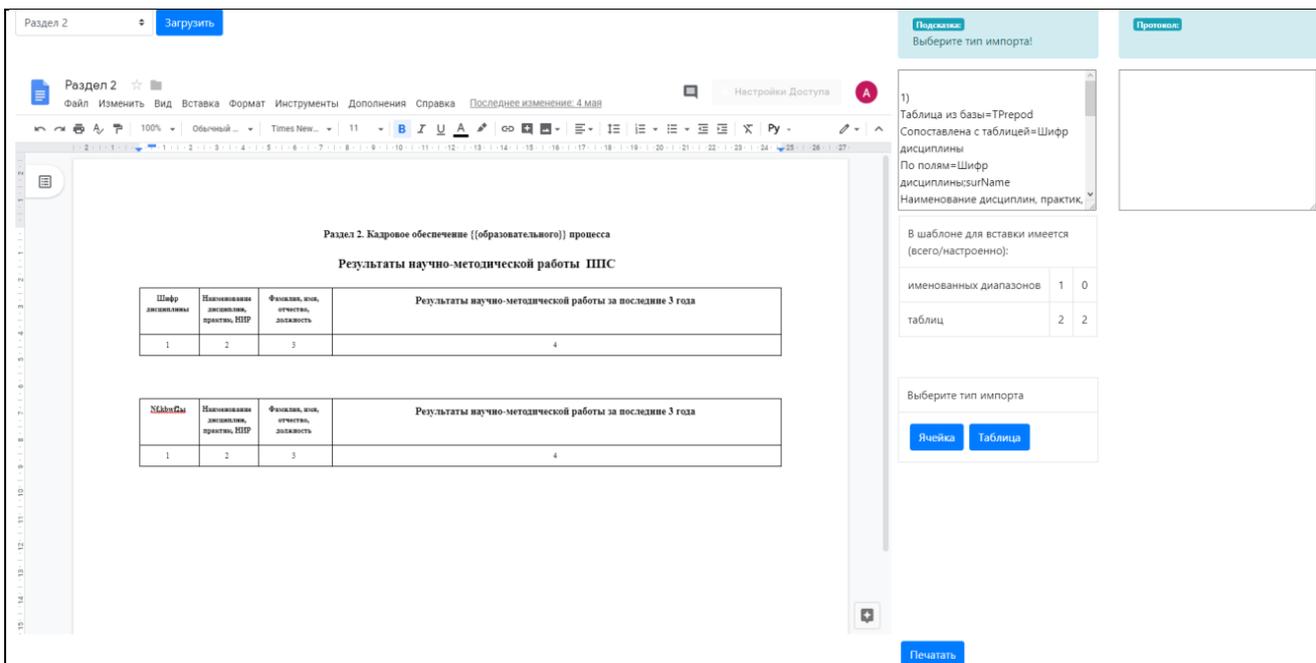


Рисунок 22 – Основной интерфейс gerort-инструмента (работа с docx)

Аналогичный вид интерфейса имеет работа с табличным редактором google-sheets.

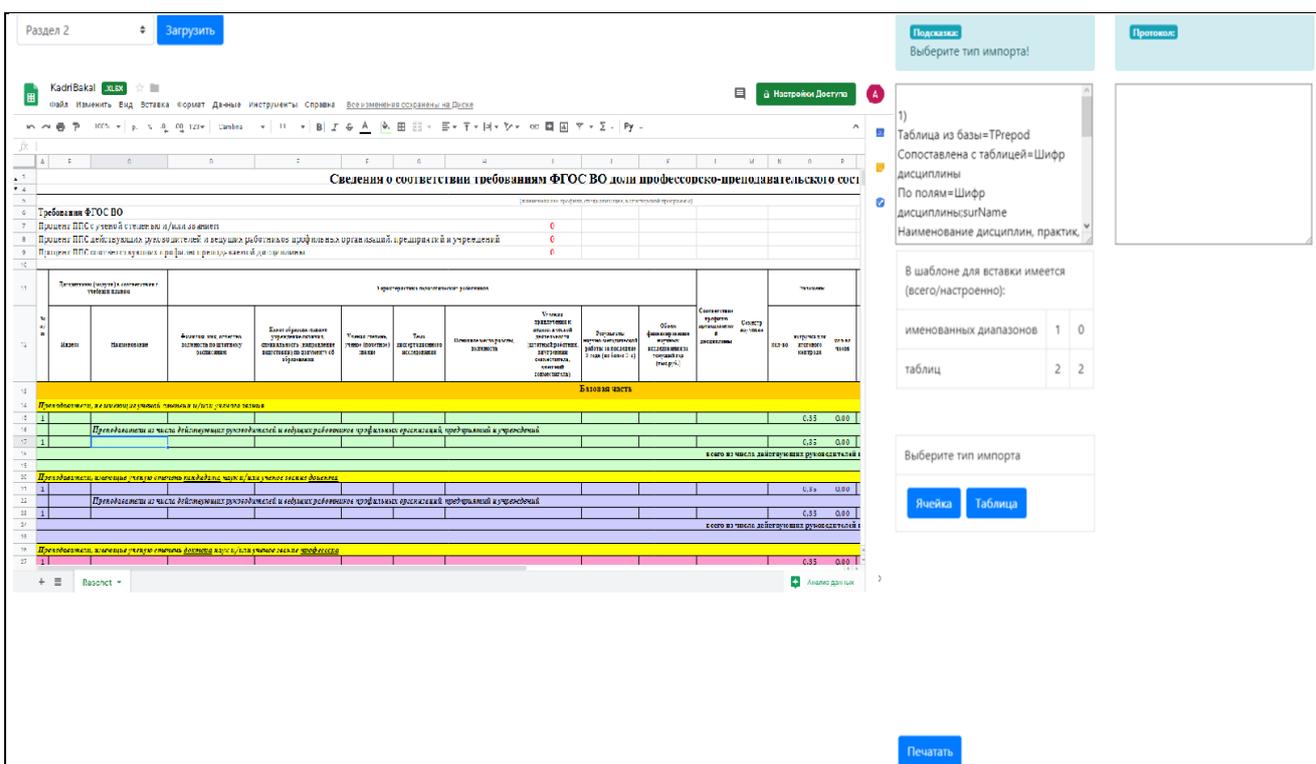


Рисунок 23 – Основной интерфейс gerort-инструмента (работа с xlsx)

Сначала пользователь выбирает таблицу, за которой стоит либо View, либо реальная таблица. Затем указывает необходимые поля, ориентируясь на всплывающие подсказки, в том числе относительно содержимого key-

содержащих полей. При этом, их можно переименовывать, а так же определять/изменять порядок их вывода. Пользовательский интерфейс характеризуется очевидностью и интуитивностью.

