

**Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет аэрокосмического
приборостроения»**

На правах рукописи

Дмитриев Павел Игоревич

**МЕТОДЫ И СРЕДСТВА УПРАВЛЕНИЯ ЗНАНИЯМИ В БАЗОВЫХ
ПРОЦЕССАХ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СРЕДСТВ**

05.02.23 – Стандартизация и управление качеством продукции

ДИССЕРТАЦИЯ
на соискание ученой степени
кандидата технических наук

Научный руководитель
д. т. н., доц.
Вершинина Л. П.

Санкт-Петербург – 2014

ОГЛАВЛЕНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	5
ВВЕДЕНИЕ	7
1. УПРАВЛЕНИЕ КАЧЕСТВОМ И ЗНАНИЯМИ НА СОВРЕМЕННЫХ ПРЕДПРИЯТИЯХ.....	14
1.1. Назначение и актуальность управления знаниями	14
1.1.1. Понятие и классификация знаний.....	14
1.1.2. Процесс управления знаниями	20
1.2. Ресурсы знаний предприятия.....	26
1.3. Управление качеством процессов разработки программного обеспечения.....	32
1.3.1. Стандарты в области разработки программного обеспечения	32
1.3.2. Процессы системы менеджмента качества	33
1.3.3. Базовые процессы жизненного цикла программных средств	36
1.3.4. Критерии оценки базовых процессов жизненного цикла программных средств.....	43
1.3.5. Разработка и оформление документации на программные средства. 46	
1.4. Системы управления знаниями.....	55
1.4.1. Классификация систем управления знаниями.....	55
1.4.2. Архитектура современной системы управления знаниями	68
1.5. Модели представления знаний	70
1.5.1. Классификация и основные типы моделей представления знаний....	70
1.5.2. Онтологический инжиниринг.....	78
1.6. Постановка цели и задач исследования	89
2. НАУЧНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ ЗНАНИЯМИ В БАЗОВЫХ ПРОЦЕССАХ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СРЕДСТВ.....	92
2.1. Метод структуризации информации	92
2.2. Комплекс онтологических моделей представления знаний	98

2.3. Методика создания системы управления знаниями	120
2.4. Выводы к главе 2	123
3. НАУЧНО-ТЕХНИЧЕСКИЕ ПРЕДЛОЖЕНИЯ ПО АВТОМАТИЗАЦИИ И СТАНДАРТИЗАЦИИ ПРОЦЕССА УПРАВЛЕНИЯ ЗНАНИЯМИ	124
3.1. Модель системы управления знаниями	124
3.2. Компоненты программно-алгоритмического обеспечения системы управления знаниями	130
3.2.1. Подсистема поиска	130
3.2.2. Подсистема компоновки документов	139
3.2.3. Подсистема разграничения доступа	143
3.2.4. Подсистема целостности онтологии	148
3.3. Методика оценки качества базовых процессов жизненного цикла программных средств при внедрении системы управления знаниями	154
3.3.1. Расчет комплексного показателя качества процесса	154
3.3.2. Оценка трудозатрат	161
3.3.3. Оценка качества процесса поиска информации о программной продукции	162
3.4. Выводы к главе 3	167
ЗАКЛЮЧЕНИЕ	169
СПИСОК ЛИТЕРАТУРЫ	171
Приложение А Описание концептов онтологии документации на программные средства	186
Приложение Б – Схемы eTOM и TAM концепции NGOSS	197
Приложение В – Описание концептов онтологии продукции	199
Приложение Г – Стандарт организации «Система управления знаниями о программной продукции»	203
Приложение Д – Примеры реализации обращений к Web-службам Microsoft SharePoint	217
Приложение Е – Таблица XSLT-стилей для преобразования XML-документа модулей документации	225

Приложение Ж – Шаблон для формирования руководства системного программиста в HTML-формате	230
Приложение З – Структура списков подсистемы разграничения доступа.....	238
Приложение И – Определение относительных весов показателей качества .	239
Приложение К – Свидетельство о государственной регистрации программы для ЭВМ.....	242

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

BSS – Business Support System

CMMI – Capability Maturity Model Integration

eTOM – enhanced Telecom Operations Map

ITIL – IT Infrastructure Library

OSS – Operation Support System

RUP – Rational Unified Process

SEI – Software Engineering Institute

SPICE – Software Process Improvement Capability Determination

TQM – Total Quality Management

XP – eXtreme Programming

ГП – группа продуктов

ИСО – Международная организация по стандартизации
(International Organization for Standardization, ISO)

МД – Модуль документации

ПО – Программное обеспечение

ПП – Программный продукт

ПС – Программные средства

РП – Рабочий процесс

СМК – Система менеджмента качества

СТО – Стандарт организации

СУЗ – Система управления знаниями

СУЗПП – Система управления знаниями о программной продукции

УЗ – Управление знаниями

ФВ – Функциональная возможность

ВВЕДЕНИЕ

Актуальность работы. Сегодня управление знаниями представляет собой новый метод менеджмента, который изменяет подходы организаций к достижению конкурентоспособности на основе обеспечения высокого качества продукции, в которых эффективно функционирует система менеджмента качества (СМК) [8, 81, 122]. Знания, интеллектуальный капитал, интеллектуальная собственность рассматриваются как ценный ресурс, определяющий производительность, жизнеспособность и успешность организаций [27, 111, 116, 118]. Эта общая тенденция приобретает особое значение для предприятий – разработчиков программного обеспечения (ПО) в связи с быстрой сменой технологий, большим ростом источников и объема информации и знаний. Предприятия, которые быстрее других получают и генерируют знания, внедряют их в ключевые бизнес-процессы, обеспечивают их хранение и усвоение сотрудниками, в итоге повышают качество производимой программной продукции, обеспечивают себе устойчивый рост и приобретают несомненные конкурентные преимущества. В связи с этим возникает необходимость в эффективном управлении знаниями и создании систем поддержки этого процесса.

Важное место среди процессов предприятия – разработчика ПО занимают базовые процессы жизненного цикла программных средств (ЖЦ ПС), связанные с определением требований, проектированием, разработкой и сопровождением программной продукции, так как они являются наиболее трудоёмкими, и качественное их выполнение во многом определяет способность организации выполнять требования, предъявляемые потребителем к поставляемой продукции. При этом быстрый и удобный доступ к знаниям, которые используются и создаются при реализации конкретных работ (задач), играет первостепенную роль, поскольку позволяет значительно повысить качество и эффективность указанных процессов (уменьшить количество

ошибок, затраты на разработку ПС и анализ сообщений от заказчиков и т.д.). Противоречие заключается в том, что существующие методы и средства доступа не могут обеспечить достаточный уровень полноты при поиске необходимых знаний, а изучение специалистами большого объема информационных ресурсов (например, документации на ПС) требует больших временных затрат.

В решение задач повышения качества производственных процессов и управления качеством применительно к продуктам, услугам и процессам, в том числе разработки ПС, большой вклад внесли такие ученые, как А.Г. Варжапетян, Е.Г. Семенова, Г.И. Коршунов, В.А. Липатников, А.П. Ястребов, Н.Н. Рожков, В.В. Бураков, Я.А. Ивакин, В.В. Липаев [71-74], А.Н. Терехов [108], Р. Фатрелл [120], И. Соммервилл [106], К. Полк [164-166] и др. Проблеме оценки качества и управления качеством ПС посвящены фундаментальные работы Б. Боэма, Т. Джилба, М. Холстеда. Огромный вклад в развитие данной проблематики внесли такие организации, как ISO, SEI, Microsoft, IBM, Oracle и др. Тем не менее, за рамками разработанных подходов остался вопрос использования методов и средств управления знаниями в управлении качеством разработки ПС. Существующие стандарты (как специфические для области разработки ПС, так и общие стандарты по управлению качеством) документально закрепляют используемые в организациях производственные знания и предлагают некоторые действия по управлению ими, однако сам процесс управления знаниями в явном виде не регламентируют.

Сегодня темой большого количества теоретических и практических исследований является тема построения автоматизированных систем управления знаниями (СУЗ), которые реализуют весь комплекс функций работы со знаниями [15, 17, 50, 111, 116]. Большой вклад в разработку подходов для построения СУЗ внесли российские и зарубежные ученые Э.В. Попов [56, 116], Т.А. Гаврилова [13-26], А.Ф. Тузовский [109-115], А.С.

Клещев [60-63], Б.В. Добров [48, 89], Д. Уотермен, Э. Хант, Р. Майер [158], А. Гомес-Перес [145], Т. Грубер [146, 147], Д. Фенсел [141, 159], А. Абекер [124, 155] и др. Несмотря на большой и растущий объем публикаций о СУЗ, их проблематика посвящена, в основном, разработке методов и средств, реализующих отдельные функции и сервисы в операциях с информацией и знаниями узких предметных областей. Анализ литературы по данной тематике показал, что на сегодняшний день не существует СУЗ, которая может быть внедрена в базовые процессы ЖЦ ПС на предприятии – разработчике ПО путем простой инсталляции готового решения, а также недостаточно исследований, посвященных структурам корпоративных СУЗ, которые можно было бы использовать для построения СУЗ на таком предприятии.

Таким образом, исследования, связанные с разработкой и применением методов и средств управления знаниями для повышения качества базовых процессов ЖЦ ПС, безусловно, являются актуальными.

Цель работы. Цель диссертационной работы заключается в повышении качества базовых процессов жизненного цикла программных средств на основе разработки и применения методов и средств управления знаниями.

Задачи исследования. Для достижения цели были поставлены и решены следующие основные задачи:

1. Разработка метода структуризации информации в документации на ПС на основе онтологических моделей представления знаний в базовых процессах ЖЦ ПС.
2. Разработка методики создания СУЗ о программной продукции (СУЗПП), обеспечивающей управление знаниями и повышение качества базовых процессов ЖЦ ПС.
3. Разработка модели и прототипа СУЗПП.
4. Разработка методики оценки качества базовых процессов ЖЦ ПС при внедрении СУЗПП.

Методы исследований. В диссертационной работе использованы методы инженерии знаний, системного анализа, экспертных оценок, процессного подхода, принципов менеджмента качества, управления проектами, теории алгоритмизации и программирования.

Тематика работы соответствует областям исследования: 3. «Методы стандартизации и менеджмента (контроль, управление, обеспечение, повышение, планирование) качества объектов и услуг на различных стадиях жизненного цикла продукции», 4. «Квалиметрические методы оценки качества объектов, стандартизации и процессов управления качеством», 5. «Методы стандартизации и управления качеством в CALS-технологиях и автоматизированных производственных системах» паспорта специальности 05.02.23.

Основные результаты, выносимые на защиту:

1. Метод структуризации информации в документации на ПС на основе разработанных онтологических моделей представления знаний в базовых процессах ЖЦ ПС.
2. Методика создания СУЗПП, обеспечивающая управление знаниями и повышение качества базовых процессов ЖЦ ПС.
3. Модель СУЗПП, основанной на синтезе компонент информационно-поисковых систем и систем искусственного интеллекта.
4. Методика оценки качества базовых процессов ЖЦ ПС при внедрении СУЗПП.

Научная новизна диссертационной работы заключается в следующем:

1. Новизна метода структуризации информации в документации на ПС на основе онтологических моделей представления знаний в базовых процессах ЖЦ ПС состоит в том, что разработанные метод и модели позволяют развить известный аппарат инженерии знаний путем разработки понятийно-смысловой структуры документации на ПС и обеспечить интеграцию знаний в базовых процессах ЖЦ ПС.

2. Впервые разработана и апробирована методика создания СУЗПП, основанная на разработанных методе и моделях, позволяющая внедрить процесс управления знаниями в базовые процессы ЖЦ ПС для повышения их качества.
3. Разработана модель СУЗПП, отличающаяся тем, что в её основе лежит комбинация двух известных типов СУЗ, а также составом компонент, что позволяет учесть семантику информации, находящейся в документации на ПС, взаимосвязи и взаимозависимости, обеспечить целостность и непротиворечивость знаний, повысить качество поиска информации, обеспечивая тем самым повышение качества базовых процессов ЖЦ ПС.
4. Впервые разработана и апробирована методика оценки качества базовых процессов ЖЦ ПС при внедрении СУЗПП, основанная на применении методов оценки качества процессов ЖЦ ПС, экспертных оценок и оценки качества поисковых систем.

Обоснованность и достоверность. Обоснованность научных результатов обусловлена корректным использованием апробированного в научной практике исследовательского и аналитического аппарата. Достоверность результатов проведенных исследований обеспечивается использованием современных методик обработки исходной информации и подтверждается совпадением результатов исследования с экспериментальными данными, практической реализацией на предприятиях.

Практическая значимость. Полученные в диссертационной работе результаты обеспечивают:

- повышение качества базовых процессов жизненного цикла ПС: для процесса проектирования и разработки ПС (подпроцесс разработки документации на ПС) значение комплексного показателя качества увеличилось на 25%, для процесса сопровождения (подпроцесс мониторинга и анализа сообщений потребителей) – на 21%;

- повышение качества документации на ПС за счет появления новых потребительских свойств (добавлен гипертекст, метаинформация, автоматическое формирование документов);
- повышение качества процесса поиска информации о программной продукции (повышены показатели полноты и точности поиска на 44% и 5% соответственно);
- уменьшение трудозатрат на разработку документации на ПС (по подсистемам до 20%, полностью генерируемой общей документации по группам продуктов до 90%);
- уменьшение трудозатрат на сопровождение (среднего времени на анализ и выработку решения по сообщению с ошибкой на 13%, среднего времени на ответ по сообщению с вопросами по эксплуатации ПС на 42%);
- активизацию обмена знаниями между сотрудниками.

Разработано информационное и программно-алгоритмическое обеспечение подсистем СУЗПП. Разработан типовой стандарт организации «Система управления знаниями о программной продукции», регламентирующий правила работы с системой, роли пользователей системы и основные операции, которые доступны в рамках роли. Стандарт прошел внедрение и утверждение на предприятии ЗАО «Петер-Сервис» (г. Санкт-Петербург) в 2013 году.

Личный вклад автора состоит в непосредственной разработке метода структуризации информации, находящейся в документации на ПС, комплекса онтологических моделей представления знаний в базовых процессах ЖЦ ПС, методики создания СУЗПП, методики оценки качества базовых процессов ЖЦ ПС при её внедрении. Автором также самостоятельно разработаны модель СУЗПП и её прототип, типовой стандарт организации «Система управления знаниями о программной продукции». Основные научные результаты и выводы, содержащиеся в диссертации, получены автором самостоятельно.

Соискатель принимал непосредственное участие в апробации и внедрении результатов исследования, подготовке публикаций по теме диссертации.

Реализация работы. Результаты диссертационной работы были протестированы и внедрены при создании СУЗПП на предприятиях ЗАО «Петер-Сервис» (г. Санкт-Петербург) и ОАО «НТЦ ЕЭС» (г. Санкт-Петербург), что подтверждается соответствующими актами.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на следующих научных конференциях и семинарах:

1. XIII Международной научно-практической конференции «Управление качеством» (г. Москва, 2014).
2. XII Всероссийской научно-практической конференции «Управление качеством» (г. Москва, 2013).
3. Научной сессии ГУАП (г. Санкт-Петербург, 2013).
4. XI, XIII Международной научно-практической конференции молодых ученых, студентов и аспирантов «Анализ и прогнозирование систем управления» (г. Санкт-Петербург, 2010, 2012).
5. Международной научно-технической конференции «Системы и процессы управления и обработки информации» (г. Санкт-Петербург, 2010).
6. IV Международном научном конгрессе «Нейробиотелеком 2010» (г. Санкт-Петербург, 2010).
7. VI научно-практической конференции «Информационные технологии в профессиональном образовании, научной и библиотечно-информационной деятельности» (г. Санкт-Петербург, 2010).
8. Семинаре молодых ученых с международным участием «Информационные технологии и вычислительные системы» (г. Улан-Удэ, 2011).
9. I, II Международной научной конференции «Актуальные вопросы технических наук» (г. Пермь, 2011, 2013).

1. УПРАВЛЕНИЕ КАЧЕСТВОМ И ЗНАНИЯМИ НА СОВРЕМЕННЫХ ПРЕДПРИЯТИЯХ

1.1. Назначение и актуальность управления знаниями

1.1.1. Понятие и классификация знаний

Практически любое предприятие, находящееся на стадии зрелости своего ЖЦ, характеризуется большими массивами накопленной и разрозненной информации по важным решениям и их обоснованиям, технологиям и методикам работы, ключевым документам по отношениям с заказчиками, а также множеством других разнообразных материалов. Такое состояние предприятия является основной причиной к возникновению следующих проблем [17, 24, 40, 51]:

- работники компании тратят много времени на поиск необходимой информации;
- опыт ведущих и наиболее квалифицированных сотрудников используется только ими самими;
- ценная информация хранится в огромном массиве документов и данных, доступ к которым затруднен;
- дорогостоящие ошибки повторяются из-за недостаточной информированности и игнорирования предыдущего опыта.

Для решения указанных выше проблем используется методология научного направления «управление знаниями». Данное направление охватывает разработку, мониторинг, поддержку и совершенствование стратегий, процессов, организационных структур и технологий для эффективной обработки знаний в рамках предприятия.

Анализ литературы показывает, что общепризнанного и универсального определения понятия «знания» на сегодняшний день не существует. Можно выделить следующие основные определения понятия «знания»:

- «Знания – это сложная сеть понятий и многообразных отношений (оценок, мнений, причинно-следственных и пространственно-временных связей и зависимостей) между ними, которая сознательно (логически) или бессознательно используется нейронной сетью головного мозга при необходимости выработки новых суждений или принятии разнообразных решений. Они отражают окружающий мир, позволяют делать выводы о его поведении. Знания передаются людьми друг другу в виде информации, которая представляет собой выделенные некоторым образом частью сложной сети» [111].
- «Знания – это обогащенная информация, находящаяся в интеллектуальной системе, где ценность добавляется через опыт, контекст, интерпретацию и личное отношение» [9].
- «Знания – это формализованная информация, на которую ссылаются или используют в процессе решения задачи» [91].
- Знание – это обобщенная и формализованная информация о свойствах и законах предметной области, с помощью которой реализуются процессы решения задач, преобразования данных и самих знаний, и которая используется в процессе логического вывода [55].
- «Знания – это описание объектов, процессов и явлений предметной области, содержащее информацию о составе, структуре, закономерностях, связях и свойствах элементов предметной области, которое позволяет решать функциональные задачи предметной области» [12].
- В рамках менеджмента знаний [65] под знанием понимается:
 - информация, подвергшаяся преобразованию в части выделения существенных зависимостей. Знание само задает контекст описания и является целостным описанием ситуации;

- результаты обобщения информации и установления определенных закономерностей в какой-либо предметной области, которые позволяют ставить и решать задачи в этой области;
 - ресурс, базирующийся на практическом опыте специалистов и на информации, существующей на предприятии.
- В работе [7] под знанием понимается некоторая надстройка над информационными потоками, определяемая наличием устойчивых связей между определенными информационными элементами. Эти связи сами по себе в информационных потоках не содержатся и являются в этом смысле внешним по отношению к информации фактором.

Каждое из приведенных выше определений в разной степени отвечает на вопрос, что такое знания.

Понятие «знание» тесно связано и с такими понятиями, как «информация», «данные». Если данные – это полученные эмпирическим путем и зафиксированные факты, дискретно описывающие ситуацию (проблему, объект) вне контекста, т.е. характеризующие отдельные свойства объектов, процессов или явлений, не предназначенные для длительного использования, то информация представляет собой преобразованные и обработанные данные в части выделения контекста описания. Т.е. информация – это результат обработки данных, адресованный конкретному пользователю и пригодный для принятия управленческих или иных решений.

В свою очередь знания, являющиеся результатом переработки информации, имеют весьма длительный цикл жизни, несут определенную идею и снабжены контекстом, определяющим область ее эффективного применения в данном месте в данное время [91, 116]. Практика показывает, что информация может вполне успешно обрабатываться вне зависимости от того, какой смысл в нее заложен. Именно информация является своего рода исходным продуктом знаний [7, 68].

В качестве рабочего определения понятия «знания» в настоящей работе используется следующее:

Знания – информация, подвергшаяся преобразованию в части выделения понятий некоторой предметной области, свойств и связей между ними.

Как отмечается в работе [50], существуют два понимания термина «знание», которые определяют внешние границы, в рамках которых нужно рассматривать вопрос об управлении знаниями и системах поддержки этого процесса на предприятиях.

Первая точка зрения, акцентированная на прагматических аспектах, представляет знания как информацию, полученную для решения практической задачи, обычно для принятия решения, в том числе выполнения действия, человеком или информационной системой.

При этом по своей структуре и способу хранения знания могут ничем не отличаться от прочих данных – любой фрагмент базы данных или полнотекстового архива документов превращается в знание, как только на него обращается взгляд заинтересованного потребителя. Именно положение этого взгляда – фокуса утилитарного интереса – определяет, какой фрагмент данных в настоящий момент интерпретируется как знание.

Вторая точка зрения, акцентированная на содержательных аспектах и принятая в области искусственного интеллекта, полагает, что знания отличаются от данных, обрабатываемых в информационной системе, прежде всего своей структурой. Именно к совокупности особым образом структурированных данных применимо понятие база знаний, подразумевающее:

- логическую упорядоченность данных на основании определенных критериев, устанавливаемых моделью предметной области;
- представление данных в соответствии с определенной моделью знаний (семантической сетью, фреймами, онтологией, набором продукций и др.);

- возможность получения новых данных из старых на основе определенного формального механизма;
- хранение данных в специальных структурах, обеспечивающих высокую эффективность типовых операций над ними (поиск на графах, анализ иерархий, логический вывод и др.).

Так же, как существует множество определений понятия «знания», имеется и множество способов их классификации. В контексте рассматриваемой темы можно привести следующую классификацию знаний.

По уровню универсальности знания можно разделить на *теоретические* (научные) знания (глобальные, универсальные) и *практические* (локальные, контекстно-зависимые, уровня предприятия). Теоретические знания представляются в форме [111]:

- основных логических форм – понятий, классификации понятий, суждений (высказываний), умозаключений;
- зависимостей, законов, закономерностей, принципов;
- методов, методологий, технологий.

Практические знания и умения человека, его профессиональные знания «о том, как делать» (ноу-хау) не могут быть выражены в виде точных определений, моделей и алгоритмов.

По содержанию знания делятся на *декларативные* и *процедурные* [56, 91, 113]. Декларативные знания – это знания о фактах, в то время как процедурные знания это знания о том, как применить эти факты для решения задач. Декларативные знания состоят из множества взаимосвязанных элементов информации и взаимосвязей между ними. Декларативные знания хорошо описываются сложной сетью понятий и отношений между ними (семантические сети, фреймы, онтологии). Процедурные знания описывают действия, порядок принятия решений и выполнения работ. Процедурные знания основываются на декларативных знаниях и определяют способы

оперирования и преобразования декларативных знаний. Процедурные знания имеют более тесную связь с деятельностью организации, но их формальное описание является затруднительным.

По форме представления знания делятся на *неявные (скрытые, неформализованные)* и *явные (формализованные)* [91, 116]. Скрытые знания трудно выразить. Они часто заключены в интуиции и в не поддающихся анализу опыте, навыках и привычках. Неявными знаниями может обладать отдельный человек или группа людей. Явные знания закодированы на естественном (русском, английском и т.п.) или искусственном (математика и т.п.) языках. Явные знания в большей степени систематизированы, закодированы и, следовательно, более легко распространяются.

Неявные и явные знания являются существенными компонентами при разработке стратегии управления знаниями.

Знания предприятия можно разделить на *индивидуальные (личностные)* и *корпоративные (организационные, коллективные)* [91, 116]. К объектам корпоративных знаний относятся носители (индивидуумы и группы) и артефакты (документы, книги, компьютерные системы и т.п.). Знания, которые содержат носители, являются субъективными, в то время как знание, содержащееся в артефактах, является объективным, представленным в форме явных знаний. Корпоративные знания делятся на *внешние* и *внутренние*. К внешним знаниям относятся знания о сторонних технологиях, методиках работы, клиентах, независимая аналитическая информация и т.д. К внутренним знаниям относятся знания о внутренних процессах, продуктах и услугах предприятия, знания сотрудников (наиболее ценный актив предприятия) и т.д.

Корпоративные знания являются важнейшим ресурсом современного предприятия, способным значительно повлиять на повышение его конкурентоспособности и инвестиционной привлекательности.

1.1.2. Процесс управления знаниями

Начиная с середины 90-х годов прошлого века, знания начали рассматриваться в компаниях в качестве важнейшего ресурса, ключевого фактора успеха и нового источника дохода [17, 21, 57, 137-140]. Хотя компании управляли своими людскими и интеллектуальными активами задолго до этого, научная дисциплина «Управление знаниями» была сформирована именно в это время для решения новых специфических задач бизнеса. Сегодня управление знаниями является актуальной темой, обсуждаемой специалистами всех уровней.

Способность эффективно использовать и развивать знания, воплощать их в новые продукты и услуги превращается в важнейший фактор выживания в условиях информационного общества. Поэтому проблема обмена знаниями, стимулирования сотрудников к участию в процессе накопления и использования коллективных знаний и внедрению СУЗ актуальна во всех отраслях [15, 134].

В настоящее время особенно важна скорость распространения и применения новых знаний – это повышает мобильность компании, её способность к переменам. Компания, сотрудники которой обладают свободным доступом к коллективным знаниям, принимают более качественные решения, быстро и эффективно реагируют на все изменения окружающей среды, получает неограниченные возможности роста и развития.

Управление знаниями (УЗ) является одной из основных концепций управления, наряду с управлением качеством, совершенствованием и реинжинирингом бизнес-процессов и т.д. [26, 117]. При этом целенаправленное использование УЗ высвобождает огромные потенциалы экономии и роста, которые не могут быть реализованы с помощью традиционных концепций реорганизации и модернизации.

В работах российских и зарубежных авторов прослеживается идея о необходимости систематической, целенаправленной и научно обоснованной

деятельности по работе со знаниями и неразрывной связи этих действий с менеджментом качества [8, 81, 116, 122].

Не существует общепринятого определения понятия «управление знаниями», что в основном связано с различными точками зрения на понимание термина «знание».

Можно отметить следующие основные определения понятия «управление знаниями»:

- «Управление знаниями – это дисциплина, которая обеспечивает интегрированный подход к созданию, сбору, организации и использованию информационных ресурсов предприятия и доступу к ним. Эти ресурсы включают базы данных, текстовую информацию, такую как документы, описывающие правила и процедуры, и, что наиболее важно, неявные знания сотрудников» [173].
- «Управление знаниями — это совокупность процессов управления созданием, распространением, обработкой и использованием информации внутри предприятия» [25].
- «Управление знаниями – это формальный процесс, который состоит в оценке организационных процедур, людей и технологий и в создании системы, использующей взаимосвязи между этими компонентами с целью предоставления нужной информации нужным людям в нужное время, что приводит к повышению продуктивности» [172].
- В работе [114] под УЗ понимается комплексный набор мероприятий, направленных на поддержание в организации системного порядка работы с информационно-знаниевыми ресурсами и специалистами для поиска, накопления и облегчения доступа к знаниям, повторного или многократного их использования.

Таким образом, УЗ рассматривается как систематический процесс накопления, поиска, использования и передачи знаний в интересах получения

конкурентных преимуществ. При этом целью УЗ является накопление интеллектуального капитала, выявление и распространение информации и опыта, создание условий для распространения и передачи знаний. На практике это – систематическое и целенаправленное формирование, обновление и применение знаний для усиления эффективности компании.

Существует ряд моделей процесса управления знаниями. Анализ концепций по УЗ [111, 116, 137] показывает, что в большинстве случаев наиболее важными являются следующие пять основных видов деятельности, связанных со знаниями:

1. Выявление – деятельность по преобразованию индивидуальных знаний в знания предприятия (его сотрудников).
2. Создание – деятельность, результатом которой являются новые знания или новые конфигурации существующих знаний.
3. Организация знаний – деятельность по классификации и категоризации знаний для навигации, запоминания, поиска и сопровождения знаний.
4. Доступ – деятельность по санкционированной передаче и распространению знаний между сотрудниками. Защита знаний от несанкционированного доступа.
5. Использование – деятельность по применению знаний для принятия решений.

Эти виды деятельности со знаниями составляют единый спиралевидный процесс УЗ (рисунок 1.1). Все они встречаются в любой организации, обычно выполняются для поддержки бизнес-процессов и в разной степени систематически, организационно и технологически поддерживаются.

Особые трудности представляет собой задача извлечения знаний, которая является предметом инженерии знаний (knowledge engineering) – совокупности процессов и методов, направленных на извлечение, структурирование и формализацию знаний. Результатом процесса инженерии знаний являются

явные знания [13, 18-20]. Инженерия знаний – ядро концепции управления знаниями.



Рисунок 1.1 – Основные виды деятельности со знаниями

Основными этапами процесса инженерии знаний являются:

- Извлечение знаний (knowledge capture) – процесс получения знаний из источников (ресурсов знаний), которыми могут быть артефакты (файлы, документы, книги) и эксперты (группы экспертов).
- Структурирование знаний (knowledge structuring) – процесс организации и установления взаимосвязей в знаниях, полученном в процессе извлечения.
- Формализация знаний (knowledge formalizing) – процесс представления структурированных знаний на определенном языке представления знаний.

Выделяют три основных компонента управления знаниями [116] (рисунок 1.2):

1. Сотрудники получают, генерируют и передают знания.
2. Бизнес-процессы. Знания потребляются и создаются при выполнении бизнес-процессов.
3. Информационные технологии обеспечивают быструю и эффективную работу сотрудников и бизнес-процессов.



Рисунок 1.2 – Компонентная структура управления знаниями

Сотрудники предприятия решают задачи, используя мозговой штурм, нововведения, творческие силы и знания, полученные из опыта. Совместная работа сотрудников умножает знания, накапливаемые предприятием, и улучшает условия достижения потенциального успеха.

Приобретение знаний, их хранение, распределение и преобразование в форму, удобную для внутрикорпоративного использования, предполагает формирование определенных организационных и технологических условий. Решение задачи управления знаниями возможно лишь в тесном контакте между специалистами самых различных сфер деятельности предприятия и

специалистами по информационным технологиям, а главное, при заинтересованности менеджеров предприятия в получении оперативного доступа к любым информационным ресурсам предприятия в удобной форме и анализу полученной информации.

Быстрый доступ к необходимым знаниям играет первостепенную роль, поскольку позволяет значительно повысить качество бизнес-процессов. Знания, приобретаемые с фактическим опытом, продуктивны только тогда, когда они используются при выполнении работы или интегрируются в процесс выполнения работы [116]. Знания в организации не используются и не создаются в результате собственных пожеланий сотрудников. Для этого у них нет ни времени, ни желания. Знания потребляются и создаются при выполнении бизнес-процессов организации и являются условиями эффективного их выполнения [107, 111].

Связь бизнес-процессов и процессов работы со знаниями имеет особое значение для предприятия – разработчика ПО. Основные технологические процессы непосредственно направлены на разработку ПС, которое является интеллектуальным активом такого предприятия. Знания являются входными и выходными элементами этих процессов, где исполнители процессов (сотрудники предприятия) не только выполняют работу, но используют и создают знания. Поэтому процесс УЗ должен быть внедрен в ключевые процессы такого предприятия.

Согласно отрасленезависимой процессной модели APQC Process Classification Framework версии 6.1.1, подразделяющей все существующие бизнес-процессы предприятия на 12 категорий, процесс УЗ (пункт 12.5, номер 11073) относится к группе процессов разработки и управления бизнес-возможностями [167]. Таким образом, процесс УЗ на различных уровнях зрелости имеется на всех предприятиях, в том числе и на предприятии – разработчике ПО.

В целом, в мировой практике УЗ используется для достижения нескольких целей, в частности, для улучшения взаимодействия между сотрудниками и подразделениями предприятия, сохранения знаний сотрудников, улучшения производственных показателей, расширения бизнеса и инноваций, улучшения качества продукции, более эффективного принятия решений, а чаще всего для решения одновременно нескольких задач. Если цели УЗ будут соответствовать стратегическим задачам предприятия, тогда эффект будет существенным.

1.2. Ресурсы знаний предприятия

Знания предприятия извлекаются из ресурсов знаний. Ресурсы знаний различаются в зависимости от отраслей индустрии и приложений, но, как правило, включают различные руководства, письма, информацию о заказчиках, сведения о конкурентах, технологии, накопившиеся в процессе разработки продукции, и т.д. [17, 113, 130]. Сами эти ресурсы рассредоточены по всему предприятию и могут находиться в различных местах, например, в электронных документах, медиа-файлах, базах данных, а также у специалистов. Связанная совокупность ресурсов знаний образует единое пространство знаний предприятия [110, 130, 142]. Пространство знаний имеет определенную структуру и делится на области знаний (предметные области), которые используют единую концептуализацию (систему понятий).

Как уже было отмечено в разделе 1.1, все знания делятся на декларативные и процедурные. Декларативные знания – это знания о фактах, в то время как процедурные знания это знания о том, как применить эти факты для решения задач. Процедурные знания ближе к реальной деятельности предприятия, но их сложнее описывать и выполнять их формальную обработку. Декларативные знания проще для описания. Они менее изменчивы во времени. Они создают каркас (структуру) пространства знаний. Декларативные знания являются базовыми, так как процедурные знания строятся на их основе. В связи

с этим для построения модели знаний предприятия в основном используются декларативные знания [80, 90, 111, 113].

Основными элементами декларативных знаний являются понятия. Описание декларативных знаний основывается на описании понятий, отношений между понятиями и их экземплярами, которые имеются в среде предприятия, т.е. множество понятий образуют основу теоретического и практического знания. При описании знаний организации выбор понятий и связей между ними выполняется в процессе концептуализации знаний предметной области.

Знания и описывающие их понятия содержатся в некоторых материальных ресурсах (носителях, контейнерах, объектах, источниках знаний). Знания о самих ресурсах также хранятся в виде некоторых понятий (т.е. ресурсы описываются понятиями и в то же время ресурсы содержат в себе наборы понятий). Все ресурсы знаний могут быть разделены на две составляющие: активы знаний и объекты знаний [111].

Под активами знаний понимаются материальные и нематериальные сущности, которые создают, изменяют и развивают знания. Активами знаний любой организации являются: сотрудники, группы или сообщества сотрудников; часть статической и динамической структуры организации (явно или неявно регулирующие поведение организации); часть неявной или явной культуры организации.

Под объектом знаний понимается явное представление информации, которая должна быть обработана (людьми или программными средствами), чтобы преобразоваться в знания. Знания получаются (извлекаются) из информации, содержащейся в объектах знаний.

На рисунке 1.3 [21] представлено разнообразие ресурсов знаний предприятия.

Таким образом, под ресурсами знаний предприятия понимаются скрытые знания сотрудников (групп и подразделений) и разнообразные объекты знаний, содержащие явные знания, что может быть представлено в следующем виде:

$$Pз = \langle Zc, Oz \rangle, \quad (1)$$

где $Pз$ – ресурсы знаний предприятия; $Zc = \langle S, P \rangle$ – множество объектов, содержащих скрытые знания; $Oz = \langle D, M, DB \rangle$ – множество объектов, содержащих явные знания; S – множество носителей индивидуальных знаний; P – множество носителей коллективных знаний; D – множество текстовых документов; M – множество медиа-файлов; DB – множество баз данных.

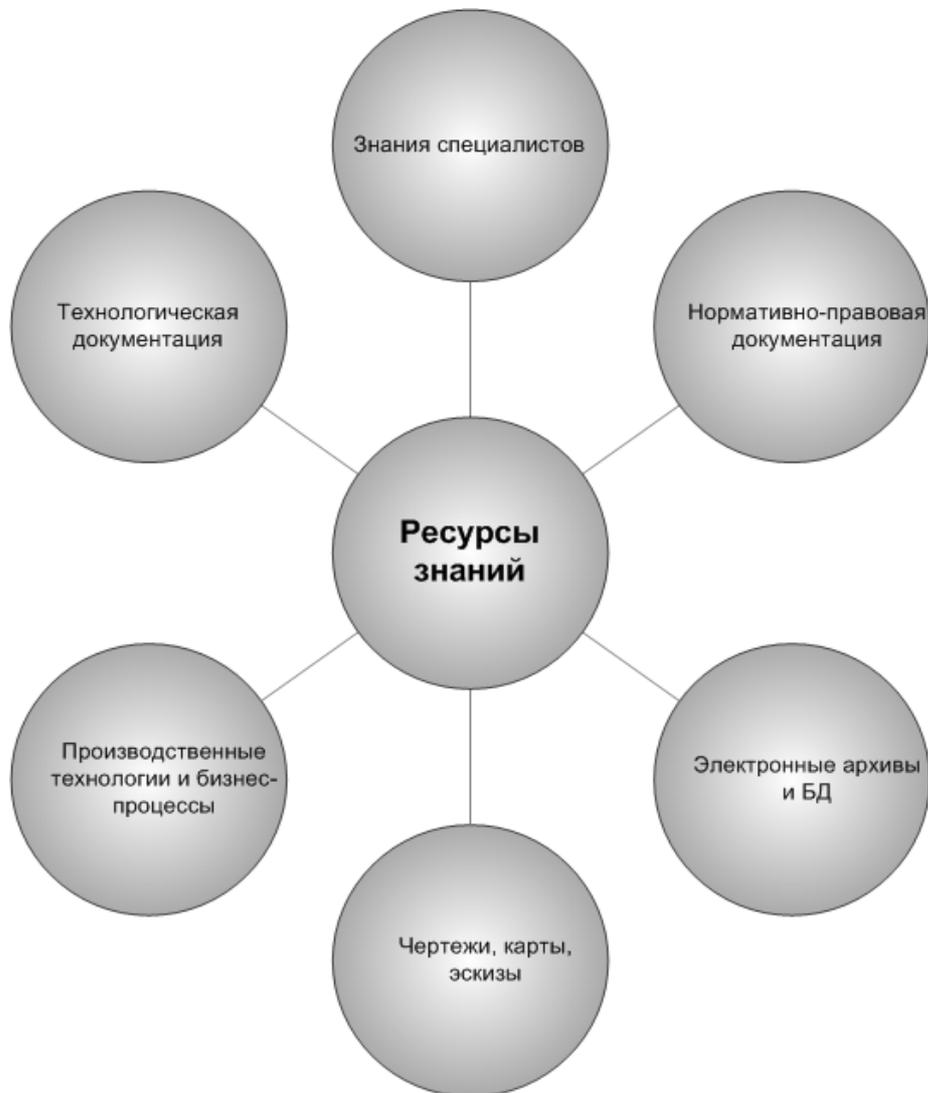


Рисунок 1.3 – Ресурсы знаний предприятия

Множество *Oz* может включать и другие объекты знаний для различных предприятий.