Следующим шагом является настройка key-содержащих полей с использованием глобального справочника. Если в системе по какой-то причине с этим полем через справочник отсутствует связь с value, то пользователь может указать в отношении ключа название, а также оба параметра – значения. И система при формировании отчета заполнит вместо ключа конкретное значение.

Предполагается, что в большинстве случаев ИТ-специалист, поддерживающий основную систему и интегрированный в нее репорт-инструмент, выполнит задачу предварительной подготовки и наполнения глобального справочника значений на подготовительном этапе – перед предоставлением в пользование report-инструмента бизнес-акторам.

Аналогично элементарно просто настраивается группировка информации в табличной части разрабатываемого самостоятельно report-инструмента. При необходимости суммирования значения какого-то поля, выбирается соответствующая функция, имеющая русскоязычный комментарий. Соответственно, система на остальные поля установит «group by» при формировании запроса.

Важно отметить, что для разных типов данных в списке доступных функций должен быть представлен их разный набор. Например, в типе данных String нет смысла выводить функцию отыскания наименьшего значения.

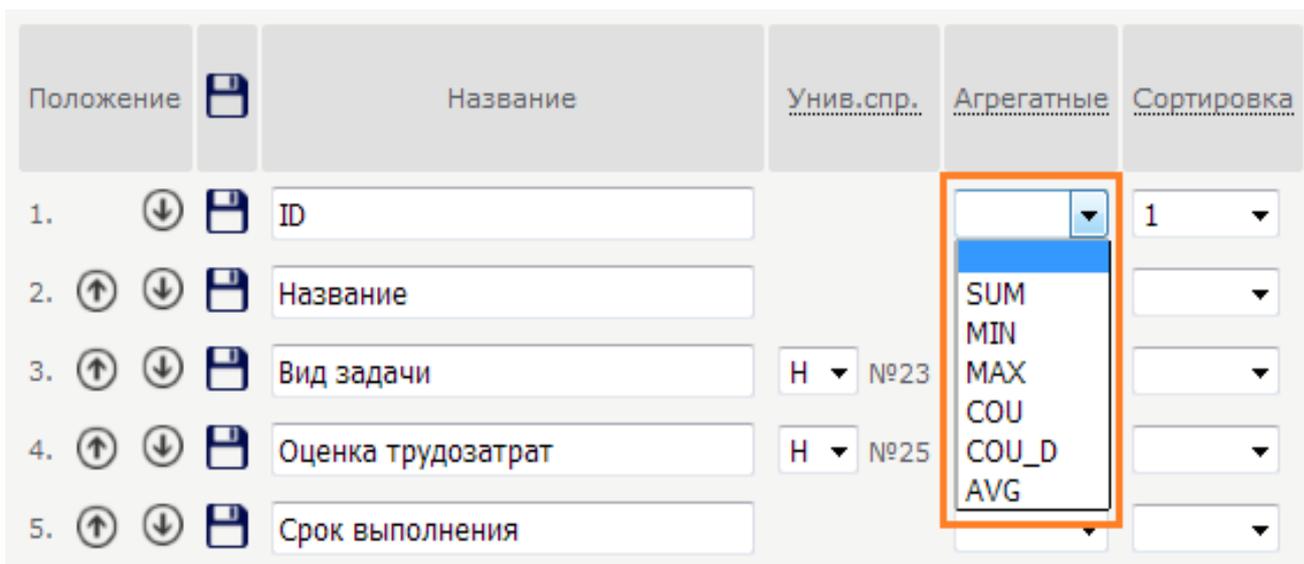


Рисунок 24 – Настройка группировки данных

В целях компактной простой реализации задуманного, целесообразно определить список сокращений для доступных функций, а также операций сравнения. В очередной раз важно отметить, что для обеспечения дружелюбности интерфейса описываемого инновационного решения требуется предусмотреть всплывающую подсказку-комментарий о заинтересовавших пользователя элементах.

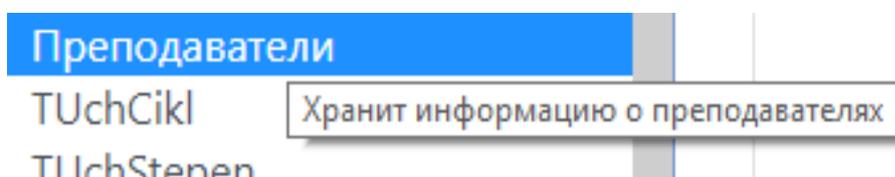


Рисунок 25 – Подсказка-комментарий

По остальным заголовкам столбцов есть аналогичная краткая информация по функционалу.

Настройка сортировки также простая и универсальная – аналогично остальным настройкам источника данных. В графе «Сортировка» выбираются поля, по которым будет осуществлена сортировка, а также порядок сортировки. Все очень просто – уровень необходимых компетенций не превышает уровень пользователя обычного табличного редактора типа Excel.