Исследуемое предприятие ЗАО «Петер-Сервис» является разработчиком решений в области систем класса OSS/BSS для поддержки операционной и бизнес деятельности крупных операторов связи. Продукты компании адаптированы для работы в мультисервисной сети и снабжены всем необходимым для dual-, triple- и quad-play операторов и предоставления услуг NGN, выполняют поддержку и тарификацию VAS для 3G/UMTS сетей и услуг широкополосного доступа, с возможностью взаимодействия с оборудованием различных типов, в том числе с оборудованием IMS-платформ. Разрабатываемое на предприятии ПС относится к категории сложных программных средств.

На исследуемом предприятии ЗАО «Петер-Сервис» можно выделить четыре типа ресурсов знаний (рисунок 1.4):

1. Текстовые документы.

В компании постоянно разрабатываются (выполняются) проекты, результаты которых сохраняются в виде электронных документов того или иного формата. Это могут быть документация на ПС, отчеты о внедрениях, учебные материалы.

2. Презентации и видеозаписи презентаций.

Презентации, посвященные новым возможностям программных продуктов или автоматизированным решениям для внутренних целей. Данные презентации записываются средствами Microsoft Live Meeting и далее размещаются на узлах подразделений компании. Сотрудники организации так же участвуют в презентациях, проводимых другими компаниями.

3. Доски обсуждений.

Сотрудники предприятия обсуждают технические или административные вопросы, используя для этого доски обсуждения, реализованные

средствами Microsoft SharePoint. Доски обсуждений могут быть отражениями переписки, ведущейся в рамках каких-то профильных списков рассылки. При помощи данного средства сотрудники переводят свои неявные знания в явные.

4. Другие объекты.

К данной группе относятся остальные формализованные ресурсы знаний, хранящиеся на локальных компьютерах, узлах и т.п.

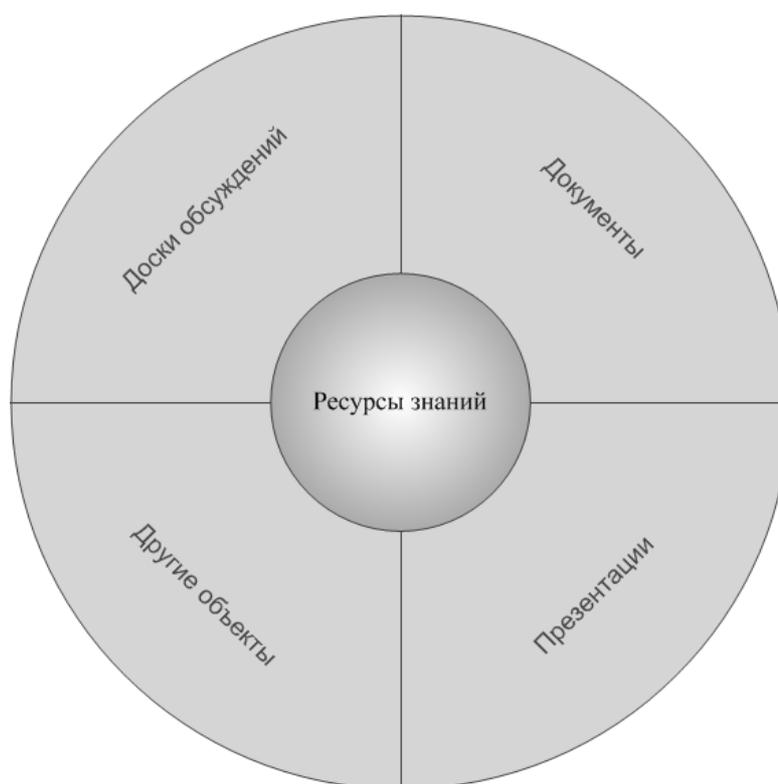


Рисунок 1.4 – Ресурсы знаний исследуемого предприятия

Все приведенные выше ресурсы знаний распределены в информационном пространстве организации и практически никак между собой не пересекаются и не взаимодействуют. Подавляющее большинство указанных ресурсов носит локальный характер, хотя информация и знания из этих ресурсов могут быть получены и использованы всеми сотрудниками предприятия, связанным с основной производственной деятельностью и другими обеспечивающими видами деятельности (маркетинг, продажи и т.д.).

Правила работы с большинством объектов из указанных ресурсов не регламентированы, развитие происходит хаотично и бессистемно. Администрирование практически отсутствует или осуществляется по остаточному принципу.

Основным ресурсом знаний является документация на ПС, которая содержит около 90% всей информации о разрабатываемых в компании продуктах. Эти знания используются во всех основных технологических процессах предприятия: при разработке проектной документации, разработке программных продуктов и документации, тестировании, сопровождении, внедрении и т.д.

Сегодня в компаниях – разработчиках ПО не уделяется достаточного внимания документации на ПС, когда все усилия сосредотачиваются только на разработке продукции. Это порождает следующие проблемы:

- знания востребованы только разработчиками;
- в случае увольнения (отпуска, командировки и т.д.) ключевых сотрудников их нечем заменить;
- непосредственное изучение исходных кодов программ и данных значительно увеличивает трудозатраты на дальнейшую разработку, сопровождение и эксплуатацию ПС;
- заказчики не понимают, как им использовать ПС.

Таким образом, знания, которые содержатся в документации на ПС, имеют максимальный приоритет с точки зрения управления ими и создания системы поддержки этого процесса на предприятии – разработчике ПО. Эти знания определяют специфику рассматриваемой предметной области. Другие объекты знаний являются общими. К ним могут применяться существующие подходы к управлению знаниями, не являющиеся узкоспециализированными и широко представленными в различных публикациях по данной тематике [16, 27, 57, 89, 111, 113, 115, 116, 118].

1.3. Управление качеством процессов разработки программного обеспечения

1.3.1. Стандарты в области разработки программного обеспечения

В решение проблем стандартизации и управления качеством программной продукции большой вклад внесли такие ученые как В.В. Липаев [71-74], А.Н. Терехов [108], Р. Фатрелл [120], И. Соммервилл [106], К. Полк [164-166] и др. Тем не менее, за рамками разработанных подходов остался вопрос использования методов и средств управления знаниями в управлении качеством разработки ПС.

Среди стандартов в области разработки СМК, оценки качества процессов и уровней зрелости компаний, разрабатывающих ПС, а также совершенствования базовых процессов, методологий разработки в настоящее время наиболее популярными являются: ISO 9000, ISO/IEC 15288, ISO/IEC 12207, ISO/IEC 15504, TickIT, CMMI, SWEBOOK, JIS/TR Q 0005, RUP, гибких методологий разработки ПС (SCRUM, XP) [10, 36, 37, 78, 88, 164-166, 174, 178]. Указанные стандарты (как специфические для области разработки ПС, так и общие стандарты по управлению качеством) документально закрепляют использующиеся в организациях производственные знания и предлагают некоторые действия по управлению ими, однако сам процесс управления знаниями в явном виде не регламентируют.

Скоординированная деятельность по общему управлению организации осуществляется путем создания и внедрения систем менеджмента. Составными частями общей системы менеджмента типовой организации являются:

- СМК;
- системы административного и организационного управления;
- система менеджмента финансовой деятельности.

Управление организацией, как единой системой взаимосвязанных и взаимодействующих процессов систем менеджмента, позволяет обоснованно

выделять и перераспределять внутренние ресурсы, устанавливать и поддерживать надежные отношения с потребителями и поставщиками, а также принимать решения, основанные только на достоверных и объективных фактах [29, 69].

Под СМК понимается совокупность взаимосвязанных и взаимодействующих процессов и необходимых ресурсов организации для достижения целей в области качества [34, 73].

1.3.2. Процессы системы менеджмента качества

СМК исследуемого предприятия ЗАО «Петер-Сервис» сертифицирована по международному стандарту ISO 9001. Основными показателями качества производимого ПО являются его соответствие востребованной рынком функциональности, высокая надежность, тиражируемость и конфигурируемость. Высокие показатели качества достигаются в результате устойчивых процессов управления, механизмом разработки, внедрения и совершенствования которых является СМК.

Основу СМК исследуемого предприятия составляет иерархия процессов, охватывающих все жизненно важные виды деятельности предприятия и направленных на обеспечение удовлетворенности потребителей продукции (рисунок 1.5). Под продукцией понимаются прикладные ПС для управления техническими средствами и технологическими процессами, продукция других компаний, которая включается в состав продукции организации, на основании проектных решений, и работы (услуги), выполняемые в ходе проектирования, установки, обучения и дальнейшего сопровождения поставляемой продукции.

Первый уровень иерархии составляют общие процессы СМК – это процессы высшего уровня иерархии – организационные.

Второй уровень состоит из базовых (ключевых) процессов СМК – это процессы, реализующие модель ЖЦ продукции (процессы, вошедшие в ЖЦ производимой и поставляемой продукции и оказывающие влияние на её

качество), а также процессы постоянного улучшения продукции, процессов и СМК в целом.

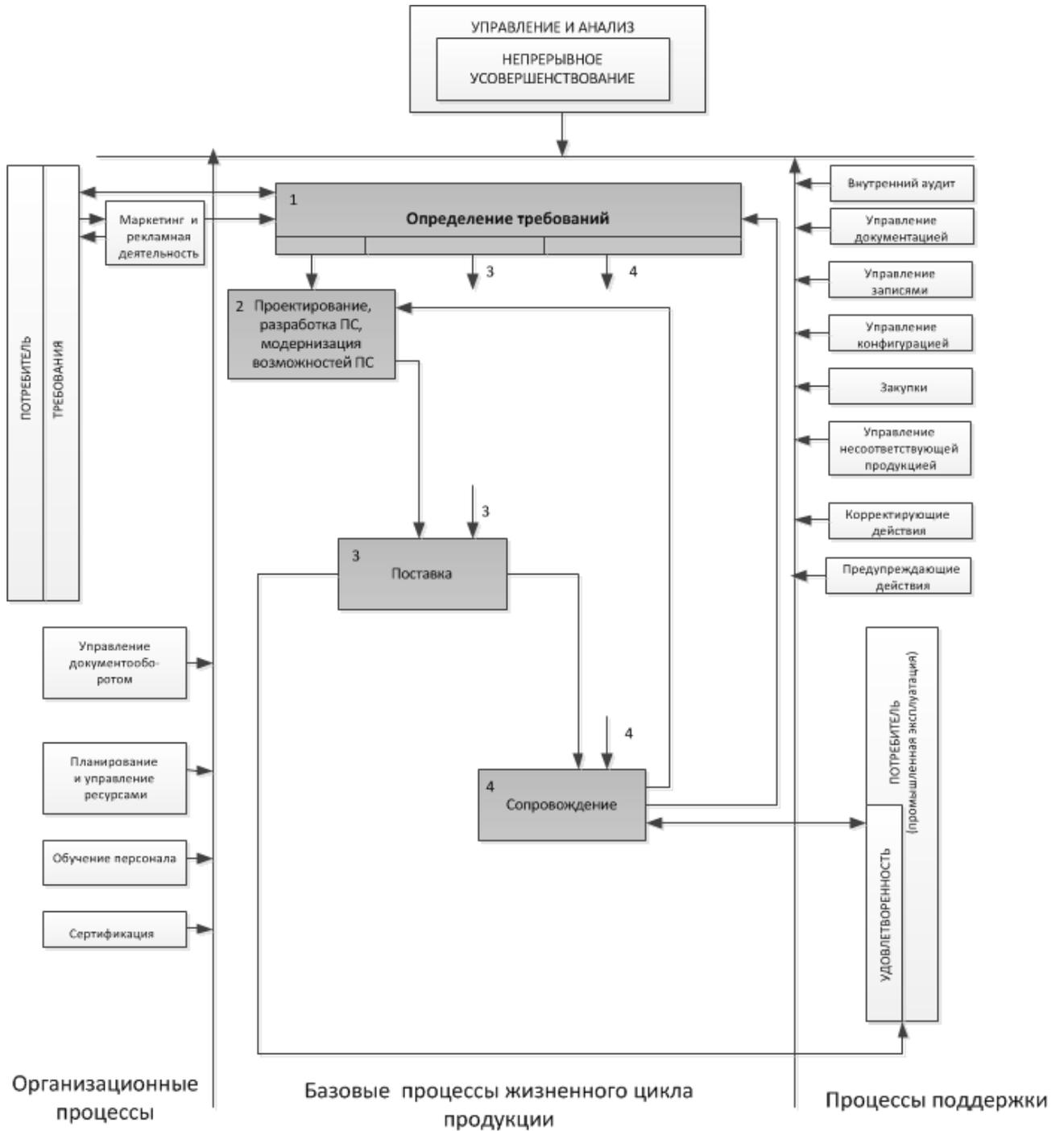


Рисунок 1.5 – Схема процессов СМК предприятия – разработчика ПО

Третий уровень содержит процессы поддержки СМК обеспечивающие функционирование базовых процессов и решение конкретных задач функционирования СМК.

Процессный подход, реализованный в СМК предприятия, направлен на удовлетворение интересов как внешних, так и внутренних потребителей, что позволяет организации постоянно улучшать свою деятельность по разработке и производству продукции.

Функционирование процессов СМК базируется на принципах TQM и реализуется в циклах PDCA.

Технология поддержки ЖЦ ПС, применяемая при работе с потребителями, базируется на основе ГОСТ Р ИСО/МЭК 12207 и ГОСТ 34.601 [34, 37].

Под ЖЦ продукции понимается совокупность процессов, связанных с последовательным изменением состояния продукции от формирования исходных требований к нему до окончания его эксплуатации. ЖЦ состоит из стадий – логически завершенных частей. Стадии характеризуются определенным состоянием продукции, видом предусмотренных работ и их результатом.

Технология проектирования прикладных ПС, предназначенных для автоматизации бизнес-процессов операторов связи включает:

- разработку архитектуры ПС и архитектуры комплекса технических средств, на базе которого устанавливаются прикладные ПС;
- разработку требований к эксплуатационно-техническим характеристикам комплекса технических средств;
- разработку прикладных ПС;
- разработку требований к системным ПС.

Важное место среди процессов предприятия – разработчика ПО занимают базовые процессы ЖЦ ПС, связанные с проектированием, разработкой и сопровождением программной продукции, так как они являются наиболее трудоёмкими, и качественное их выполнение во многом определяет

способность организации выполнять требования, предъявляемые потребителем к поставляемой продукции.

1.3.3. Базовые процессы жизненного цикла программных средств

Прикладные ПС разрабатываются или модернизируются в соответствии с планом инициативных работ или по заказам потребителей.

Технологическая цепочка создания прикладных ПС начинается с выбора модели ЖЦ продукции, которая соответствует области реализации, объему и сложности разрабатываемой продукции. При этом выбираются и структурируются в модель ЖЦ разрабатываемой продукции подпроцессы и задачи разработки. Критериями выбора модели ЖЦ служат следующие данные:

- о наличии четких и подробных требований к продукции;
- о наличии ресурсов, необходимых для выполнения работ;
- о наличии и доступности заказчика в процессе разработки;
- о новизне используемых в процессе разработки технологий и (или) инструментальных средств.

Большинство программных продуктов разрабатывается по каскадной модели ЖЦ ПС, представленной на рисунке 1.6.

Базовые процессы ЖЦ ПС включают в себя четыре процесса (рисунок 1.7):

- определение требований;
- проектирование и разработка;
- поставка;
- сопровождение.

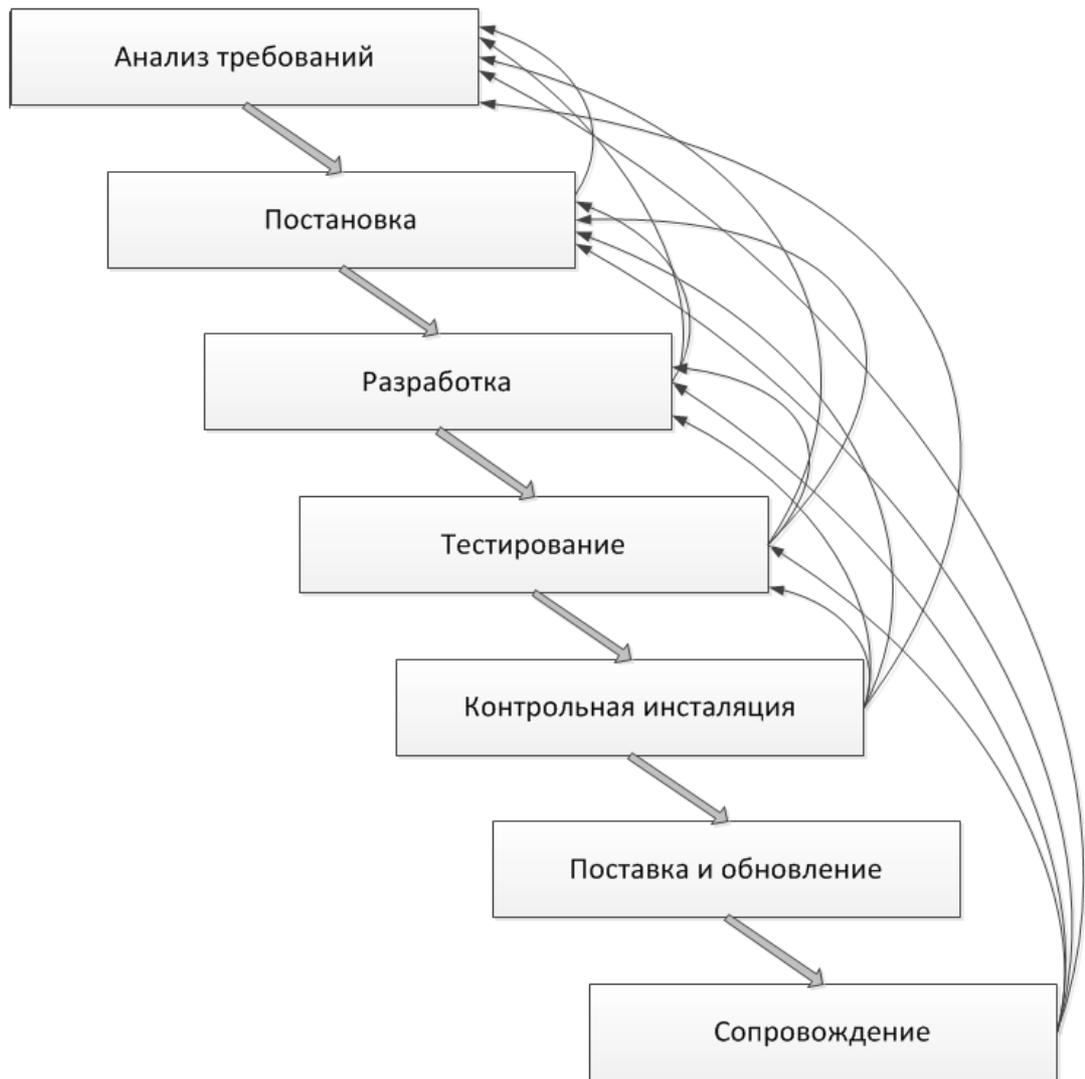


Рисунок 1.6 – Модель ЖЦ ПС

Базовые процессы могут разделяться на набор подпроцессов (работ), которые будут выполняться в ЖЦ разрабатываемой и поставляемой продукции.