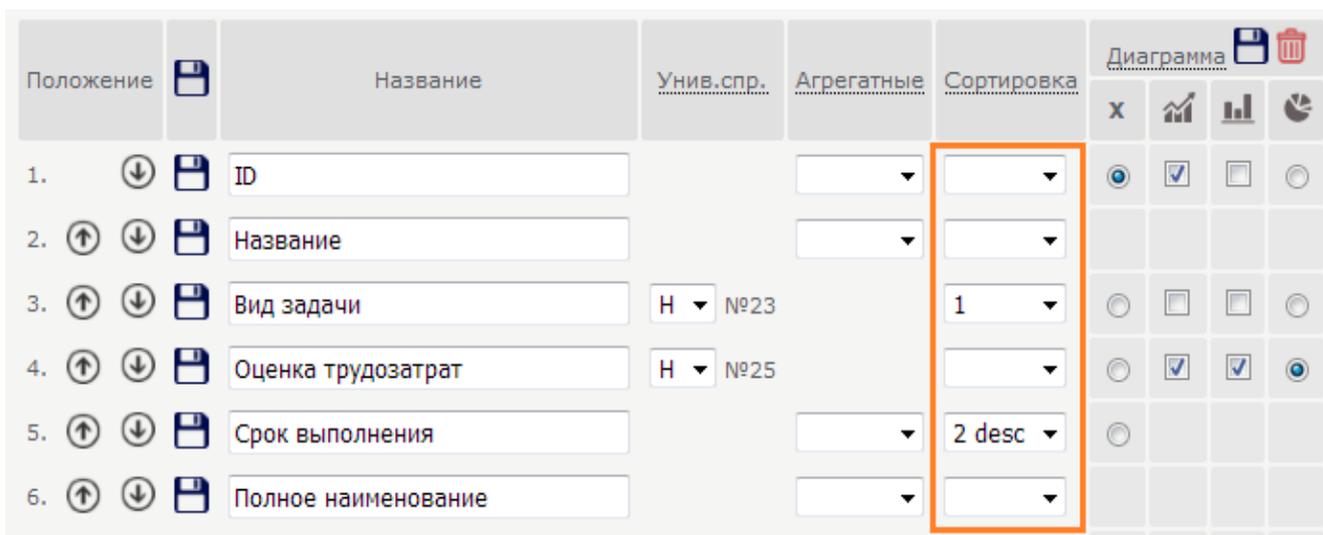


Рисунок 26 – Настройки сортировки

Очевидное преимущество данного подхода заключается в том, что пользователь может настроить сортировку любой сложности.

Все настройки выборки данных условно состоят из контейнеров и условий. Контейнер – это выражение в скобках. Условие – типовая операция сравнения ($A \geq B$, $A = B$, $A \leq B$). Условия могут быть вложенными, благодаря каскадному использованию контейнеров.

Например: выражение ($\langle \text{условиеА} \rangle$ И $\langle \text{условиеБ} \rangle$ И $\langle \text{условиеВ} \rangle$) – является контейнером И, соответственно условие ($\langle \text{условиеА} \rangle$ ИЛИ $\langle \text{условиеБ} \rangle$ ИЛИ $\langle \text{условиеВ} \rangle$) является контейнером ИЛИ. Из таких конструкций И/ИЛИ и условий внутри них можно генерировать потоки условий любой сложности.

В случае, если необходимо выбрать данные, отвечающие $\langle \text{условиеА} \rangle$ и одному из условий $\langle \text{условиеБ} \rangle$ или $\langle \text{условиеВ} \rangle$, то необходимо эти условия (2 и 3) объединить в контейнер типа ИЛИ. ($\langle \text{условие 1} \rangle$ И ($\langle \text{условие 2} \rangle$ ИЛИ $\langle \text{условие 3} \rangle$)).

В следующем примере в контейнере типа И находятся $\langle \text{условиеА} \rangle$ и контейнер $\langle \text{ИЛИ} \rangle$, а внутри контейнера – ($\langle \text{условиеБ} \rangle$ ИЛИ $\langle \text{условиеВ} \rangle$).

На интерфейсе пользователя это будет выглядеть следующим образом:

Рисунок 27 – Настройка условий

Однако, недостаточно просто создать интерфейс настройки условий. Результат операции сравнения (условия) может быть либо константой, либо переменной. Изменяющееся условие определяется просто – оно не определено явно на этапе настройки источника данных. Если значение условия Null – то требуется уточнить его у пользователя на этапе настройки отчета.

Для целей наглядности используется понятное название, которое может совпадать с именем поля либо отличаться.

В случае, если мы ввели константу в поле «Период», однако не заполнили поле «Состояние», то система спросит значение поля «Состояние» у пользователя на этапе настройки – до формирования отчета.

Рисунок 28 – Итоговый интерфейс настройки условий

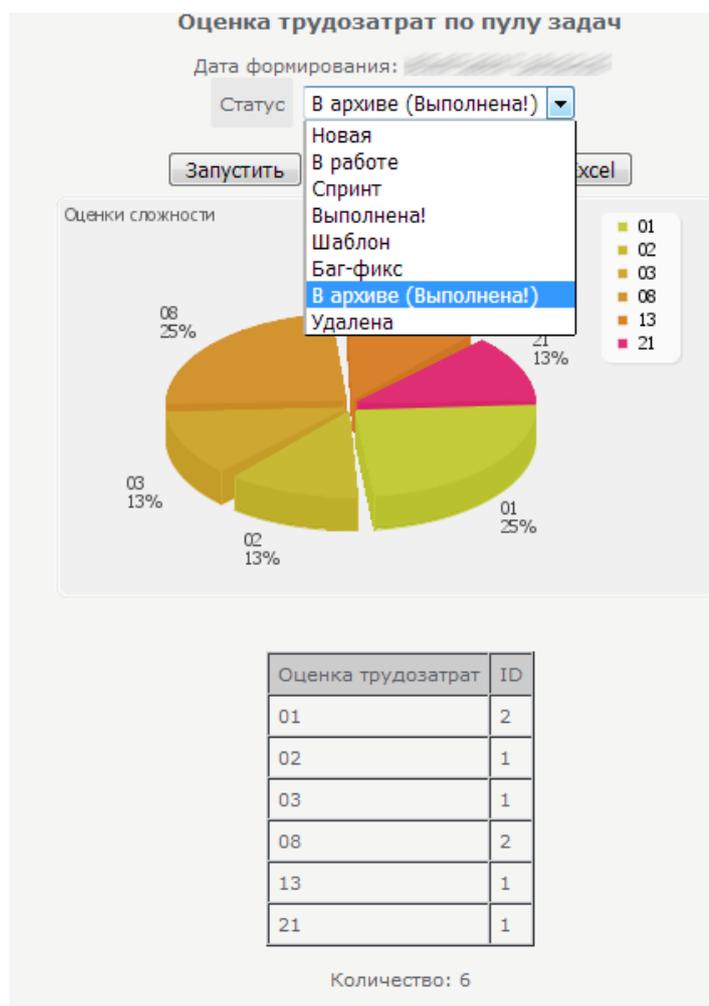


Рисунок 29 – Предпросмотр результатов настройки условий

Другими словами алгоритм такой: если значение определено явно – это константа, если Null, то необходимо спрашивать пользователя. Элегантно и элементарно.

Важно иметь в виду, что поле может быть связано с глобальным справочником значений. Тогда можно указать, какое из полей глобального справочника требуется выводить в отчет – аналогично вышеописанному примеру – вместо числового поля «Состояние» будет выведен список выбора доступных значений.

Тестовое использование разработанного программного обеспечения показало состоятельность выбранной концепции, достижимость поставленных целей и практичность найденных решений.