Необходимость реализации конкретных работ, задач и условия выполнения процесса определяются владельцами процессов.

Детализация процессов производится в стандартах, методологических инструкциях и руководящих документах по конкретным направлениям деятельности предприятия при формировании технологии и комплекса инструментальных средств поддержки разработки и сопровождения конкретной продукции.

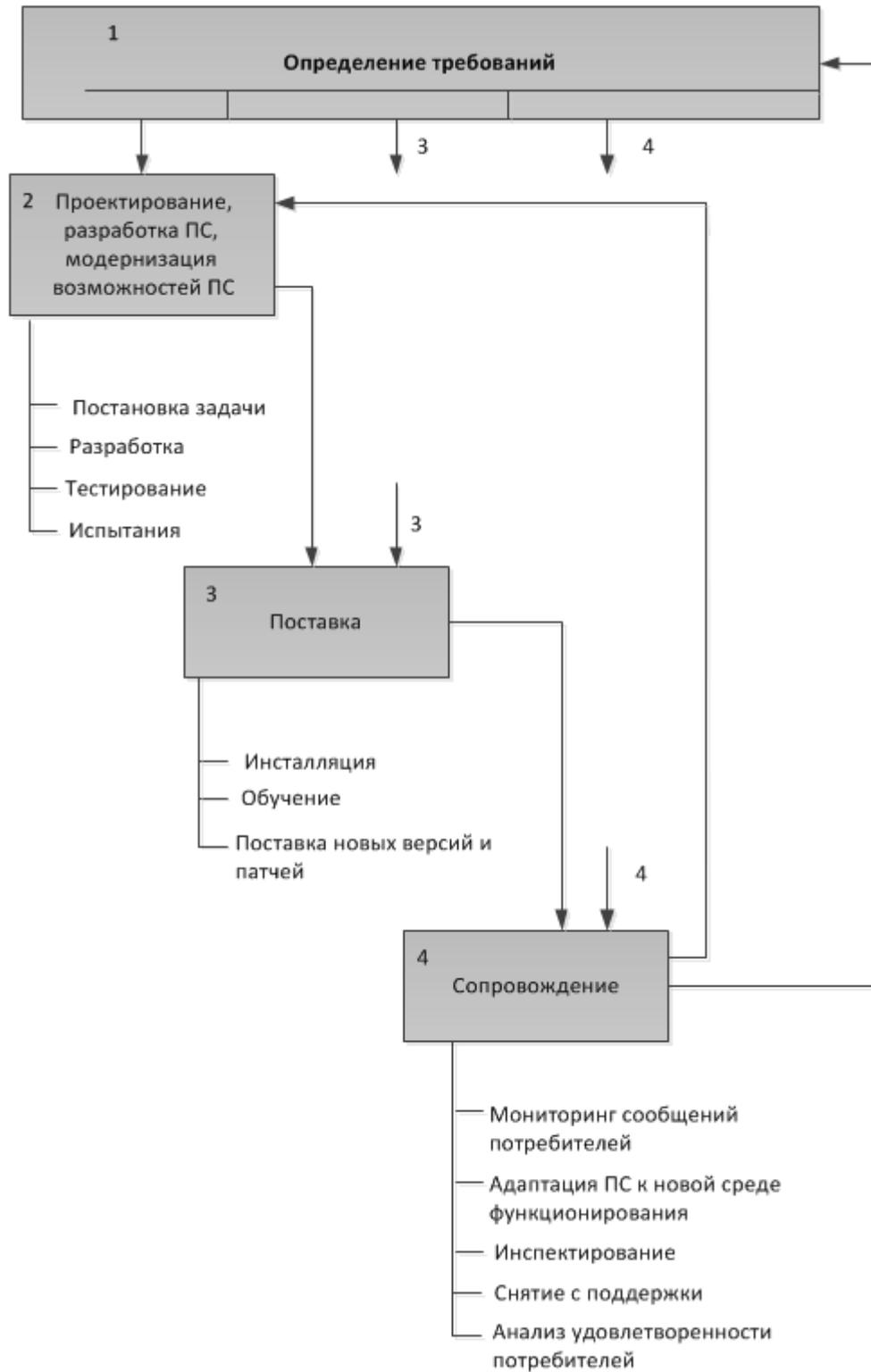


Рисунок 1.7 – Базовые процессы ЖЦ ПС

В ходе реализации базовых процессов осуществляется управление несоответствующей продукцией. Продукция предприятия признается

соответствующей установленным требованиям только после прохождения всех испытаний, предусмотренных технологическим циклом создания. Не прошедшая испытаний продукция может передаваться потребителям лишь с целью ознакомления со способом реализации, без права ее использования в промышленной эксплуатации. Ответственность и полномочия сотрудников и руководителей подразделений при выявлении несоответствующей продукции определены их функциональными обязанностями.

Процесс определения требований

Процесс определения требований – это формальное задание действий, которые по замыслу заказчика должны выполнять прикладные ПС. Этот процесс является основной точкой связи между потребителем и организацией. В ходе процесса выполняется эскизное проектирование и формируется документ «Функциональная спецификация», который описывает формальные требования к прикладным ПС.

Исполнителями процесса являются выделенные группы аналитиков, постановщиков или руководители проектов.

На входе процесса: формализованные или неформализованные требования потребителя к продукции.

На выходе процесса: согласованный и утвержденный документ «Функциональная спецификация».

Документ «Функциональная спецификация» должен содержать формализованное описание автоматизируемых бизнес-процессов; описание функциональных и нефункциональных требований, ограничений, описание форматов выходных документов; подробное описание выбранного способа реализации, концептуальной модели данных, при необходимости эскизы экранных форм. Так же этот документ должен содержать ссылки на документы, на основании которых в ходе выбора способа реализации принимались решения (документированные требования потребителя, маркетинговая информация, протоколы совещаний и т.д.).

В качестве документа «Функциональная спецификация» может быть использован документ, предоставленный потребителем. Если документ достаточно полон и удовлетворяет требованиям к содержанию функциональной спецификации, то эскизное проектирование не производится.

Процесс проектирования и разработки

Процесс проектирования и разработки состоит из подпроцессов:

- реализации определенных в предыдущем процессе требований (постановка задач, разработка компонентов программного обеспечения, разработка документации);
- тестирования (верификации) – испытания, направленные на предотвращение выпуска продукции, не соответствующей требованиям;
- окончательного контроля продукции (контрольной инсталляции) – проверка разработанной продукции на готовность использования в заданных условиях эксплуатации. По результатам испытаний продукция признается пригодной к поставке или требующей доработки.

Исполнителями процесса являются выделенные группы постановщиков, разработчиков, документаторов, инженеров по тестированию или выделенные в составе групп по направлениям деятельности сотрудники.

Постановка задач – это процесс формализации требований к функциональности и производительности продукции, подготовки документа «Техническая спецификация», описывающего состав и алгоритмы функционирования с подробностью достаточной для его реализации.

На входе процесса: документ «Функциональная спецификация».

На выходе процесса: документ «Техническая спецификация».

Разработка компонентов программного обеспечения – это процесс создания/изменения набора файлов, содержащих исходные коды, которые необходимы для компиляции и (или) создания дистрибутива (дистрибутивов) продукции.

На входе процесса: проектная документация.

На выходе процесса: компоненты программного обеспечения.

Разработка документации – это процесс создания/изменения документации на поставляемую продукцию.

На входе процесса:

- проектная документация;
- компоненты программного обеспечения.

На выходе процесса: эксплуатационная документация.

Тестирование ПС – процесс исследования ПС с целью получения информации о качестве программного продукта.

На входе процесса:

- проектная документация;
- компоненты программного обеспечения;
- эксплуатационная документация.

На выходе процесса: собранные компоненты ПС (дистрибутив) и эксплуатационная документация.

Контрольная инсталляция – это процесс испытания поставляемой продукции, проводимый для демонстрации того, что поставляемая продукция удовлетворяет установленным требованиям (функционирует в полном соответствии с требованиями и условиями, реализованными в предшествующих работах) и готова к использованию в заданных условиях эксплуатации.

Процесс поставки

Процесс поставки объединяет работы и задачи, связанные с передачей продукции потребителю и состоит из подпроцессов инсталляции продукции, обучения заказчика и поставки новых версий и патчей.

В ходе процесса поставки осуществляется установка и настройка продукции на территории потребителя, включая приемо-сдаточные испытания, обучение пользователей (в соответствии с условиями договора на поставку), оформление и выдача потребителю лицензии на использование. Исполнителями процесса являются сотрудники группы установки.

Процесс сопровождения

Процесс сопровождения осуществляется в соответствии с условиями соглашения или договора (контракта) и объединяет работы и задачи, связанные с обеспечением обратной связи с потребителями:

- мониторинг и анализ сообщений о проблемах и заявок на внесение изменений (модификацию) продукции. Основанием для проведения работ по анализу проблем и изменений являются: официальное предложение по модификации или отчет о проблеме;
- инспектирование (мониторинг) лицензионных параметров и условий использования лицензии;
- адаптация (перенос) программного продукта (включая данные) из прежней в новую эксплуатационную среду (например, в случае замены комплекса технических средств);
- мониторинг и контроль удовлетворенности потребителей и последующий анализ полученных от потребителей данных;
- снятие программного средства с поддержки.

Исполнителями процесса сопровождения являются выделенные группы сопровождения.

Таким образом, знания являются входными и выходными элементами рассмотренных базовых процессов. Часть знаний и информации становится явными (формализуется в виде определенного набора документов, текстов программ и данных), другая часть остается неформализованной. Взаимосвязь

ресурсов знаний и основных участников процессов на разных этапах ЖЦ ПС показана рисунке 1.8.

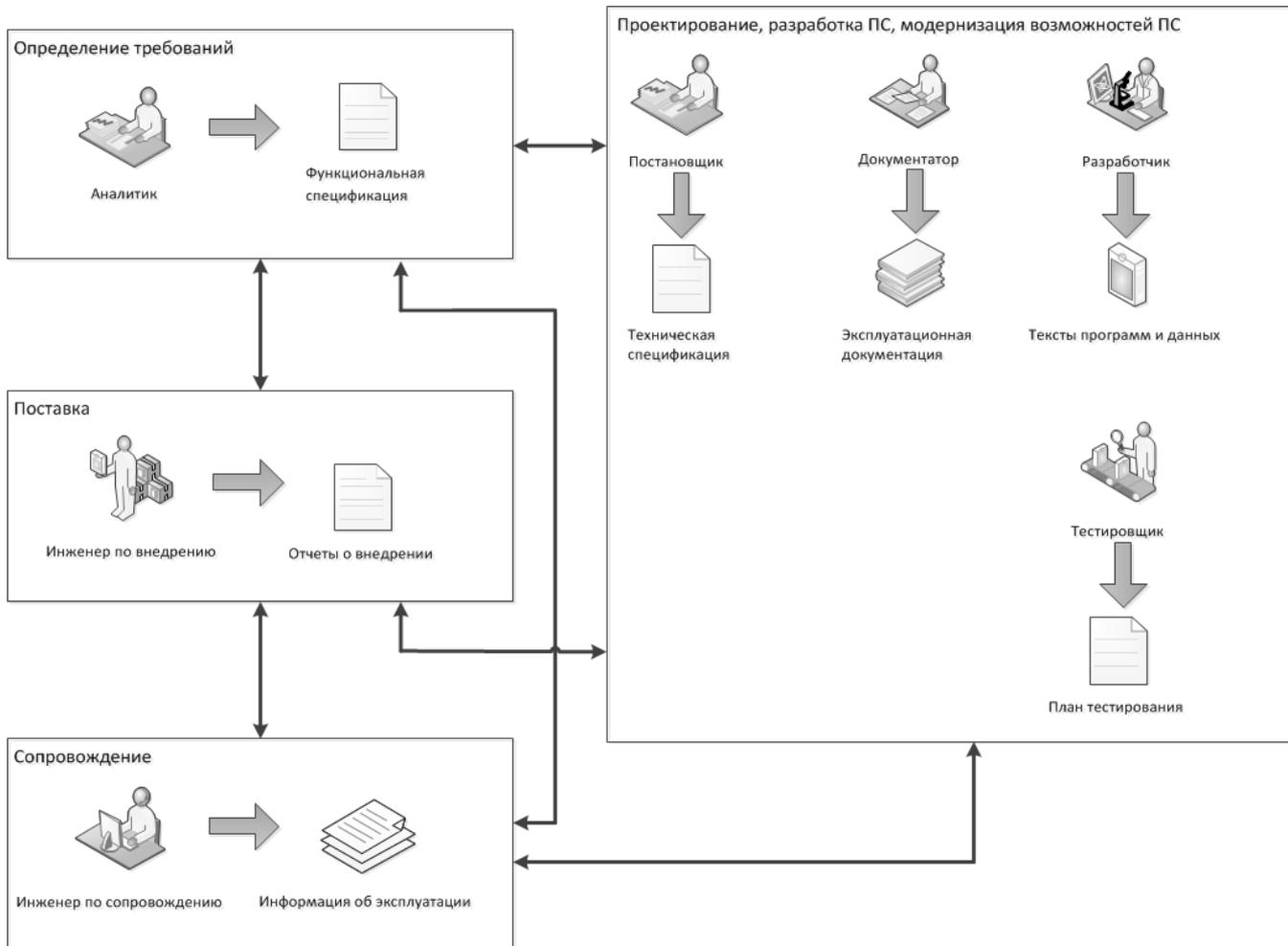


Рисунок 1.8 – Взаимосвязь ресурсов знаний на разных этапах ЖЦ ПС

1.3.4. Критерии оценки базовых процессов жизненного цикла программных средств

В таблице 1 представлены критерии оценки процессов ЖЦ ПС исследуемого предприятия.

Таблица 1 – Критерии оценки процессов

Процесс (подпроцесс)	Период сбора данных	Измерения в ходе процесса	Измерения после выполнения процесса	Критерий
Определение требований (анализ требований потребителя)	Квартал	Анкетирование потребителей. Вопрос в анкете: "Степень соответствия функциональных возможностей продуктов существующим потребностям".	-	Средняя оценка, поставленная респондентами в анкете по оценке удовлетворенности потребителей: $C_{r_{analysis}} = \sum RATING_a / N$, где $RATING_a$ – оценка, поставленная респондентом в анкете; N – количество респондентов, давших оценку.
Проектирование и разработка ПС (постановка задач)	Квартал	-	Замечания по постановке задачи (категория P).	Суммарное количество сообщений категории P: $C_{r_{des}} = \sum P$.
Проектирование и разработка ПС (реализация – общий)	Квартал	Трудоемкость реализации – I_{prog} .	Трудоемкость исправления ошибок, выявленных на этапе тестирования – I_{error} .	Трудоемкость исправления ошибок, выявленных на этапе тестирования, отнесенная к трудоемкости реализации: $C_{r_{dev}} = I_{error} / I_{prog}$.
Проектирование и разработка ПС (разработка компонентов ПС)	Квартал	-	Ошибки в компонентах ПС (категория S).	Суммарное количество сообщений категории S: $C_{r_{dev_s}} = \sum S$. Может быть разбит на соответствующие этапы, входе которых выявлены ошибки (тестирование, контрольная инсталляция, сопровождение).

Продолжение таблицы 1

Процесс (подпроцесс)	Период сбора данных	Измерения в ходе процесса	Измерения после выполнения процесса	Критерий
Проектирование и разработка ПС (разработка документации)	Квартал	-	Ошибки документации (категория D).	Суммарное количество сообщений категории D: $Cr_{dev_d} = \sum D$. Может быть разбит на соответствующие этапы, входе которых выявлены ошибки (тестирование, контрольная инсталляция, сопровождение).
Проектирование и разработка ПС (разработка документации)	Квартал	-	Сообщения с вопросами по эксплуатации ПС, полученные от заказчиков (категория Q).	Суммарное количество сообщений категории Q: $Cr_{dev_q} = \sum Q$.
Проектирование и разработка ПС (тестирование)	Квартал	Общее количество ошибок, найденное в период тестирования – N_t .	Общее количество ошибок, найденных заказчиками – N_e .	Процент ошибок, обнаруженных заказчиком: $Cr_{test} = N_e / (N_e + N_t) \times 100\%$
Проектирование и разработка ПС (контрольная инсталляция)	Квартал	Количество замечаний, связанных с процессом инсталляции подсистем продукта (категория I).	-	Суммарное количество сообщений категории I: $Cr_{inst} = \sum I$

Продолжение таблицы 1

Процесс (подпроцесс)	Период сбора данных	Измерения в ходе процесса	Измерения после выполнения процесса	Критерий
Поставка (поставка и обновление версий)	Квартал	-	Анкетирование потребителей. Вопроса в анкете: "Соблюдение согласованных с потребителем сроков поставки патчей и новых версий продукции".	Средняя оценка, поставленная респондентами в анкете по оценке удовлетворенности потребителей. $Cr_{deliv} = \sum RATING_e / N,$ где $RATING_e$ – оценка, поставленная респондентом в анкете; N – количество респондентов, давших оценку.
Сопровождение (мониторинг и анализ сообщений потребителей)	Квартал	Время поступления сообщения.	Время на анализ сообщения и выработку решения.	Среднее время на анализ сообщения и выработку решения в разрезе по категориям сообщений: $Cr_{mon} = \sum T_c / N_m,$ где T_c – время, на анализ сообщения и выработку решения (по категориям сообщений), N_m – количество сообщений (по категориям).

1.3.5. Разработка и оформление документации на программные средства

Документация на ПС является органичной, составной частью любого программного продукта (ПП). Программы могут стать ПП только в совокупности с комплексом документов, полностью соответствующих их содержанию и достаточных для его освоения, применения и изменения.

Ошибки в документации не менее опасны, чем ошибки в структуре, интерфейсах, текстах программ. Поэтому к разработке, полноте, корректности и качеству документации необходимо такое же тщательное отношение, как к разработке и изменениям текстов программ и данных [71].

Документация на ПС позволяет полностью использовать всю функциональность, заложенную на этапах проектирования и разработки.

Существует три основных типа документации на ПС: проектная (технические задания, спецификации и т.п.), маркетинговая (адаптированные описания, релизы, аннотации) и техническая (эксплуатационная). Основные знания для работы с программами и их использования несет в себе техническая документация.

Комплект технической документации состоит из различных общих описаний продуктов (решений, системы) и руководств на подсистемы, входящих в различные продукты.

После принятия решения о создании ПП, создается проектная документация – ряд документов, которые определяют его вид и функциональное наполнение.

Таковыми документами обычно являются:

- Функциональная спецификация – документ, создаваемый на этапе эскизного проектирования и определяющий автоматизируемые бизнес-процессы, методы, средства и общий подход к решению задачи.
- Техническая спецификация – документ с описанием состава ПП, требований к программному интерфейсу приложений (API), структурам данных, алгоритмам и другим объектам ПС.

В технической спецификации содержится описание технической реализации функциональности, заданной в функциональной спецификации. Техническая спецификация используется на всех остальных процессах создания ПП: разработка, тестирование, документирование и т.д.

В технической спецификации содержатся сведения обо всех эксплуатационных документах, которые должны сопровождать создаваемый ПП. Техническая спецификация используется на всех этапах разработки и документирования.

Техническая документация – это описания, руководства, справочная система, сопровождающие программу или ПП. Эти документы описывают то, как работает программа и/или то, как её использовать.

Технологический процесс создания документации состоит из нескольких основных этапов (рисунок 1.9):

- определение необходимого набора документации на ПП;
- разработка документации;
- тестирование документации и внесение исправлений согласно предложениям и замечаниям тестировщика;
- перевод документации;
- фиксация документации.

Исполнителями процесса являются документатор, редактор, переводчик, тестировщик.

Документатор – лицо, осуществляющее создание документации на ПС.

Редактор – лицо, осуществляющее редактуру, корректуру и нормоконтроль документации на ПС (на определенном языке).

Переводчик – лицо, осуществляющее перевод на иностранный язык.

Тестировщик – лицо, осуществляющее проверку ПС и документации.

В процессе документирования для прояснения особенностей технической реализации и деталей функциональности, документатор проводит консультации с остальными участниками других технологических процессов (постановщиками, разработчиками и т.д.).

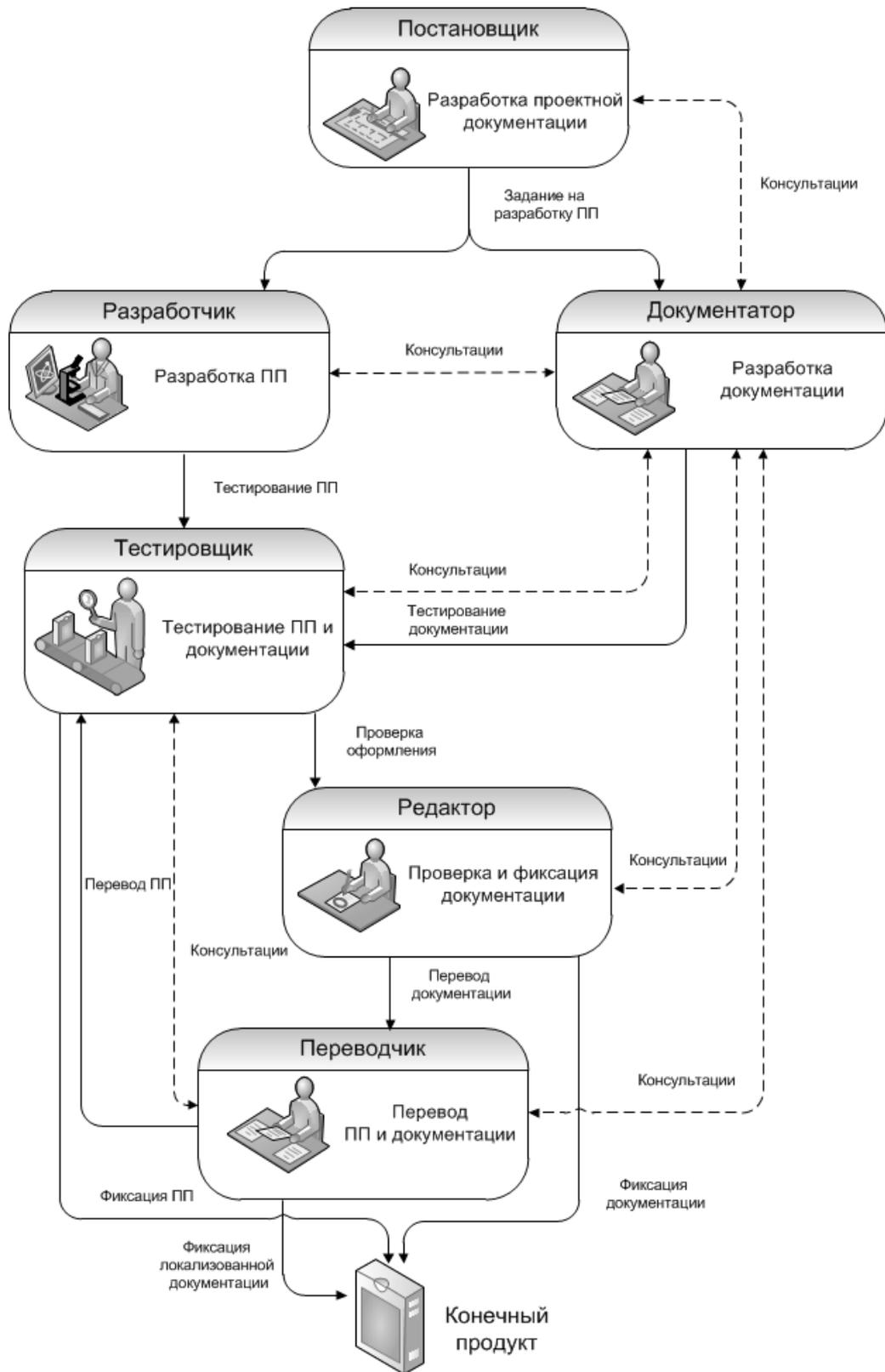


Рисунок 1.9 – Общая схема и основные участники процесса разработки документации на ПС

На исследуемом предприятии средствами технологического процесса разработки документации на ПС являются:

- Microsoft Word, Microsoft Visio – разработка документации и создание рисунков;
- WebWorks Publisher – создание справочной системы;
- Language Localizator – перевод;
- вспомогательные средства и инструменты: система управления проектами, средство версионного контроля, система регистрации ошибок, системы разработки ПС и т.д.

Сегодня не существует единого и общепринятого стандарта, регламентирующего требования к документации на ПС.

Разработка эксплуатационной документации на ПС осуществляется в соответствии с требованиями собственных стандартов предприятия, построенных на основе базовых принципов следующих стандартов:

- ГОСТ 2.601 Единая система конструкторской документации. Эксплуатационные документы.
- ГОСТ 19.XXX – группа стандартов ЕСПД.
- ГОСТ 34.201 Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
- РД 50-34.698 Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
- ГОСТ Р 51904 Программное обеспечение встроенных систем. Общие требования к разработке и документированию.

Указанные стандарты активно используются на других предприятиях – разработчиках ПО. Некоторые предприятия при разработке документации на

ПС используют стандарты серии IEEE, AECMA, DEF STAN [71]. Также стоит выделить стандарты и методологии в области управления информацией и обеспечения документирования ПС: CWA 14924, ISO 26162, ISO/IEC 19506, ISO 15489, ISO 23081, ISO/IEC 27001, ISO/IEC 27002, ISO/IEC/IEEE 26511, ISO/IEC 12207, ISO/IEC/IEEE 26512, ISO/IEC 26514 [137-140, 149-154].

Анализ указанных стандартов показал, что на сегодняшний день крайне мало результатов исследований, посвященных подходам, методам, моделям и средствам управления знаниями, содержащихся в документации на ПС, которые можно было бы использовать в базовых процессах ЖЦ ПС.

Всю совокупность разрабатываемой на предприятии программной продукции можно рассматривать как основной компонент единой информационной системы, в которой информационные процессы и методы работы с информацией осуществляются с применением средств вычислительной техники и средств телекоммуникаций.

Рассмотрим базовый ПП предприятия ЗАО «Петер-Сервис» – информационно-биллинговую систему PETER-SERVICE BIS, которая предназначена для автоматизации деятельности телекоммуникационных операторов, предоставляющих услуги в стандартах и технологиях GSM, CDMA, UMTS, WiMax, операторов фиксированной связи, а также операторов, одновременно эксплуатирующих несколько сетей названных стандартов. В состав PETER-SERVICE BIS входят более 600 тиражируемых и заказных подсистем, на каждую из которых создается свой комплект документации, а также на сам продукт. На одну подсистему и одного пользователя создается один документ. Т.е. если пользователей несколько (оператор, системный программист, программист и т.д.), то создается несколько руководств.

Документация на подсистему, как правило, покрывает только часть функциональности продукта, таким образом, получается, что документация фрагментарна. Для того чтобы получить полное описание системы, необходимо прочитать большой объем документов. Заказчику, сотрудникам компании –

разработчика ПО выгодно иметь документацию, которая будет предлагать не ссылки на множество документов, в каждом из которых надо искать интересующую информацию, а будет предоставлять возможность отбора актуальной информации по интересующей теме.

Как пример минимального знания можно привести настроечный параметр – настройка (переключатель), включающая/выключающая некоторую функциональность, изменяющая алгоритм функционирования информационной системы или разрешающая/запрещающая определенные действия. Описание параметра является структурированной информационной единицей и можно считать, что информации в ней достаточно для того, чтобы принять решение. Хотя на самом деле настроечный параметр это только информация, т.е. данные, рассматриваемые в каком-либо контексте, на основе которого пользователь может составить свое собственное понимание. Для того чтобы настроечный параметр стал знанием, его необходимо рассматривать совместно с дополнительными, связанными с этим настроечным параметром сведениями, указывающими на то, в каком бизнес-процессе применяется и что (какой другой параметр, пользовательская функция и т.п.) также влияет на этот бизнес-процесс.

Группировки настроечных параметров по подсистемам не хватает для удовлетворения потребностей поиска и работы со всей информацией о ПП. Возникает потребность уменьшения времени доступа к этой информации в условиях, когда отсутствует полное соответствие между формулировкой и искомым результатом. Создание отдельного электронного справочника настроечных параметров с категориями на ПП позволяет ускорить поиск необходимого решения. При этом работа пользователя сводится к постоянному переходу «справочник-документ-справочник», что явно неудобно. Для устранения этого перехода требуется каждый раз создавать справочник внутри документа на подсистему, или включать документ на подсистему внутрь

справочника на весь ПП. Такую документацию очень трудно поддерживать в актуальном состоянии.

Таким образом, необходимо создание такой системы управления документацией, где настроечный параметр является одной из многих информационных единиц, т.е. неделимым, с точки зрения хранения и использования, фрагментом документации, связанным с другими фрагментами.

Сегодня существуют различные системы, автоматизирующие процесс разработки документации. Наибольшую популярность среди систем такого класса завоевали системы управления контентом, основанные на технологии единого источника (single source) [105, 160].

Системы управления контентом облегчают работы с большими объемами текстовой информации, включая техническую документацию на ПС, учебные материалы и т.д. Существует несколько типов систем управления контентом:

- управление контентом Web-порталов;
- управление мультимедиа;
- управление документацией;
- корпоративные системы управления контентом;
- управление контентом на основе технологии единого источника.

Последние предоставляют возможность хранения разных типов информации (текст, графика, документы) в единой базе данных и создания разных типов документов для публикации в Web-среде и т.д. Наличие централизованного хранилища всех данных позволяет облегчить поиск уже существующей информации и избежать повторного написания многих документов.

В технической документации повтор – явление не менее редкое, чем в тексте программы. Приходится описывать сходные функции программ и систем, сходные формы, сходные действия пользователей. Описания одних и тех же объектов приходится полностью или частично дублировать в разных

документах. Поскольку техническая документация – это текст на естественном языке, применить к нему модульный подход оказывается сложнее, чем к исходному коду программы. В нем хватает нерегулярностей, обусловленных лексическими и грамматическими исключениями, сложившимися традициями, соображениями стилистики и эстетики [92].

Принцип единого источника и реализующие его технологии позволяют применить модульный принцип к документированию ПС. Применение технологии единого источника требует определенной подготовки. Сначала проектируется структура единого источника, разрабатываются шаблоны и стили оформления, устанавливается и настраивается инструментарий для формирования документов.

Принцип единого источника в документировании, как и модульный принцип в программировании, помогает организовать работу коллектива документаторов, распределив между ними более-менее изолированные подзадачи и области ответственности. Таким образом, единый источник – это не только техническое, но еще и организационное решение.

Формирование документов на основе принципа единого источника – основной принцип, на котором сегодня построены все наиболее известные и развитые инструментальные средства для автоматизации документирования: AuthorIT, DITA, DocBook/XML, Rational SoDA и др. [3, 85, 93, 127, 135].

Применение технологии единого источника в особенности целесообразно при разработке и длительном сопровождении комплектов документации на сложные решения: программные и программно-аппаратные комплексы, автоматизированные системы, большие корпоративные ИТ-инфраструктуры. Документирование таких решений в особенности требует эффективности при создании и обновлении наборов типизированных документов, описывающих типизированные предметы. Кроме того, при документировании крупных комплексов и систем исключительно важной становится масштабируемость комплекта документации, возможность быстро распространить его на новые

предметы (компоненты, подсистемы, пользовательские роли, операции, процессы) и быстро освоить выпуск новых типов документов.

Технология единого источника позволяет все изменения делать один раз, исправления в каждый конкретный документ при этом вносятся автоматически. Система управления контентом автоматически обновляет все ссылки на общие фрагменты информации, так что в документации никогда не возникает путаницы. В отличие от обычного средства разработки документов в XML-формате система управления контентом автоматизирует документооборот. В то же время, в отличие от простой системы управления контентом, использование технологии единого источника позволяет выпускать документы в требуемом формате быстро и с минимальными затратами. Система управления контентом на основе единого источника снижает затраты на локализацию (перевод) документации.

В настоящее время системы разработки технической документации, построенные на основе принципа единого источника, внедряются и успешно применяются в компаниях самого разного масштаба и профиля. Использование таких систем позволяет автоматизировать сам технологический процесс разработки документации, однако не позволяют реализовать весь комплекс задач управления знаниями, содержащихся в документации на ПС (например, структурирование знаний, поиск с учетом семантических свойств и связей и т.д.).

1.4. Системы управления знаниями

1.4.1. Классификация систем управления знаниями

Для эффективного управления знаниями на современном предприятии должна быть создана технологическая инфраструктура – информационно-программная система управления знаниями (СУЗ) [15, 17, 50, 111, 116].

СУЗ – это информационно-программная система, которая систематически и целенаправленно поддерживает процессы преобразования всех видов

знаний/информации в организации [158]. СУЗ решает вопросы создания на предприятии единого информационного пространства, организации групповой работы сотрудников для приобретения, представления и обмена знаниями, предоставления доступа к единой корпоративной базе знаний и создает условия для эффективного использования знаний сотрудников в общих интересах.

СУЗ является системой, которая объединяет и интегрирует функции для контекстной обработки как явных, так и скрытых знаний всей организации или той ее части, на которую нацелена инициатива по УЗ [111]. Информационные технологии не решают всех проблем УЗ, напротив, они могут породить и дополнительные сложности – при разработке и внедрении таких систем. Однако, при умелом использовании, программные средства УЗ могут помочь перевести инфраструктуру знаний предприятия на новый материально-технологический уровень, и превратить ее в мощный инновационный ресурс.

Задача СУЗ – накапливать не разрозненную информацию, а структурированные, формализованные знания – закономерности и принципы, позволяющие решать реальные производственные и иные задачи. Основная цель СУЗ – сделать знания доступными и повторно используемыми [27, 116].

Процесс управления знаниями в общем случае содержит этапы извлечения и обработки знаний, которые реализуются в СУЗ. При этом технологические составляющие этапа извлечения определяют, каким образом информация превращается в знания, а составляющие этапа обработки определяют то, как из знаний порождается новое знание, используемое для принятия решений, что находит отражение в двух различных *типах* СУЗ, автоматизирующих два этапа работы со знаниями [50].

В первом случае в центре внимания оказывается этап извлечения знания, поддержка которого обеспечивается информационно-поисковой составляющей СУЗ. Этап обработки найденных знаний реализуется за рамками СУЗ в голове аналитика или лица, принимающего решения. В итоге, ключевым компонентом СУЗ является поисковая машина, обеспечивающая оперативный отбор и

доставку адекватной информации по запросам. Поисковый компонент является предметно-независимым или допускает простую настройку даже малоподготовленным пользователем, что определяет основное достоинство этого типа СУЗ: применимость к широкому кругу заранее неизвестных задач.

Во втором случае основное внимание направлено на этап обработки знаний, получения из него нового знания на основании обобщения, сопоставления, логического вывода и т.п. В зависимости от формальной модели представления знаний, на этом этапе могут применяться различные математические методы. Результатом являются обобщения, выявление скрытых зависимостей, прогнозы. В этом случае за рамками СУЗ реализуется этап извлечения знаний, их формализации и размещения в базе знаний, логическая структура и фактическое наполнение которой зависят от особенностей предметной области и должны разрабатываться в тесном взаимодействии экспертов и инженера по знаниям. Узкая специализация базы знаний и трудоемкость ее разработки являются недостатками СУЗ данного типа. Достоинством же СУЗ является возможность быстрого получения решения для тех типовых задач, на решение которых они ориентированы.

Использование только компонент СУЗ первого типа при создании СУЗ в базовых процессах ЖЦ ПС не позволяет учесть семантику информации, находящейся в документации на ПС, что существенно образом оказывает влияние на качество и эффективность процессов. Использование только компонент СУЗ второго типа потребовало бы затрат на реализацию СУЗ и создание базы знаний, существенно превышающих затраты на разработку самих ПС.

Таким образом, создание СУЗ в базовых процессах ЖЦ ПС предложено выполнять на основе комбинации двух типов СУЗ. Ключевым компонентом такой СУЗ является подсистема поиска, обеспечивающая оперативный отбор и выдачу релевантной информации по запросам. При этом поиск информации производится не в хранилище документов, а в базе знаний.

Различают системы *вертикального* и *горизонтального* управления знаниями [30].

Системы вертикального управления знаниями, как правило, преследуют тактические цели (решение прикладных задач) и разрабатываются для конкретных ситуаций, встречающихся в процессе функционирования организаций. Такие системы доказали свою высокую эффективность, но являются очень узконаправленными и не могут быть перенесены на другие ситуации. Тем не менее, такие СУЗ могут быть одним из решающих факторов в достижении стратегической цели предприятия.

Системы горизонтального управления знаниями являются системами, которые могут быть применены ко многим управленческим ситуациям. Фактически они представляют собой технологии, поддерживаемые программно-инструментальными комплексами, которые позволяют решать весь комплекс задач управления знаниями и локализовать его применительно к конкретной управленческой ситуации.

СУЗ вынуждены интегрировать разнообразные технологии:

- электронная почта и Интернет;
- системы управления базами данных и сами базы данных;
- средства создания хранилищ данных;
- системы поддержки групповой работы;
- локальные корпоративные системы автоматизации;
- системы документооборота;
- экспертные системы и др.

При этом ни одна из этих указанных технологий (кроме последней) не включает знания в контексте интеллектуальных (экспертных) систем [24, 25].

В таблице 2 приведены примеры подобных систем, упорядоченные в соответствии с их ролью в процессах УЗ [179].

Таблица 2 – Сравнение традиционных и интеллектуальных систем в процессах УЗ

Типы систем	Обмен знаниями	Распределение знаний	Сбор кодирование и знаний	Создание знаний
Традиционные системы (приложения созданные на основе обычного программного обеспечения)	Е-mail, групповая работа, дискуссионные группы, технологии Р2Р, интранет-порталы	Текстовые процессоры, системы обработки текстов, управление документами	–	Программы поддержки мозгового штурма, построение карт знаний, статистический анализ
Интеллектуальные системы (приложения созданные на основе интеллектуальных программных средств)	Основанные на онтологиях порталы	Экспертные системы, системы работы с лучшими практическими решениями и полученными уроками	Инструменты для приобретения и кодирования знаний	Системы анализа и обнаружения знаний, закономерностей в данных, системы поддержки творческой работы

В работе [22] все инструменты СУЗ разбиты на семь классов:

1. Анализ имеющейся в компании информации.
2. Извлечение, структурирование и формализация знаний (инженерия знаний).
3. Обмен и использование знаний.
4. Поиск информации по запросу.
5. Организация хранения.
6. Обучение.
7. Интегрированные системы обработки, объединяющие вышеперечисленные функции.

Ценность данной классификации заключается в представлении единого взгляда на весь комплекс задач управления знаниями.