ГЛАВА 4 АНАЛИЗ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ И ПОЛУЧЕННОГО ПРОГРАММНОГО РЕШЕНИЯ РАЗРАБОТКИ ОТЧЕТОВ

4.1 Анализ результатов применения оптимизированного процесса по разработке отчетов

В целях всестороннего анализа результатов проведенной исследовательско-практической работы важно рассмотреть сравнительную информацию в разных аналитических разрезах.

Учитывая большое количество информации, в целях компактного изложения фактов воспользуемся табличным представлением данных по причине лаконичности и тезисности.

Для начала проведем анализ результатов оценки затрат времени на разработку типового отчета на примере поквартирной карточки, форма которой приведена после таблицы.

Таблица 14 – Сравнение процессов разработки отчетных документов

Критерии сравнения	Типичный процесс (на основе классических report-инструментов)	Двухфазный процесс на основе office-шаблона и интегрированного в документ VB-скрипта	Инновационный процесс
Степень вовлеченности программиста	Выполняет $\frac{2}{3}$ работы	Выполняет $\frac{2}{3}$ работы	отсутствует (в рамках разработки конкретного отчета)
Временные характеристики (пользователь + программист)	0,85ч. + 2,3ч.	0,85ч. + 2,25ч.	1,1ч. + 0ч.

Критерии сравнения	Типичный процесс (на основе классических report-инструментов)	Двухфазный процесс на основе office-шаблона и интегрированного в документ VB-скрипта	Инновационный процесс
Оценка когнитивной доступности для пользователя (от 1 до 10)	1	3	10
Оценка функциональности (от 1 до 10)	10	4	10
Список форматов файлов + возможности	docx, xlsx, pdf	docx, xlsx	docx, xlsx, pdf
Встраиваемость в готовую инфраструктуру	10	6	10
Наличие редактора шаблонов	имеется выбор графических редакторов		online-редакторы офисных документов (Google, Microsoft)

Критерии сравнения	Типичный процесс (на основе классических report-инструментов)	Двухфазный процесс на основе office-шаблона и интегрированного в документ VB-скрипта	Инновационный процесс
Наличие предпросмотра результата	нет	нет	Редактор в режиме просмотра (можно наблюдать заполнение риал-тайм)

Сравнительный анализ характеристик предложенного решения (и процесса) показывает явное превосходство над оппонентами.

Сравнение производилось на примере разработки поквартирной карточки представленной на рисунке 30.

Форма №10

ПОКВАРТИРНАЯ КАРТОЧКА

Улица пр-кт. Ленинский
дом 13, корп. _____, кв. 36
Телефон _____
Жилая площадь 0.0 кв. метров
Наниматель (владелец) _____
(Ф. И. О)

Республика, край, область, округ обл. Самарская
район _____
населенный пункт г. Тольятти
ЖЭО, ЖСК, домоуправление № ООО УК № 2 ЖКХ,
юридическое лицо _____

№ п/п	Отношение к нанимателю (владельцу)	Фамилия, имя, отчество (при наличии)	День, месяц и год рождения	Дата регистрации	Дата снятия с регистрац. учета
1	Собственник	Иванов Иван Иванович			
2	Собственник	Иванов Иван Иванович			
3	Племянница	Иванов Иван Иванович			
4		Иванов Иван Иванович		02.09.2011	
5	Жена	Иванов Иван Иванович		19.01.2010	12.12.2013
6		Иванов Иван Иванович		20.01.2012	
7	Мать	Иванов Иван Иванович		16.01.2007	31.10.2013

Рисунок 30 – Пример поквартирной карточки

4.2 Анализ результатов и перспектив реализованного программного решения для разработки отчетов

Определяющую роль с точки зрения удобства использования пользовательского интерфейса играет простота и понятность. Произведем сравнение редакторов шаблонов в формате таблицы.

Таблица 15 – Таблица характеристик редактора шаблонов

Наименование решения	Тип UI	Степень удобства и функциональности (от 1 до 10)	Обоснование
Типичный процесс	имеется web-инструментарий, например: FastReport, CrystalReport	3	Основное неудобство в том, что образец отчета, выполненный бизнес-актором, не может быть прямо применен в качестве шаблона, т.к. требуется разработка дизайна в специализированных редакторах дизайна шаблона. Такой подход абсолютно не доступен для простого пользователя. Используемый инструментарий полностью определяет процесс разработки каждого нового отчета.

Двухфазный процесс на основе office-шаблона и интегрированного в документ VB-скрипта	Редактор шаблона может быть запущен как онлайн-приложение без возможности интеграции скрипта заполнения	7	Удобство заключается в использовании макета отчета в качестве шаблона разрабатываемого отчета.
Инновационный процесс	web	10	Лучший редактор шаблонов – это полнофункциональный офисный редактор, который позволяет использовать привычные для простого пользователя приемы форматирования и оформления отчетного документа. Редактор в режиме просмотра позволяет в риал-тайме наблюдать процесс заполнения данными. Также позволяет подгружать шаблоны.

В результате сравнения подтверждено предполагаемое лидерство предложенного решения в части удобства и простоты интерфейса.

В целях сравнительного анализа удобства интерфейса важно оценить UX-характеристики различных подходов. Краткое сравнение представлено в таблице 18 в соответствии с методикой, предложенной известным специалистом в области разработки UX – Э. Гарретом[14].

Таблица 18 – UX-характеристики интерфейса настройки источника данных по Гарретту.

Список возможностей	Типичный	Двухфазный процесс на основе office-шаблона и вставленного в документ VB-скрипта	Иновационный
<p>1. Уровень стратегии UX: Для чего мы создаем продукт? Какие цели мы перед собой ставим? Кто наши пользователи? Для кого мы делаем продукт? Какие задачи мы поможем решить пользователям? Почему они будут пользоваться именно нашим решением?</p>	<p>Инструментарий пригоден только для использования квалифицированным и программистами. Решаются задачи, связанные с разработкой стабильных форм отчетов. Плюсом является функциональная широта и универсальность инструментария.</p>	<p>Подготовка шаблона доступна обычному пользователю с помощью удобного офисного приложения. Интересная попытка передать разработчику задумку отчета в неискаженном виде.</p>	<p>Разработанное web приложение позволяет создавать бизнес-актору отчетные документы любой сложности. Привлечение программиста для реализации конкретного отчета не требуется. ИТ-специалист должен осуществлять лишь настройку приложения, что находится за рамками процесса разработки отчета.</p>
<p>2. Уровень возможностей UX: Какие функции нужны для решения пользовательских задач?</p>	<p>Важна вся возможная функциональность. Все универсальные встраиваемые report-инструменты обладают сопоставимой функциональностью.</p>	<p>Важна возможность самостоятельно формировать простые шаблоны отчетов в word- и excel-</p>	<p>Важна полная самостоятельность разработки отчетов любой сложности. У такого решения принципиально нет конкурентов (продуктов).</p>