Среди многообразия программных средств выделяет следующие категории, которые поддерживают решение задач УЗ [23] (рисунок 1.10):

1. Средства доступа к информации (поиска информации). Лидерами в данной категории являются продукты компаний Microsoft (FAST Search Server, SharePoint Server), IBM (Classification Module, WebSphere Portal) и Autonomy (IDOL Server) [86, 125].
2. Средства совместной работы. Эти программные продукты поддерживают коммуникации (мгновенные сообщения, почту, IP-телефонию, видеоконференции) и предоставляют единую виртуальную рабочую среду, совместную работу с документами, рабочие доски и пр. Так можно организовать совместную разработку документа в едином пространстве, выделив сообщество участников, или организовать коллективное обсуждение. Программы этой категории создают персональные профили сотрудников с описанием интересов, компетенции, опыта работ и т.п. Лидерами в данной группе являются продукты компаний Microsoft (SharePoint), IBM (Lotus Notes, Lotus Domino), Oracle (Fusion Middleware) и Socialtext (Socialtext) [98, 163].
3. Системы управления контентом. Данную категорию можно рассматривать как инфраструктуру для представленных выше групп. Эта группа программ включает различные инструменты для гибкой и надежной работы с контентом, например, регистрацию, управление версиями, безопасность, сканирование и распознавание. Также к данной категории относят создание архивов, наполнение, обновление и структурирование содержимого сайтов, организацию взаимодействия с посетителями сайтов. Лидерами в данной категории являются продукты компаний Microsoft (SharePoint), IBM (Enterprise Content Manager) и EMC (Documentum) [98, 123, 128].
4. Web 2.0 инструменты для работы со знаниями [131]. Неполный список инструментов Web 2.0: интернет почта (mail), средства обмена сообщениями (ICQ и т.д.); блоги – online журналы и дневники, позволяющие обмениваться мнениями, размещать новости; социальные сети, позволяющие узнать о знаниях, опыте, интересах других людей и представить свои, средства

интеграции и чтения новостных RSS-потоков; P2P-сети – средство эффективного обмена информацией; wiki-средства – средства создания и функционирования Web-сайтов. Страницы wiki-сайта совместно редактируются сотрудниками организации и выступают альтернативой письмам и вложениям электронной почты.

5. средства структурирования знаний. Эта группа программ включает различные инструменты работы со знаниями, например, редакторы онтологий (Protege, Ontolingua, Altova SemanticWorks), программы визуализации рассуждений и т.д.
6. Компоненты УЗ в специализированных системах. К данной категории относятся библиотеки, справочники и шаблоны в системах автоматизированного проектирования и управления проектами и процессами, аналитические компоненты в системах управления взаимоотношениями с клиентами и поставщиками и т.д.



Рисунок 1.10 – Систематизация программных средств для СУЗ

Две первые категории соответствуют двум различным взглядам на знание [124, 157]:

1. Знание как «запас». Согласно этому подходу, знание – это актив, продукт, который может существовать независимо от человека и управление которым подобно управлению любыми другими ресурсами.
2. Знание как «поток». В таком подходе знание не может передаваться как некий объект, поскольку в формирование знания вносит существенный

вклад человек-участник процесса передачи. Данный взгляд объясняется тем, что знание – результат интерпретации информации в определенном контексте. Знание не передается, а воссоздается в мозгу человека при получении информации в соответствии с исходными знаниями, опытом, ценностями и ожиданиями.

На практике оба взгляда часто сочетаются.

Основными поставщиками интегрированных решений для разработчиков программных средств управления знаниями являются компании Microsoft и IBM. Эти компании стараются покрыть функциональными возможностями своих программных продуктов весь комплекс задач управления знаниями.

Таким образом, многообразие технологий и программных средств для СУЗ, специфика конкретных предприятий показывают, что процессы УЗ не могут осуществляться за счет простой инсталляции и запуска готовых приложений, но могут поддерживаться без создания новых программных средств и базироваться на существующих платформах.

На исследуемом предприятии ЗАО «Петер-Сервис», как уже было отмечено в разделе 1.2, внедрен продукт Microsoft SharePoint с целью обеспечения совместной работы, предоставления средств управления контентом и представления доступа к сведениям, важным для организационных целей и процессов предприятия.

По данным известной компании Gartner, продукт Microsoft SharePoint на протяжении 2010-2013 годов занимает лидирующее место в третьей категории, указанной выше (рисунок 1.11) [86, 128, 143, 144]. Также данный продукт и его расширения входят в первые две категории. Поэтому целесообразно рассмотреть его функциональные возможности с точки зрения технологической платформы для построения СУЗ, основным ресурсом знаний которой является документация на ПС.

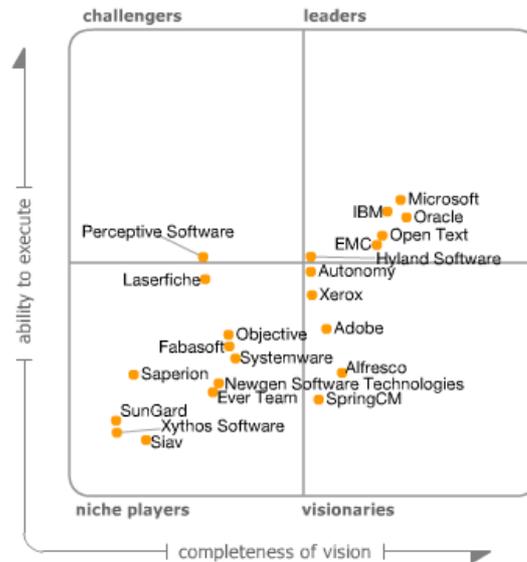


Рисунок 1.11 – Анализ сегмента рынка систем управления контентом

Microsoft SharePoint представляет собой бизнес-платформу для организации совместной работы, которая позволяет объединить сотрудников и предоставить им широкие возможности формального и неформального взаимодействия, как в пределах предприятия, так и с партнерами и клиентами. Платформа также позволяет эффективно управлять контентом на протяжении всего ЖЦ информации.

В Microsoft SharePoint есть три уровня функциональных возможностей, для каждого из которых заданы соответствующие лицензии: SharePoint Foundation; SharePoint Server Standard Edition; SharePoint Server Enterprise Edition. Конкретные лицензии, которые требуются для использования SharePoint в организации, зависят от таких факторов, как задействованные возможности, сценарий развертывания и место размещения. SharePoint Foundation является бесплатным продуктом. У организаций, использующих SharePoint Foundation, должна быть соответствующая лицензия на Windows Server [75].

С выходом версии 2010 функциональность стандартных возможностей SharePoint расширяется технологиями поиска и бизнес-аналитики, позволяя

пользователям быстро принимать взвешенные управленческие решения, а также новыми инструментами создания Web-приложений, которые предназначены для быстрого построения необходимых сотрудникам решений.

SharePoint дает возможность увеличить производительность труда за счет использования интегрированного набора гибких функций, позволяет объединить данные из различных систем компании в общее решение за счет использования единого входа в систему и возможностей интеграции корпоративных приложений. Возможности совместной работы в SharePoint, а также поддержка настроек SharePoint Designer, позволяют с легкостью создавать доступные по всему предприятию сайты SharePoint и управлять ими [98].

Функциональные возможности продукта SharePoint (рисунок 1.12):

1. Создание сайтов.

Сайты SharePoint обеспечивают создание единой инфраструктуры для построения порталов и средств совместной работы во внутренней сети, Экстранете и Интернете. Объединяют пользователей, предоставляя им возможность обмениваться информацией, данными и опытом.

2. Создание сообществ и сетей.

Сообщества в SharePoint позволяют работникам организовывать совместную работу наиболее удобным для них способом. Обеспечивают совместную работу сотрудников в группах, обмен знаниями и идеями, взаимодействие с коллегами, а также простой поиск информации и экспертов.

3. Управление контентом.

Управление контентом в SharePoint позволяет пользователям участвовать в контролируемом и соответствующем требованиям цикле управления контентом. Возможности SharePoint обеспечивают достижение оптимального баланса между опытом пользователей, политиками и процессами.

4. Корпоративный поиск.

Функции поиска в SharePoint дают пользователям возможность находить контент, информацию и сотрудников за счет использования интегрированной и простой в управлении платформы, построенной на базе лучшей технологии корпоративного поиска.

5. Бизнес-аналитика.

Средства бизнес-аналитики в SharePoint дают пользователям возможность получать доступ и использовать информацию из структурированных и неструктурированных источников. Предоставляет пользователям средства исследования информации и поиска нужных сотрудников и экспертов, чтобы ускорить принятие бизнес-решений и повысить их точность.

6. Конструирование приложений.

Конструктор приложений в SharePoint позволяет пользователям быстро реагировать на потребности бизнеса путем создания собственных решений с написанием программного кода и без, что достигается за счет использования набора стандартных блоков, программных средств и функций самообслуживания.



Рисунок 1.12 – Функциональные возможности продукта
Microsoft SharePoint Server

Для построения СУЗ, где основным ресурсом знаний является документация на ПС, могут быть использованы функциональные возможности продукта SharePoint «управление контентом» и «корпоративный поиск» [44].

Управление контентом

Управление контентом – это управление документами и другими типами контента, а также их хранение, обработка и доставка в масштабах предприятия. В Microsoft SharePoint Server управление контентом разделено на следующие основные функциональные области: управление медиа-контентом, управление документами, управление записями и управление метаданными [95, 96, 119].

Управление медиа-контентом предоставляет функциональные возможности для сбора, комментирования, хранения, извлечения и отображения цифрового контента, включая изображения, звук и видео.

Управление документами предоставляет функциональные возможности для создания, хранения и отслеживания документов, наборов документов и других типов контента. В состав предоставляемых функциональных возможностей также входят организатор контента, шаблон для центра документов и инфраструктура, требуемая для поддержки создания наборов документов. Кроме того, в состав управления документами входит технология, предназначенная для навигации по наборам документов и фильтрации на основе метаданных, а также способ поиска документов на основе значений по умолчанию в метаданных.

Управление записями предоставляет функциональные возможности для идентификации документов в качестве записей и способы управления записями на протяжении их ЖЦ, т.е. идентификация и создание записей, eDiscovery (удержание, аудит, назначение срока действия и удаление).

Управляемые метаданные – это иерархически упорядоченная, централизованно управляемая совокупность терминов, которые можно определять и использовать в качестве метаданных в Microsoft SharePoint Server и других приложениях Microsoft Office. Метаданные являются управляемыми,

поскольку управление терминами или наборами терминов (совокупность терминов) осуществляется независимо от столбцов или полей, для которых они требуются.

Термин – это слово или фраза, которую можно связать с элементом в SharePoint Server. Набор терминов – это коллекция связанных терминов. Для столбца SharePoint Server можно указать, что он должен содержать термины из конкретного набора терминов. Управляемые метаданные позволяют отразить тот факт, что термины и наборы терминов создаются и изменяются независимо от столбцов.

Локальные наборы терминов создаются в контексте семейства сайтов. Например, при добавлении столбца в список в библиотеке документов и создании нового набора терминов для привязки к этому столбцу этот набор терминов становится локальным по отношению к семейству сайтов, в котором хранится библиотека документов. Глобальные наборы терминов создаются вне контекста семейства сайтов.

Корпоративный поиск

Поиск в Microsoft SharePoint позволяет сотрудникам компании находить информацию, необходимую для выполнения работы. Доступны функции поиска в корпоративной сети, поиска людей и платформа для создания приложений на основе поиска. Уникальность поиска в SharePoint заключается в сочетании релевантности, точности и ориентированности на пользователя [97].

SharePoint Server предоставляет конечным пользователям возможность создавать и выполнять эффективные поисковые запросы. Поддерживается использование в поисковых запросах логических операторов AND, OR и NOT. Можно использовать звездочку (*) в качестве подстановочного знака в конце строки текста. По мере ввода пользователем ключевых слов в поле поиска центр поиска предоставляет подсказки (предложения), чтобы помочь завершить ввод запроса. Эти предложения выдаются на основе предыдущих запросов

других пользователей. Кроме того, когда система поиска возвращает результат запроса, центр поиска отображает предложения для связанных поисков.

В SharePoint Server имеются средства, позволяющие повысить релевантность и полезность результатов поиска. Если какой-либо документ в наборе результатов поиска посещался со страницы результатов поиска гораздо более часто, чем другие документы, то он при ранжировании результатов поиска он выдвигается на передний план. При выполнении обхода контента в качестве части процесса индексирования выполняется синтаксический анализ метаданных. Метаданные – это данные о данных. А если учесть, что под данными понимаются объекты любой формы, выступающие в качестве средства представления информации в информационной системе, то метаданные – это семантическая информация об информации.

В некоторых случаях система поиска может также логически вывести метаданные из контента документа. Это может быть полезным, когда метаданные документа отсутствуют или неправильны. Например, в презентации Microsoft PowerPoint может быть не указан автор, но система поиска может определить автора по фразе в документе.

Таким образом, продукт Microsoft SharePoint обладает необходимым набором базовой функциональности для разработки СУЗ, основным ресурсом знаний которой является документация на ПС.

1.4.2. Архитектура современной системы управления знаниями

Рассмотренные программные средства могут использоваться для создания СУЗ в произвольной предметной области. Однако почти в любой компании существуют свои программные средства автоматизации. Например, дизайнеры используют ПП для обработки изображений, проектировщики системы автоматизированного проектирования, программисты среды программирования, менеджеры и бизнес-аналитики системы организационного моделирования и проектирования. Задача формирования СУЗ на программном

уровне требует сопряжения отдельных программных систем в единую систему, ядром которой является некоторая база знаний. На сегодня не существует единого стандарта на формат такой базы, но основное направление связано с онтологиями и языками их описания [17, 45, 46, 49, 90, 94, 104, 109, 112].

Типовая современная архитектура СУЗ представлена на рисунке 1.13 [23].



Рисунок 1.13 – Пример архитектуры СУЗ

Нижний уровень архитектуры обеспечивает получение знаний из разнородных источников структурированной (базы данных и знаний, таблицы, формы) и неструктурированной (документы) информации. Получение знаний подразумевает аннотирование разнородных источников информации с помощью онтологии, которая позволяет отразить содержание информации. Для получения онтологии используются программные средства

автоматического формирования онтологии, поддержки ручной разработки и редактирования онтологии, средства слияния, объединения и оценки качества онтологий. Для аннотирования используются средства автоматической классификации документов по онтологии, извлечения информации из текста и баз данных.

Промежуточный уровень обеспечивает хранение онтологий и метаданных, управление версиями, доступом, передачу данных для взаимодействия с внешними системами и хранилищами, обработку запросов и логический вывод.

Пользовательские приложения предоставляют пользователям доступ к знаниям, находящимся в системе. Доступ может осуществляться с помощью поискового механизма, средств навигации и визуализации, а также путем непосредственного обмена знаниями между пользователями (например, путем обмена ссылками на ресурсы). Взаимодействие пользовательских приложений с промежуточным уровнем осуществляется с помощью языков запросов и манипулирования данными.

Представленная архитектура может воплощаться в СУЗ полностью или частично. В одних случаях упор может делаться на программную реализацию задач уровня пользовательских приложений – поиск, визуализацию знаний [5, 48, 94, 145, 155], в других – на получение, хранение, интеграцию знаний [52, 53, 57, 99, 133, 169].

1.5. Модели представления знаний

1.5.1. Классификация и основные типы моделей представления знаний

Существуют различные способы описания знаний. Естественный язык является одним из таких способов. Использование естественного языка для работы со знаниями в информационных системах является плохим выбором,

так как для него характерны отсутствие формализма, неявный синтаксис и семантика. Особой проблемой при создании баз знаний с текстами на русском языке является его флективность. Это усложняет создание баз знаний и не позволяет применять методы, используемые при создании англоязычных баз знаний.

Для компьютерного представления и обработки знаний и данных о предметной области (об объектах, процессах, явлениях, их структуре и взаимосвязях), они должны быть формализованы и представлены в определенном виде. Для формализации и представления знаний разрабатываются специальные модели представления знаний и языки описания знаний.

Модели представления знаний – это одно из важнейших направлений исследований в области искусственного интеллекта [56, 59, 76, 168]. На сегодняшний день разработано достаточное количество моделей. Каждая из них обладает своими плюсами и минусами, и поэтому для каждой конкретной задачи выбирается наиболее подходящая модель. От этого зависит не столько эффективность выполнения поставленной задачи, сколько возможность ее решения в требуемые сроки и в объеме выделенных ресурсов. Модели представления знаний относятся к прагматическому направлению исследований в области искусственного интеллекта. Это направление основано на предположении о том, что мыслительная деятельность человека представляет собой «черный ящик». При таком подходе не ставится вопрос об адекватности используемых моделей представления знаний тем моделям, которыми пользуется в аналогичных ситуациях человек, а рассматривается лишь конечный результат решения конкретных задач.

Не существует единой системы классификации моделей представления знаний. В контексте рассматриваемой темы можно привести следующую классификацию моделей представления знаний.

По идеям, лежащим в их основе, с точки зрения математической обоснованности, модели представления знаний делятся на *эмпирические* и *теоретические*.

Первый подход, называемый эмпирическим, основан на изучении принципов организации человеческой памяти и моделировании механизмов решения задач человеком. На основе этого подхода в настоящее время разработаны и получили наибольшую популярность следующие модели: продукционные модели, семантические сети и фреймовые модели.

Второй подход можно определить как теоретически обоснованный, гарантирующий правильность решений. Он, в основном, представлен моделями, основанными на формальной логике (исчисление высказываний, исчисление предикатов), формальных грамматиках, комбинаторных моделях, в частности моделями конечных проективных геометрий, теории графов, тензорными и алгебраическими моделями. В рамках этого подхода до настоящего времени удавалось решать только сравнительно простые задачи из узкой предметной области.

Модели представления знаний в интеллектуальных информационных системах традиционно делятся на *логические (формальные)* и *эвристические (формализованные)* [91]. К логическим моделям относятся:

- логика высказываний;
- логика предикатов;
- трехзначная логика;
- модальные логики;
- нечеткая логика и т.д.

К эвристическим моделям представления знаний можно отнести:

- деревья решений;
- семантические сети;
- фреймы и сети фреймов;

- онтологии;
- объектно-ориентированное программирование;
- реляционную алгебру;
- правила-продукции т.д.

В современных интеллектуальных информационных системах, в основном, применяются четыре типа моделей представления знаний:

- продукционная модель;
- формально-логическая модель;
- семантико-сетевая модель;
- фреймовая модель.

Первые два типа также часто называют *модульными* в связи с тем, что они оперируют элементами процедурных знаний – правилами и аксиомами, являющимися истинными (имеющими смысл) в конкретной предметной области. Фреймовые модели и семантические сети часто называют *сетевыми* в связи с тем, что связи между элементами знаний в этих моделях выражаются через бинарные отношения. Рассмотрим данные модели более подробно.

Продукционная модель

Продукции (наряду с сетевыми моделями) являются наиболее понятными и популярными средствами представления знаний в информационных системах [56, 70, 84]. Продукционная модель основана на правилах, позволяющих представить знания в виде предложений типа «Если (условие), то (действие)». Под «условием» (антецедентом) понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под «действием» (консеквентом) – действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

Правила, по сравнению с другими способами представления знания имеют следующие преимущества [84]:

- 1) модульность;
- 2) единообразие структуры;
- 3) естественность (вывод заключения в такой системе аналогичен процессу рассуждения эксперта);
- 4) гибкость иерархии понятий, которая поддерживается только как связи между правилами (изменив правило, вы можете изменить иерархию).

Недостатки продукционных моделей:

- 1) процесс вывода менее эффективен, чем при других способах представления, так как большая часть времени затрачивается на непроизводительную проверку применимости правил;
- 2) этот процесс трудно поддается управлению;
- 3) сложно представить иерархию понятий.

Представление знаний в виде правил иногда называют плоским (по аналогии с реляционными базами данных), так как в них отсутствуют средства для установления иерархии правил. Объем базы знаний растет линейно по мере включения в нее новых фрагментов знаний. Большинство существующих коммерческих экспертных систем основаны на правилах.

Логическая модель

Логические модели представления знаний – это модели, основанные на правилах формальной логики [64]. Все логические модели представляются четверкой:

$$M = \langle T, P, A, F \rangle, \quad (2)$$

где T – алфавит (множество базовых элементов); P – множество правил построения синтаксически правильных выражений; A – априорно истинные выражения (аксиомы); F – правила вывода новых истинных выражений в рамках модели M .

В качестве T и P в логических модели в настоящий момент чаще всего используются исчисление высказываний и исчисление предикатов первого

порядка, преобразованные к определенному виду – хорновским дизъюнктам, F-метод линейных резолюций.

Логические модели хорошо зарекомендовали себя при создании баз знаний, имеющих характер высокомодульных структур. Такие знания характерны единственностью теоретического обоснования, наличием системы формально точных определений и выводов. В связи с этим описательная мощность формально-логических моделей, как системы представления знаний, выше, чем у других. Используемый в логике исчисления предикатов метод резолюций является одним из наиболее эффективных, высоко формализуемых и широко применяемых методов.

Однако достоинства этих моделей и их развитость, если речь идет об исчислении предикатов первого порядка, обуславливают и присущие им недостатки: трудночитаемость логических описаний и невысокая производительность обработки знаний. Даже простые утверждения не так просто перевести с естественного языка на язык исчисления предикатов, если необходимо точно отобразить все аспекты исходного текста. Кроме того, человеческой логике присуще использование в высказываниях нечетких или неоднозначных сравнительных и выразительных категорий, а формальная логика не в состоянии оперировать ими.

Семантико-сетевая модель

Семантико-сетевая структура представления знаний подразумевает создание информационной модели предметной области, напоминающей некую сеть или граф [56, 67, 70, 84].

Элементами конструкции семантической сети являются узлы и дуги, соединяющие узлы. Сама сеть может быть упорядоченной, т.е. представлять собой ориентированный граф, или неупорядоченной, т.е. представлять собой неориентированный граф. Узлы такой сетевой структуры отображают значимые понятия (сущности) предметной области.

В семантико-сетевых моделях декларативные знания не отделены от процедурных. А это означает, что база знаний должна содержать и набор правил, с помощью которых может быть сформирована процедура логического вывода, обеспечивающая решение задачи, содержащейся в запросе к системе.

Достоинство семантико-сетевых моделей состоит в том, что они естественнее других представляют знания об отношениях между сущностями предметной области и их свойствами. Эти модели обладают преимуществами при описании с помощью бинарных отношений всех семантически значимых (для различных классов решаемых задач) типов межобъектных связей статической или динамической природы.

Вместе с тем, расширение семантической сети, ее дополнение новыми фактами, связано с проблемой группирования на основе общности признаков новых фактов. При этом неизбежны трудности, связанные с нарушением однородности сети. Нарушение однородности в представлении отображаемой сетью информации приводит к необходимости увеличения арсенала методов решающего вывода. С увеличением размеров сети существенно растет время поиска объектов и связей, участвующих в реализации конкретной вычислительной или логической процедуры. В этом аспекте сетевые модели проигрывают моделям, не имеющим инструментов вычисления оптимальных стратегий поиска.

В зависимости от класса задач, для решения которых создается интеллектуальная система, в семантико-сетевой модели представления знаний может оказаться затруднительным учет и других типов связей прагматического, синтагматического или парадигматического характера.

Фреймовая модель

Фреймовая модель представления знаний основана на фреймовой теории, предложенной М.Минским в 1974 году, которая представляет собой систематизированную в виде единой теории психологическую модель памяти человека и его сознания [80].

Важным моментом во фреймовой теории является понятие фрейма – определенной структуры данных для представления некоторого концептуального объекта. Информация, относящаяся к конкретному фрейму, содержится в слоте. Все фреймы объединяются в иерархическую структуру, интегрирующую в себе декларативные и процедурные знания. Данная структура отображает целостный образ знаний, которому свойственна иерархичность концептуального представления.

Каждый фрейм описывает один концептуальный объект, а конкретные свойства этого объекта содержатся в слотах. В качестве слота может использоваться специфичная процедура вывода – присоединенная процедура. Фреймовую систему без механизма присоединенных процедур часто рассматривают как базу данных системы продукций.

Представление знаний, основанных на фреймах, является альтернативным по отношению к системам, основанным на правилах: оно дает возможность хранить иерархию понятий в базе знаний в явной форме [56, 70, 84].

Основными достоинствами фреймовых моделей являются:

- 1) возможность сочетания в одной структуре декларативных и процедурных знаний;
- 2) возможность иерархий в соответствии со степенью абстрактности понятий;
- 3) возможность организации любой системы вывода на основе объектно-ориентированного принципа управления выводом и обменом свойствами и процедурами вычисления их значений между объектами;
- 4) адекватное отображение концептуальной организации памяти человека и ее способности гибко и наглядно представлять сильно структурированные знания.

Фреймовые модели, если исключить из структур фреймов присоединенные процедуры, могут применяться в интеллектуальных системах,

использующих продукционную модель представления процедурных знаний, для реализации декларативной компоненты базы знаний.

Основным недостатком фреймовых моделей является то, что создание на их основе системы требует привлечения высокопрофессиональных специалистов с практическим опытом создания фреймовых структур и достаточно длительного времени на доведение системы до рабочего состояния.

1.5.2. Онтологический инжиниринг

Для того чтобы определить структуру взаимосвязей между элементами системы знаний предприятия, необходимо выделить (абстрагировать) понятия из содержания элементов знаний (документов, опыта сотрудников) и структурировать (организовать) их формальным способом, путем задания взаимосвязей между этими понятиями. Одно из важнейших и перспективных направлений в области формализации знаний в виде множества понятий и взаимоотношений между ними является онтологический инжиниринг – технология проектирования и разработки онтологий [1, 15, 16, 23, 57, 60, 61, 90, 141, 177].

В 1980-х годах термин «онтология» перешел из философии в область информационных технологий, когда начал использоваться исследователями по искусственному интеллекту вначале в области инженерии знаний, при обработке естественных языков, а затем в представлении знаний. В конце 1990-х годов начались активные исследования возможности использования онтологии в таких областях, как интеграция информации, поиск информации в Интернете и управление знаниями [27, 60, 61, 68, 83].

На сегодняшний день нет общепринятого определения понятия «онтология». Наиболее известным являются следующие определения понятия:

- Онтология – спецификация на концептуальном уровне [146].

- Онтология – база знаний, описывающая факты, которые предполагаются всегда истинными в рамках определенного общества на основе общепринятого смысла используемого словаря [147].
- Онтология — это структурная спецификация некоторой предметной области, ее формализованное представление, которое включает словарь (или имена) указателей на термины предметной области и логические выражения, которые описывают, как они соотносятся друг с другом [25].
- Онтология – это словарь предметной области, наполненный множеством точных определений, или аксиом, ограничивающих смысл терминов в этом словаре и позволяющий согласованно интерпретировать данные [148].

В качестве рабочего определения понятия «онтология» в настоящей работе используется следующее:

Онтология – модель представления знаний какой-либо предметной области в виде совокупности понятий этой предметной области и существующих между ними отношений.

Таким образом, онтология представляет модель предметной области в виде некоторой сетевой структуры, в которой семантика каждого понятия определяется через его отношения с другими понятиями. Причем во множестве отношений существует отношение типа «класс-подкласс», упорядочивающее понятия предметной области в иерархию – таксономию понятий. К отношениям общего типа также относится и отношение «целое-часть».

Важно отметить, что таксономия позволяет одному понятию-потомку иметь несколько понятий-родителей. Именно таксономия является той составляющей, которая отличает онтологию от наиболее близких к ней сетевых моделей представления знаний (семантико-сетевых и фреймовых моделей) [111]. Задачи, которые решают исследователи в области искусственного интеллекта с помощью онтологии, отличаются от задач в других областях компьютерной науки. Целью искусственного интеллекта является создание интеллектуальных информационных систем, имитирующих интеллектуальную

деятельность человека. Подобная система должна быть способной заменить человека в какой-либо области деятельности. Однако онтология может быть использована и для других целей – описания информационных ресурсов предприятия, учитывающих семантику обрабатываемой информации. Для этой цели используется онтология в данной работе.

Классификация онтологий

Существуют различные классификации онтологий. Это связано с разнообразием задач, решаемых с помощью онтологий.

По ***уровню выразительности***, согласно классификации в [14, 83], онтологии подразделяют на легковесные и тяжеловесные. Легковесные онтологии – это простые таксономические структуры примитивов или композиции термов с соответствующими определениями. Тяжеловесные онтологии сильно аксиоматизированы, такой уровень аксиоматизации позволяет осуществлять онтологическое связывание явно. Цель аксиоматизации – избежать терминологической и концептуальной неоднозначности из-за неправильной интерпретации. Каждая тяжеловесная онтология может иметь легковесную версию. Как и в определении любых размеров, граница между тяжеловесными и легковесными онтологиями размыта.

По ***степени формальности***, согласно классификации в [14, 83, 89, 175], онтологии подразделяют на:

- Неформальные онтологии, выраженные на естественном языке.
- Полуформальные онтологии на упрощенном естественном языке. Они описываются на ограниченном по структуре и словарю естественном языке, что значительно уменьшает многозначность определений, свойственную естественному языку.
- Полуформальные онтологии на искусственном языке. Для описания онтологии используется искусственный, формально определенный язык, в

результате чего онтология может использоваться в работе программных систем.

- Формальные онтологии. Описываются на формальном языке с явно определенным синтаксисом и семантикой и обладающим свойствами непротиворечивости и полноты. В качестве формальных способов записи онтологии используются логические языки.

На рисунке 1.14 [5, 159] представлен спектр онтологий в зависимости от степени детализации при их реализации.

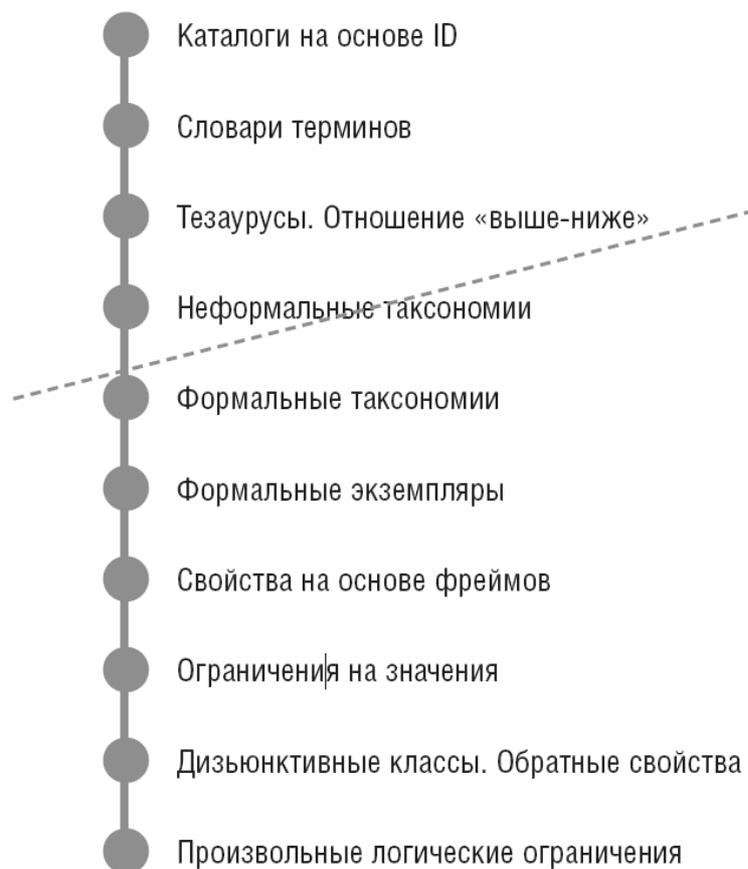


Рисунок 1.14 – Спектр онтологий

Каждая точка соответствует наличию некоторых ключевых структур в онтологии, отличающих ее от других. Косая черта условно отделяет онтологии от других ресурсов, имеющих онтологический характер, т.е. отделены «машинопонятные» и «человекопонятные» описания.

Первой точкой на спектре соответствует контролируемый словарь, т.е. конечный список терминов (простейшим примером является каталог на основе идентификаторов).

Словари терминов (глоссарии) дают грамматическую и стилистическую характеристику терминов, примеры употребления и другие сведения, непригодны для автоматической обработки программными агентами.

Тезаурусы – нормативные словари ключевых слов и дескрипторов (словарных единиц в виде слов, словосочетаний или кодов, называющих класс условной эквивалентности, в который включены эквивалентные и близкие по смыслу ключевые слова). Тезаурусы несут дополнительную семантику, определяя связи между терминами. Отношения, свойственные для тезаурусов: синонимия, иерархическое отношение и ассоциация. Обычно тезаурусы в явном виде не имеют иерархии терминов, но она может быть восстановлена.

Формальные таксономии – это онтологии, включающие точное определение отношения «класс-подкласс». Здесь соблюдается отношение транзитивности: если В – подкласс класса А, то каждый подкласс класса В тоже является подклассом класса А. В неформальных таксономиях это отношение выполняется не всегда.

Не все онтологии содержат только имена классов. Многие содержат на нижнем уровне экземпляры (индивиды) классов. Такого рода онтологии относятся к онтологиям с формальными экземплярами.

Далее для соответствующих онтологических структур увеличивается выразительность, т.е. увеличивается степень детальности описания вводимых в онтологии понятий. Для каждого понятия в таких онтологиях дано определение в терминах других понятий. Для каждого отношения задана область определения и область значения, дополнительно могут быть заданы правила использования отношения.

Онтологии часто классифицируют по *языку представления онтологических знаний* [27, 28, 83]:

- RDF. Язык разработан в рамках проекта Семантик-веб (Semantic Web). Основное предназначение языка – описание метаданных документов, размещаемых в Интернете. RDF использует базовую модель представления данных «объект – атрибут – значение».
- DAML+OIL – семантический язык разметки Web-ресурсов, который расширяет стандарты RDF и RDF Schema за счет более полных примитивов моделирования. В последнюю версию DAML+OIL включен набор дополнительных конструкций для создания онтологий и разметки информации в легкоинтерпретируемом машиной виде.
- OWL (Web Ontology Language) – язык представления онтологий следующего поколения после DAML+OIL. Обладает более богатым набором возможностей, чем XML, RDF, RDF Schema и DAML+OIL.
- KIF (Knowledge Interchange Format) – основан на S-выражениях синтаксиса для логики. KIF – специальный язык, предназначенный для обмена знаниями между разными компьютерными системами. Разрабатывался для описания общего формата представления знаний, независимого от конкретных систем.
- CycL (язык описания онтологии CYC) – это гибридный язык, в котором объединены свойства фреймов и логики предикатов. Синтаксис языка CycL схож с синтаксисом языка Lisp. CycL различает такие сущности, как экземпляры, классы, предикаты и функции.
- OCML (Operational Conceptual Modeling Language) язык поддерживает построение нескольких типов конструкций представления знаний.
- LOOM и PowerLoom® – языки представления знаний, разработанные исследователями из группы Artificial Intelligence Research Group Университета Южной Калифорнии (University of Southern California's Information Sciences Institute). Цель проекта Loom – разработка и внедрение

продвинутых средств для представления знаний и рассуждений в области искусственного интеллекта.

- Ontolingua предоставляет распределенную среду для совместного просмотра, создания, редактирования, изменения и использования онтологий. Ontolingua состоит из KIF-парсера, инструментов для анализа онтологии, и набора трансляторов для преобразования исходных данных Ontolingua в форму, приемлемую для внедрения в системы представления знаний.
- F-Logic – онтологический язык, который базируется на логиках первого порядка, однако классы и свойства в нем представлены как термины, а не как предикаты. Язык создавался для осуществления взаимодействия между онтологиями, построенными на основе предикатов, и онтологиями, построенными на основе F-Logic.

По *степени зависимости от конкретной задачи или прикладной области*, согласно классификации в [2, 27], выделяют следующие виды онтологий:

- Верхнего уровня. Такие онтологии описывают наиболее общие концепты (пространство, время, материя, объект, событие, действие и т. д.), которые независимы от конкретной проблемы или области. Примером такой общей онтологии является коммерческий проект онтологии СУС. Это база знаний, содержащая все общие понятия окружающего мира. Её могут использовать самые разные программные средства.
- Ориентированные на предметную область. Во многих дисциплинах сейчас разрабатываются стандартные онтологии, которые могут использоваться экспертами по предметным областям (доменам) для совместного использования и аннотирования информации в своей области. Например, в области медицины созданы большие стандартные, структурированные словари, такие как SNOMED и семантическая сеть

Системы Унифицированного Медицинского Языка (The Unified Medical Language System).

- Ориентированные на задачу. Это онтология, используемая конкретной прикладной программой и содержащая термины, которые используются при разработке ПС, выполняющего конкретную задачу. Она отражает специфику приложения, но может также содержать некоторые общие термины (например, в графическом редакторе будут и специфические термины — палитра, тип заливки, наложение слоев и т. д., и общие — сохранить и загрузить файл). Онтологии предметных областей и онтологии задач описывают, соответственно, словари, которые относятся к определенной предметной области (например, медицина, дистанционное обучение, Интернет-технологии) или типичной задаче (например диагностика, продажа). При этом они используют специализацию терминов, представленных в онтологиях верхнего уровня.
- Прикладные онтологии описывают концепты, которые зависят как от онтологии задач, так и от онтологии домена. Примером может служить онтология для автомобилей, строительных материалов, вычислительной техники.

Формальная модель онтологии

Формальная модель онтологии определяется как упорядоченная тройка конечных множеств [14, 80]:

$$O = \langle C, R, A \rangle, \quad (3)$$

где C – совокупность концептов предметной области, которую описывает онтология O ; R – множество отношений между концептами; A – совокупность аксиом (законов и правил, которые описывают принципы существования концептов).

Такая модель является тяжеловесной или «весомой». Легковесная модель или «легкая» онтология определяется как:

$$O = \langle C, R \rangle. \quad (4)$$

Около 80 % разработанных онтологий относятся к «легким» [14].

Все современные онтологии (независимо от спецификации языка и его программной реализации) строятся примерно одинаково. Основными компонентами онтологий являются [2, 162]:

- концепты (иногда их называют классы, понятия, сущности, категории);
- свойства концептов (иногда их называют слоты, атрибуты, роли);
- отношения;
- ограничения (иногда их называют фацеты, ограничения ролей).

Концепт – это шаблон, содержащий множество правил, определяющих форму экземпляра, т.е. то, каким образом может быть построен экземпляр. Экземпляр концепта служит для представления элемента предметной области.

Концепты могут иметь свойства (атрибуты) – имена или структуры полей записи и характеризуют размер или тип информации, содержащейся в поле. Концепты используются для хранения информации об экземпляре концепта. Значение атрибута может быть как сложным, так и простым типом данных.