Список возможностей	Типичный	Двухфазный процесс на основе office-шаблона и вставленного в документ VB-скрипта	Иновационный
Какую функциональность предлагают конкуренты?		форматах. Конкурентных преимуществ практически нет. Решение сформировалось из практики конкретной организации.	
3. Уровень структуры: Как помочь пользователю решить задачу за минимальное время? Понятен ли подход (процесс работы) пользователям? Соответствует ли дизайн ожиданиям пользователей?	Бизнес-актор полностью несамостоятелен при разработке отчета, т.к. он только описывает задачу, а реализацией от начала до конца занимается программист.	Единственным способом помочь пользователю решить задачу является подготовка понятных Materialized View. Предполагается, что пользователь не в состоянии сам подготовить скрипт автозаполнения данных из источника.	Решение ориентировано на сокращение затрат ресурсов времени за счет исключения непроизводительных потерь: на коммуникации с исполнителем, на необоснованное применение трудоемких технологий и т.д. Пользователь является первоисточником знаний о задаче и мериллом корректности исполнения, а также исполнителем.

Список возможностей	Типичный	Двухфазный процесс на основе office-шаблона и вставленного в документ VB-скрипта	Иновационный
<p>4. Уровень компоновки: Как помочь пользователю решить задачу за минимальное количество шагов?</p>	<p>Процесс создания нового отчета принципиально статичен и не подлежит оптимизации. Пользователь не участвует и не имеет представления о приемах разработки отчета.</p>	<p>Потенциал оптимизации процесса разработки отчета низкий и ограничен применяемым и технологическими приемами.</p>	<p>Приложение позволяет оптимизировать сам процесс разработки, т.к. исключает программиста, как лишнего участника. UI доступен для понимания простому пользователю.</p>
<p>5. Уровень поверхности: Удобен ли интерфейс для пользователей? Решает ли он задачи, описанные на первых уровнях? Соответствует ли он тем функциям, которые в него должны быть заложены? Помогает ли дизайн пользователю?</p>	<p>Пользователь никак не взаимодействует с интерфейсом инструментария по подготовке отчета.</p>	<p>Пользователь взаимодействует лишь с интерфейсом офисного приложения на этапе подготовке шаблона. Непосредственно разработкой функционала заполнения данными занимается программист.</p>	<p>За счет инновационных интерфейсных решений приложение позволяет сократить общее время разработки. Соответствует концепции простого и полностью функционального report-инструмента, который полностью удовлетворяет потребности пользователя.</p>

Список возможностей	Типичный	Двухфазный процесс на основе office-шаблона и вставленного в документ VB-скрипта	Инновационный
Решает ли визуальное оформление задачи, которые были заложены при проектировании UX?			

Таким образом, произведено сопоставление их-характеристик интерфейса различных подходов к технической организации процесса настройки источника данных.

Важным этапом является настройка импорта данных из базы данных в созданный шаблон. Проведем в форме таблицы анализ технологичности подходов к импорту данных в отчет. Данный анализ приведён в таблице 16.

Таблица 16 – Характеристики импорта данных в отчет

Наименование решения	Наименование инструмента	Оценка скорости (1 до 10)	Удобство (связанность с популярными технологиями)
с использованием FastReport	Программная библиотека FastReport	10	Подход реализован для всех популярных языков программирования.
с использованием JasperReport	Программная библиотека JasperReport	10	Подход реализован для Java
Двухфазная технология на	Офисный редактор +	4	Подход применим для VBA.

Наименование решения	Наименование инструмента	Оценка скорости (1 до 10)	Удобство (связанность с популярными технологиями)
основе office-шаблона и интегрированного в документ VB-скрипта	VB-редактор + инструмент для подготовки View		
Инновационный (Google-окружение)	Разработано web-приложение, использующее полный функционал онлайн-редактора документов	10	Подход основан на использовании: <ul style="list-style-type: none"> - Google Docs API - Google Sheets API - Google Docs API
Инновационный (Microsoft-окружение)		10	Подход основан на возможностях .Net и вообще экосистемы Azure

В результате сравнения технологий импорта данных, используемых при заполнении шаблона определилось два явных лидера: решения от Google и Microsoft с небольшим перевесом в сторону Google, т.к. имеется возможность применения с разными языками программирования.

Для пользователя важна широта функциональности используемого инструментария. Поэтому проведем сравнение возможностей различных инструментов, используемых в разных подходах.

Таблица 17 – Общие характеристики возможностей инструментария, используемого в различных подходах разработки отчетов

Подход/ процесс/ инструмент	Виды отчетов				
	Текст	Таблица	Диа- грамма	Дэш- борды	Электрон. формы

Типичный	все популярные форматы файлов	все популярные табличные форматы файлов	графические форматы	графические форматы	нет
Двухфазный процесс на основе office-шаблона и интегрированного в документ VB-скрипта	только офисные форматы		Базовые возможности на основе офисных файлов.		в составе окружения офисного приложения
Инновационный подход с применением представленного приложения	все популярные форматы файлов	все популярные табличные форматы файлов	графические форматы и офисные форматы	графические форматы	в составе окружения офисного приложения

В результате сравнения подтверждено явное лидерство по функциональности решения, разработанного в рамках данной работы.

С точки зрения комплексности анализа достигнутых результатов важен анализ удовлетворенности исходных концептуальных требований к ПО.

Таблица 18 – Анализ итогового удовлетворения исходных требований

№	Требование	Отметка об удовлетворении требований
1.	Элементарная когнитивная доступность.	Удовлетворено. Интерфейс разработанного приложения не сложнее чем офисные приложения.
2.	Многофункциональность.	Удовлетворено. Приложение позволяет пользователю создавать

№	Требование	Отметка об удовлетворении требований
		шаблонизированные и простые табличные виды отчетов.
3.	Возможность дополнений и расширения состава данных.	Удовлетворено. За рамками процесса разработки конкретного отчета программист может подключать новые источники данных, готовить новые Materialized View.
4.	Поддержка многоуровневых (вложенных) запросов.	Удовлетворено. Реализовано за счет View и глобального справочника значений.
5.	Оперирование сложными структурами данных, в т.ч. гетерогенными.	Удовлетворено. Реализовано за счет View и глобального справочника значений.
6.	Требуется создавать все популярные виды отчетов и диаграммы.	Удовлетворено. Приложение позволяет пользователю создавать все популярные отчеты любой сложности.
7.	Требуется иметь возможность получить отчет в общедоступных форматах файлов таких, как PDF, xlsx, docx.	Удовлетворено. Приложение позволяет получать на выходе файлы отчетов в указанных форматах. Имеются технологические возможности расширения списка.
8.	Интегрируемость решения стандартными способами в различные ИС.	Удовлетворено. За рамками процесса разработки конкретного отчета программист может подключать новые источники данных, используя стандартные приемы работы.

Как и предполагалось в начале работы, ключевые потребности должны быть удовлетворены, благодаря инновационному репорт-инструменту, элементы которого описаны в таблице 19.

Таблица 19 – Элементы решения, реализующие концепцию инновационного report-инструмента

№	Концептуальное требование	Элемент решения
1.	Элементарная когнитивная доступность.	1. Авторский интерфейс настройки импорта данных в шаблон создаваемого отчета. 2. Использование в качестве редактора шаблона и вьювера результата привычных всем офисных онлайн приложений внутри интерфейса описанного в настоящей работе приложения.
2.	Многофункциональность.	Использование глобального справочника значений, а также использование всех возможных способов настройки источника данных.
3.	Возможность дополнений и расширения состава данных.	За рамками процесса разработки каждого отдельного отчета, сопровождающий ИТ-специалист может стандартными средствами добавить дополнительные базы данных.
4.	Поддержка многоуровневых (вложенных) запросов.	За рамками процесса разработки каждого отдельного отчета, сопровождающий ИТ-специалист может подготовить Materialized View и донстроить глобальный справочник значений.
5.	Оперирование сложными структурами данных, в т.ч. гетерогенными.	
6.	Требуется создавать все популярные виды отчетов и диаграммы.	В качестве основы используются возможности экосистемы приложений Google: Docs, Sheets, Forms, Analytics и т.д.
7.	Требуется иметь возможность	Использование типовых

№	Концептуальное требование	Элемент решения
	получить отчет в общедоступных форматах файлов таких, как PDF, xlsx, docx.	возможностей Google Docs API, Google Sheets API, Google Docs API.
8.	Интегрируемость решения стандартными способами в различные ИС.	За рамками процесса разработки каждого отдельного отчета, сопровождающий ИТ-специалист может стандартными средствами добавить дополнительные базы данных.

4.3 Анализ результативности проведенного исследования

Важным аспектом работы является доказанность используемых утверждений. В целях подведения итогов работы целесообразно привести информацию о ключевых посылах исследования, а также их прямые доказательства или ссылки на авторитетные источники.

Таблица 20 – Обоснованность утверждений в диссертации

№	Список утверждений, требующих обоснования	Доказательства утверждений	Источники, обосновывающие утверждения
1.	Принципиально возможно создание отчетов простым пользователем без использования труда программиста.	Доказано по результатам исследовательской работы практической реализацией web-приложения	Имеются описанные попытки создания похожего приложения с разной степенью успешности: [3], [28].
2.	Самостоятельное создание отчетов простыми пользователями может быть эффективным и	Доказано в настоящей работе в ходе сравнительного исследования различных процессов: предложенный процесс	Имеется описание процесса, подкрепленного условно-успешной разработкой похожего приложения: [3].

№	Список утверждений, требующих обоснования	Доказательства утверждений	Источники, обосновывающие утверждения
	способно сократить затраты.	занимает в 3 раза меньше времени по сравнению с классическим подходом.	
3.	Создание отчетов простыми пользователями возможно не только с помощью специализированных встроенных report-инструментов.	Реализованная на практике концепция предложенного в данной работе приложения предполагает использование пользователем-непрограммистом. При этом инструмент универсальный интегрируемый.	Проанализированные источники приводят примеры только расширения возможностей не универсальных report - инструментов. За исключением попытки описанной в источнике.
4.	Интегрируемые report -инструменты могут предоставить функциональность не хуже встроенных (платформенных), не потеряв доступность для пользователя без навыков программирования.	Реализованное в ходе данной работы успешно комбинирует лучшие интерфейсные и архитектурные наработки с инновационными и эвристическими UI/UX-решениями. Реализована полная функциональность без потери когнитивной доступности для простого пользователя.	Имеются описанные попытки создания похожего приложения с разной степенью успешности: [3], [28].

№	Список утверждений, требующих обоснования	Доказательства утверждений	Источники, обосновывающие утверждения
5.	Известные доступные report - инструменты не отвечают современным потребностям бизнес-акторов в самостоятельной подготовке отчетов.	В настоящей работе проведен анализ истинных потребностей бизнес-акторов по процессу подготовки отчетных документов.	В работе упоминаются результаты тематических исследований: [7], [8].
6.	Требуемый репорт-инструмент может быть разработан на основе широко распространенных подходов и технологий.	При разработке решения, представленного в настоящей работе использовано: Google Docs API, Google Sheets API, Google Docs API, популярные библиотеки java и т.д. Разработка авторских низкоуровневых базовых технологий не потребовалось.	Имеются описанные примеры создания похожего приложения с применением разных, но известных подходов: [3], [2].
7.	Предлагаемое техническое решение может иметь коммерческие перспективы развития.	Варианты коммерческого развития предлагаемого решения, а также комплексная оценка экономической эффективности предлагаемого решения представлены в	Описанные попытки создания похожего приложения не преследуют коммерческие цели и направлены на локальное применение в организации.

№	Список утверждений, требующих обоснования	Доказательства утверждений	Источники, обосновывающие утверждения
		соответствующих таблицах.	

Аспекты, подтверждающие экономическую значимость результатов работы приведены в таблице 21.

Таблица 21 – Подтверждение экономической значимости работы

№	Тезис	Обоснование
1.	Результаты работы могут заинтересовать сообщество профессионалов и энтузиастов развивать данный опыт и направление знаний в целях исключения необоснованной сложности при разработке информационных систем.	Представленное в настоящей работе приложение решает конкретные организационно-технические проблемы различных компаний.
2.	Результаты работы могут мотивировать IT-предпринимателей активнее заниматься изыскательской деятельностью с последующей разработкой и продвижением новых нишевых программных продуктов для широкого круга пользователей.	Основой представленного web-приложения является обоснованная комбинация уже известных технологий и эвристических UI/UX решений, что привлекательно с точки зрения запуска стартапа. Анализ коммерческого потенциала приведен в настоящей работе.
3.	Практическая значимость проявляется при анализе целесообразности использования нового подхода в разрезе	Представленное web-приложение позволяет оптимизировать процесс разработки новых отчетов, сократив время разработки в три

№	Тезис	Обоснование
	экономической эффективности.	раза. Анализ приводится в настоящей работе.
4.	Исследование имеет значение для оптимизации деятельности предметных специалистов, нуждающихся в самостоятельном режиме использования report-инструментария.	Повышение эффективности деятельности бизнес-акторов достигается за счет возможности разрабатывать отчеты полностью самостоятельно, исключая таким образом непроизводительные потери. В соответствии с популярными практиками управления процессами, непроизводительные потери характеризуются затратами любых ресурсов без получения пользы, т.е. значимого для потребителя результата. Например, устраняется потеря времени на вербальные и письменные коммуникации с программистом.
5.	Значимость работы проявляется при устранении обычной организационной инертности, связанной с привлечением программистов (негативно влияет очередность выполнения задач, необходимость предварительного четкого формулирования задачи и контроля корректности исполнения).	

К подтверждению экономической значимости целесообразно добавить комплексную оценку экономической эффективности предлагаемого решения, которая приводится в таблице 22.

Таблица 22 – Комплексная оценка экономической эффективности предлагаемого решения

№	Наименование показателя	Значение до внедрения	Влияющий фактор	Значение в случае внедрения
1.	Общее время разработки типового отчета на примере поквартирной	3 часа – суммарное время работы программы	Внедрен инновационный процесс разработки отчетных документов с помощью web-	суммарное время – 1 час самостоятельной работы бизнес актора.

№	Наименование показателя	Значение до внедрения	Влияющий фактор	Значение в случае внедрения
	карточки паспортной службы УК в сфере ЖКХ (образец представлен на скриншоте).	ста и бизнес-актора	приложения, предложенного и описанного в настоящей работе.	
2.	Количество различных отчетов типовой сложности, которые могут быть подготовлены в единицу времени (3 часа).	1 отчет	Исключены непроизводительные потери в виде необоснованной сложности подхода к разработке каждой формы отчета, а также связанные с этим временные затраты.	3 отчета
3.	Косвенные положительные экономические эффекты в результате увеличения глубины анализа	Привычная продуктивность аналитиков	Возможность строить отчеты в различных представлениях и аналитических разрезах	Трехкратное превосходство в продуктивности

Качественная оценка экономической эффективности может быть подкреплена описанием вариантов коммерческого развития предлагаемого решения, о чем приводится информация в таблице 23.

Таблица 23 – Варианты коммерческого развития предлагаемого решения

№	Вариант	Краткое обоснование
1.	Предложение онлайн-сервиса подготовки отчетов,	Позиционирование: нишевый продукт для эффективных

№	Вариант	Краткое обоснование
	распространяемого по подписке.	компаний. Подписка освобождает от значительных единовременных финансовых затрат, т.е. предполагается лишь абонентская плата за пользование сервисом.
2.	Коробочная версия продукта.	Для корпоративных и гос.клиентов, не желающих расшаривать свои данные, но нуждающихся в динамичных возможностях современных IT-решений.
3.	Education-варианты распространения для сферы образования.	Бесплатная подписка для учебных заведений и студентов (учеников).

В результате проведения комплексного сравнительного анализа установлено, что проведенное исследование имеет реальную теоретическую и практическую пользу по причине необходимости для бизнеса преодоления явных организационно-технических проблем, связанных с применением необоснованно сложного инструментария подготовки отчета. Подтверждено наличие коммерческих перспектив развития предложенного программного продукт, дано экономическое обоснование полезности его внедрения. В результате реализации предлагаемого решения выполнены требования, включая доступность инструмента для самостоятельной работы бизнес-акторов. Достигнуты значительные конкурентные преимущества разработанного программного обеспечения.

ЗАКЛЮЧЕНИЕ

В ходе работы над магистерской диссертацией была проанализирована литература и электронные интернет-источники, касающиеся качества поддержки организационных и технических вопросов организации деятельности по подготовке отчетов.

В результате анализа тематической информации было установлено, что разработка инновационных инструментов создания отчетов является перспективным направлением в области организации управленческой деятельности (управление процессами). Анализ позволил установить недостаточную гибкость использования современных технологий в виду инертности профессионального опыта ИТ-специалистами.

Были подвергнуты анализу основные варианты процессов создания отчетных документов, что показало их непригодность для современных условий бизнес-среды.

Разработана модель web-приложения на основе определенных концептуальных требований, позволяющих изменить сам процесс. Реализовано программное обеспечение, предоставляющее реальный инструмент, явно меняющий процесс и создающее тем самым предпосылки для экономии ресурсов.

В тестах и экспериментах использовались реальные данные, которые имеют место в конкретной компании, а также свойственны в значительной степени другим организациям. Проведенный анализ результатов показал соответствие их целям оптимизации всех составляющих работы по созданию отчетов. Затраты времени, понесенные пользователем при использовании программного продукта, показывает положительное соотношение «было-стало», а значит очевидную полезность разработанного решения.

Тестирование предложенного программного обеспечения показало положительное количественное изменение метрик, а также достижение

качественных результатов, так как позволяет своевременно создавать отчетные документы заинтересованным в этом лицам (бизнес-акторам).

Первостепенный научный результат магистерской диссертации заключается в том, что разработанный инновационный процесс самостоятельной разработки отчетов, подкрепленный функционалом разработанного программного обеспечения, позволяет повысить качество и глубину аналитической работы, которых можно достичь благодаря высвобождению ресурсов (рабочего времени).

Эксплуатация реализованного программного обеспечения дает следующие преимущества: увеличение скорости разработки каждого нового отчета; беспроблемное интуитивно понятное взаимодействие с пользовательским интерфейсом, а значит и самостоятельность работы; возможность интенсивной самостоятельной работы; оперативное реагирование на запросы бизнес-процессов организации.

Таким образом, внедрение в организациях предлагаемого организационно-технического решения является эффективным, а также экономически обоснованным.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Научная и методическая литература

1. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектноориентированный анализ и проектирование / Д. Арлоу. – 2-е изд. (пер. с англ.) – СПб: Символ-Плюс, 2014. – 624 с.
2. Белов, Д.Е. Разработка модуля генерации отчетности, позволяющего экспортировать данные в форматы pdf, xls, doc / Д.Е. Белов, А.Ф. Шалин, Кузнецов И.М., Макеев М.В. // Сборник научных трудов Всероссийского научно-исследовательского института овцеводства и козоводства . 2013.
3. Большаков, С.А. Двухфазная технология формирования выходных документов в информационных системах на основе MS Word / С.А. Большаков, С.Б. Спиридонов // Наука и образование МГТУ им. Н.Э. Баумана. – 2015. – № 8. – С. 252-268.
4. Веллингтон, Л. Разработка веб-приложений с помощью PHP и MySQL 4-е изд. – 2016. – 1025с.
5. Вон, В. Реализация методов предметно-ориентированного проектирования / В. Вон. – Вильямс, 2018. – 688 с.
6. Грекул В.И. Проектирование информационных систем / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. – М.: Интернет-Уни-т информ. технол., 2015. – 304 с.: ил.
7. Ивненко, А.С. Оптимизация документационного обеспечения образовательного процесса кафедры средствами разработки автоматизированной информационной системы документационного обеспечения / А.С. Ивненко // Информационные технологии в моделировании и управлении: подходы, методы, решения: Сборник научных статей I Всероссийской научной конференции: 12-14 декабря 2017 г. В двух частях. – Тольятти: Издатель Качалин Александр Васильевич, 2017. – С. 100-106.
8. Ивненко, А.С. Обзор классических подходов, используемых при проектировании подсистемы отчетов в информационных системах / А.С.

Ивненко // Передовые инновационные разработки. Перспективы и опыт использования, проблемы внедрения в производство. Сборник научных статей по итогам третьей международной научной конференции (30 апреля 2019 г.). Часть 2. – Казано: ООО «Конверт», 2019. – С. 73-75.

9. Фаулер, М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер – М.:Символ-Плюс, 2018. – 600 с.

10. Хрусталева, Е.Ю. Облачные технологии «1С:Предприятия» / Е.Ю. Хрусталева. – М.:1С-Пабблишинг, 2016. – 217 с.

11. Положение об основной профессиональной образовательной программе высшего образования Тольяттинского государственного университета. – Тольятти: ТГУ. – 2017.

12. Пенькова, Т. Г. Модели и методы оперативного формирования документов / Т.Г. Пенькова // Вычислительные технологии. – 2009. -№ 2 – С. 98-109.

Электронные ресурсы

13. 1С-Битрикс [Электронный ресурс]. – Режим доступа: <https://www.1c-bitrix.ru/products/intranet/features/reports.php>

14. 5 уровней UX: как создавать удобные интерфейсы [Электронный ресурс]. – Режим доступа : <https://netology.ru/blog/5level-ux>

15. Busy Developers Guide to HSSF and XSSF Features [Электронный ресурс]. – Режим доступа : <https://poi.apache.org/components/spreadsheet/quick-guide.html>

16. Crystal Reports [Электронный ресурс]. – Режим доступа : <http://www.interface.ru/fset.asp?Url=/crystal/crystalh.htm>

17. Службы SQL Server Reporting Services (SSRS) [Электронный ресурс]. – Режим доступа : <https://docs.microsoft.com/ru-ru/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-2017>

18. Documentation JasperSoft Community [Электронный ресурс]. – Режим доступа: <https://community.jaspersoft.com/documentation?version=>
19. FastReport – генератор отчетов [Электронный ресурс]. – Режим доступа: <https://www.fastreport.ru/>
20. Java Persistence API [Электронный ресурс]. – Режим доступа: <http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>.
21. Oracle Technology Network. JavaServer Pages Technology [Электронный ресурс]. – Режим доступа: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>
22. Spring Boot [Электронный ресурс]. – Режим доступа: <https://projects.spring.io/spring-boot/>
23. Stimulsoft [Электронный ресурс]. – Режим доступа: <https://www.stimulsoft.com/ru>
24. The Apache OpenOffice API Project [Электронный ресурс]. – Режим доступа : <http://www.openoffice.org/api/>
25. User-Friendly Programming-Fre Reporting for Automation [Электронный ресурс]. – Режим доступа : https://fiord.com/images/industry_avt/soft/dream_report/Dream-Report-Getting-Started-RUS.pdf
26. Генерация отчетов. Crystal Reports или Веб-технологии? [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/36748/>
27. Диаграмма Исикавы – полезный инструмент в жизни и работе User-Friendly [Электронный ресурс]. – Режим доступа : <https://4brain.ru/blog/диаграмма-исикавы/>
28. Каким должен быть идеальный конфигуратор отчетов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/331884/>
29. Мегаплан [Электронный ресурс]. – Режим доступа: <https://help.megaplan.ru/article/reports>
30. Мультиплатформенная система электронного документооборота и ее преимущества [Электронный ресурс]. – Режим доступа:

http://www.intertrust.ru/press_center/articles/view/824-multiplatformennaya-sistema-elektronnogo.htm

31. Сервис 1С:ИТС Информационная система [Электронный ресурс]. – Режим доступа : <https://its.1c.ru>

32. Сервис визуализации и анализа данных [Электронный ресурс]. – Режим доступа : <https://cloud.yandex.ru/services/datalens>

33. Анализ клиентов и отчетность с помощью Zendesk Explore [Электронный ресурс]. – Режим доступа : <https://www.zendesk.com.ru/explore/>

34. YARG — open-source библиотека для генерации отчётов [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/company/haulmont/blog/224125/>

35. Вышла новая версия Apache POI 3.8 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/140898/>

Литература на иностранном языке

36. Charmy Patel, Ravi Gulati. Software Performance Testing Measures. International journal of management & information technology, Vol. 8, 2014.

37. Eljona Proko, Iia Ninka. Analysis and Strategy for the Performance Testing in Cloud Computing. Global Journal of Computer Science and Technology, Volume 12, 2013.

38. Joensson W. Iconix: Exceptional Product Design / W. Joensson – Skyhorse, 2018. 264 с.

39. Lesyuk, A. Mastering Redmine / A. Lesyuk – Packt, 2013. 366 с.

40. Sanderson, S. Pro ASP.NET MVC 3 Framework, Third Edition / S. Sanderson, A. Freeman. – Apress, 2012. – 837 с.

41. Scott, B. Tberesa, N. Designing Web Interfaces: Principles and Patterns for Rich Interactions / B. Scott, N. Tberesa – O'Reilly Media, 2009. – 334 с.

42. Surhone, L. Apache Poi / L. Surhone, M. Tennoe , S. Henssonow – Betascript Publishing, 2010. – 120 с.

43. Turley, P. Microsoft SQL Server Reporting Services Recipes: for Designing Expert Reports / P. Turley, R. Bruckner – Wrox, 2010. 648 с.