Отношениями называются зависимости между концептами (экземплярами) онтологий. Обычно отношением является атрибут, ссылающийся на другой экземпляр.

Свойства концептов могут иметь различные ограничения, которые описывают тип значения, разрешенные значения, число значений (мощность) и т.д.

Построение онтологии

Онтологический инжиниринг подразумевает глубокий структурный анализ предметной области. Простейший алгоритм онтологического инжиниринга состоит из следующей последовательности шагов [15, 25]:

- выделение концептов — базовых понятий данной предметной области;

- определение «высоты дерева онтологий» — количества уровней абстракции;
- распределение концептов по уровням;
- построение связей между концептами — определение отношений и взаимодействий базовых понятий;
- консультации с различными специалистами для исключения противоречий и неточностей.

Другими словами, практическая разработка онтологии включает [27]:

- определение классов в онтологии;
- расположение классов в иерархию;
- определение слотов и описание допускаемых значений этих слотов;
- создание экземпляров и заполнение значений слотов.

При разработке онтологий придерживаются следующих фундаментальных правил [162]:

- не существует единственно правильного способа моделирования предметной области — всегда существуют жизнеспособные альтернативы. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений.
- Разработка онтологии — это обязательно итеративный процесс.
- Понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в исследуемой предметной области.

Для машинной обработки онтологических моделей используются формализованные способы их представления, которые позволяют организовать их эффективное хранение и работу с ними. В настоящее время можно выделить два основных формализма для представления онтологии: дескриптивные логики и реляционные модели [57, 80, 111, 113]. В данном исследовании при создании онтологической базы знаний использована реляционная модель, так

как реализация процесса формализации знаний, содержащихся в документации на ПС, через дескриптивные логики потребовала бы затрат, существенно превышающих затраты на разработку самих ПС.

Онтологии в системах управления знаниями

Удобство и эффективность СУЗ предприятий зависят не только от заложенных в её ядро принципов и предоставляемых инструментов, но также в значительной степени от используемой модели предметной области. Как уже было отмечено, в настоящее время одной из главных парадигм формирования концептуальных представлений слабоструктурированных предметных областей являются онтологии.

Онтологии играют решающую роль в модели представления знания предприятия. Основным преимуществом онтологического инжиниринга в СУЗ является целостный подход к автоматизации предприятия [15, 25, 113]. При этом достигаются:

- системность — онтология представляет целостный взгляд на предметную область;
- единообразие — материал, представленный в единой форме, лучше воспринимается и воспроизводится;
- научность — построение онтологии позволяет восстановить недостающие логические связи во всей их полноте.

Онтология является целью и средством формирования СУЗ [17]. Важность онтологического инжиниринга в СУЗ обусловлена также тем, что знание, которое не описано и не тиражировано, становится устаревшим и бесполезным. Напротив, знание, которое распространяется, приобретает и обменивается, генерирует новое знание.

1.6. Постановка цели и задач исследования

Анализ базовых процессов ЖЦ ПС, стандартов и публикаций в области управления качеством разработки ПС, управления знаниями позволяет сделать следующие выводы:

1. Знания предприятия извлекаются из ресурсов знаний. Ресурсы знаний предприятия представляют собой связанное множество скрытых знаний сотрудников (групп) и объектов знаний, содержащих явные знания (текстовые документы, медиа-файлы, базы данных и т.д.).
2. Знания на современных предприятиях создаются и используются в ходе выполнения бизнес– и технологических процессов.
3. Внедрение процесса УЗ на предприятии является важным фактором достижения стратегических целей, таких, как повышение качества продукции, приобретение конкурентных преимуществ, уменьшение затрат, способствование инновациям, более эффективного принятия решений и т.п.
4. Проведен анализ основных стандартов в области разработки СМК, оценки качества процессов и уровней зрелости компаний, разрабатывающих ПС. Рассмотренные стандарты документально закрепляют использующиеся в организациях производственные знания и предлагают некоторые действия по управлению, однако сам процесс управления знаниями в явном виде не регламентируют.
5. Основным ресурсом знаний предприятия – разработчика ПО является техническая документация на ПС. В настоящее время существуют системы поддержки разработки документации на ПС, позволяющие автоматизировать процесс формирования комплектов документации. Однако такие системы не поддерживают решение задач УЗ, содержащихся в документации на ПС (например, структурирование знаний, поиск информации с учетом семантических свойств и связей).

6. УЗ выполняется с использованием как организационных, так и технологических средств. Технологической составляющей управления знаниями является СУЗ. Обоснована необходимость создания СУЗ на исследуемом предприятии.
7. Проведен анализ программных средств для СУЗ. Показано, что СУЗ не может быть внедрена в базовые процессы ЖЦ ПС на предприятии – разработчике ПО путем простой инсталляции готового решения, но может быть построена на базе существующей технологической платформы.
8. Обоснован выбор технологической платформы для создания прототипа СУЗ в виде программного продукта Microsoft SharePoint.
9. Формализация и представление знаний осуществляется с использованием специальных моделей представления знаний. Сравнительный анализ моделей представления знаний показал, что наиболее перспективной технологией разработки моделей знаний в базовых процессах ЖЦ ПС является онтологический инжиниринг.

На основе сделанных выводов определена следующая цель диссертационной работы.

Цель работы заключается в повышении качества базовых процессов ЖЦ ПС на основе разработки и применения методов и средств управления знаниями.

Для достижения указанной цели необходимо решить следующие основные задачи:

1. Разработка метода структуризации информации в документации на ПС на основе онтологических моделей представления знаний в базовых процессах ЖЦ ПС.
2. Разработка методики создания СУЗ о программной продукции (СУЗПП), обеспечивающей управление знаниями и повышение качества базовых процессов ЖЦ ПС.

3. Разработка модели и прототипа СУЗПП.
4. Разработка методики оценки качества базовых процессов ЖЦ ПС при внедрении СУЗПП.

2. НАУЧНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ ЗНАНИЯМИ В БАЗОВЫХ ПРОЦЕССАХ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СРЕДСТВ

2.1. Метод структуризации информации

Разработан метод структуризации информации, находящейся в документации на ПС, позволяющий выделить из документов концепты разрабатываемых ПС, свойства и связи между ними, структурировать их формальным способом в виде модулей документации (МД). В качестве рабочего определения понятия «модуль документации» в настоящей работе используется следующее:

Модуль документации – дискретный фрагмент текста документа в электронном формате, который может содержать как изображения, так и текст с форматированием и гиперссылками.

МД является неделимым с точки зрения хранения и использования фрагментом документации. МД оформляется как самостоятельный документ.

Каждый вид МД описывает соответствующий концепт (понятие) разрабатываемой информационной системы. Например, настроечный параметр, назначение продукта, назначение подсистемы, описание операции (последовательности ручных или автоматических действий), описание процедуры/функции, описание таблицы базы данных, описание визуальной формы (окна) приложения и т.д. Каждый конкретный МД – это описание соответствующего экземпляра понятия. МД является неделимым с точки зрения хранения и использования фрагментом документации на ПС. МД оформляется как самостоятельный документ (файл).

Инструментом создания/редактирования остается Microsoft Word. Формат хранения – Word XML Document (*.xml). Для хранения МД используется библиотека документов SharePoint.

Вместо разработки и поддержания в актуальном состоянии цельных документов документаторы создают или изменяют МД. В качестве примера МД можно привести описание настроечного параметра. Как уже было отмечено в разделе 1.3, настроечный параметр – настройка (переключатель), включающая/выключающая некоторую функциональность, изменяющая алгоритм функционирования информационной системы или разрешающая/запрещающая определенные действия. Описание настроечных параметров обычно создается в руководстве системного программиста на подсистему (рисунок 2.1).

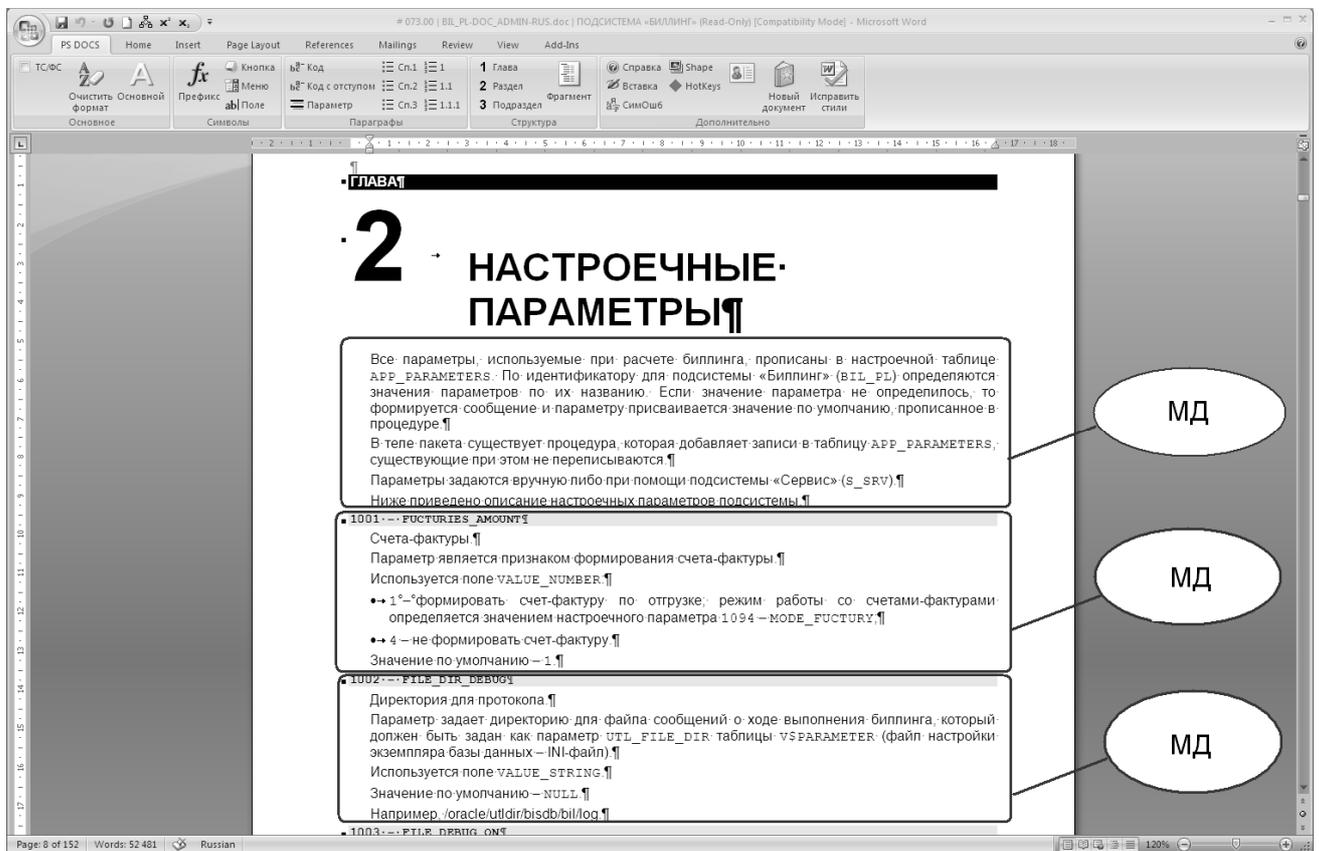


Рисунок 2.1 – Фрагмент документа с описанием настроечных параметров

Метод структуризации информации, находящейся в документации на ПС, состоит из следующих этапов:

1. Выявление видов МД, их свойств и связей.
2. Разработка правил идентификации, оформления и описания МД.

3. Перевод документации на модульную структуру.

С функциональной точки зрения основным понятием типовой информационной системы является функциональная возможность. Функциональные возможности (ФВ) – это возможности системы, которые позволяют заказчику реализовывать у себя на площадке соответствующие технологические и бизнес-процессы. Функциональные возможности описываются двумя видами МД: назначение ФВ (текст с ответом на вопрос «Зачем?») и реализация ФВ (текст с ответом на вопрос «Как?»).

У любой системы ФВ могут выстраиваться в многоуровневую систему иерархии. Коренные ФВ, как правило, концептуальны, а их реализация – это перечисление возможностей более низкого уровня, которые, в порядке выполнения могут быть как последовательными, так и параллельными.

В случае ФВ нижнего уровня, описание реализации ФВ представляет собой описание входа (условий для реализации), выхода (результата) и перечень действий пользователя (ручных операций) с упоминанием внутренних процессов (автоматических операций) в порядке, который приведет к реализации возможности. Функциональная возможность, её реализация и операции описываются соответствующими видами МД.

В общем случае информационная система состоит из подсистем, которые могут группироваться в программные продукты, группы продуктов и т.д. На нижнем уровне функциональную возможность может реализовывать одна подсистема, а также несколько подсистем могут реализовывать одну функциональную возможность в рамках продукта. ФВ верхнего уровня реализуются в рамках нескольких продуктов, групп продуктов.

Подсистемы состоят из элементов, описание таких элементов являются соответствующими видами МД. Часть МД участвует в операциях – наборе действий (например, процедура/функция) и настройке (например, настроенный параметр) в рамках одной подсистемы. Другая часть описывает саму подсистему, например, назначение подсистемы, комплектация и т.д.

Алгоритм выявления видов МД, их свойств и связей состоит из следующей последовательности шагов:

1. Формирование иерархии функциональных возможностей.
2. Формирование списка операций для каждой функциональной возможности.
3. Формирование перечня видов МД, участвующих в операциях.
4. Фиксация видов МД, их свойств и связей.

Выявления видов МД, их свойств и связей выполняется экспертами-документаторами с возможным привлечением для этого других экспертов в области соответствующего направления разработки ПС (аналитиков, разработчиков и т.д.).

Процесс создания компонентной документации может быть эффективен только при использовании на предприятии с централизованным администрированием ресурсов, когда структура и содержание всех разрабатываемых документов, в том числе и МД, определяются стандартами предприятия и тщательно тестируются.

Для МД вводится уникальная идентификация. Каждому виду МД соответствует трёхсимвольный идентификатор. Например, уникальный идентификатор для настроечного параметра состоит из следующих элементов:

- символьный код подсистемы;
- символьный код вида МД (для настроечного параметра – «STP»);
- символьное обозначение МД, уникальное в пределах вида;
- язык описания.

Пример: BIL_PL.STP.1001.RUS.xml. Пример МД для настроечного параметра приведен на рисунке 2.2.

Правила оформления регламентируют стили заголовков и текста МД. МД могут комбинироваться для создания различных документов и Web-представления, поэтому изложение текста в МД должно подразумевать возможность использования этого текста вне конкретного контекста – текста

модулей, расположенных перед и после рассматриваемого МД. Правила описания регламентируют содержание МД. Правила идентификации, оформления и описания МД разрабатываются редакторами совместно с инженером по знаниям.

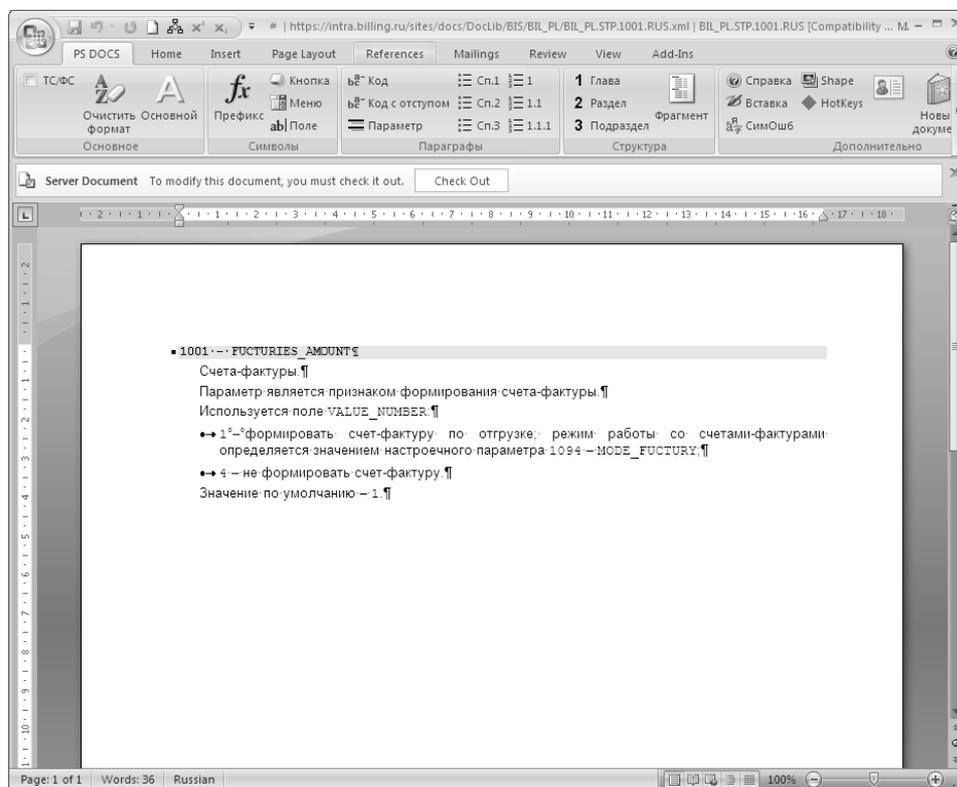


Рисунок 2.2 – Пример МД для настроечного параметра

Оформление и описание настроечного параметра, как и других МД, выполняется в соответствии со стандартами предприятия. Основными принципами оформления и описания настроечного параметра являются:

- Описание настроечных параметров в документации должно быть однотипным. Описание параметра начинается с маркера – номера настроечного параметра (для указываемых по номеру) и через дефис кода настроечного параметра. Маркер оформляется соответствующим стилем Microsoft Word. Маркер настроечного параметра приводится обязательно.
- На следующей строке приводится полное наименование параметра на языке, доступном пользователю. Наименование должно позволять его

однозначно идентифицировать. Полное наименование параметра приводится обязательно. Полное наименование параметра оформляется соответствующим стилем Microsoft Word.

- Далее следует подробное описание настроечного параметра. В подробном описании настроечного параметра поясняется механизм работы, влияния настроечного параметра. Подробное описание настроечного параметра приводится обязательно. Подробное описание настроечного параметра оформляется соответствующим стилем Microsoft Word.

После описания настроечного параметра обязательно перечисляются все возможные установки (значения) этого настроечного параметра и указывается значение по умолчанию. При необходимости приводится описание установки настроечного параметра, приводится пример заполнения.

Стили Microsoft Word используются для автоматического форматирования документа. При сохранении МД в формате Word XML название стиля обрамляет соответствующий текст в виде тега, который оформлен данным стилем.

Таким образом, обеспечивается возможность автоматического форматирования МД при помощи XSL-трансформации.

Перевод документации на модульную структуру выполняется документаторами в процессе выполнения рабочих заданий на создание или модификацию ПС. Хорошо структурированные документы легко разбиваются на МД соответствующего вида. Неструктурированные документы требуют предварительной переработки перед переводом на модульную структуру (создание недостающих описаний).

Очевидна невозможность мгновенного внедрения модульной структуры на всём массиве документации на ПС. Сначала МД включаются в цельные документы. Постепенно выделяются новые виды МД до полного перевода документации на модульную структуру. Вся новая документация на ПС сразу создается модульной. При этом часть МД может быть сгенерирована из

программного кода (например, таблица базы данных, процедура/функция) и при необходимости дополнена документатором.

Включение МД в цельные документы Microsoft Word производится при помощи поля IncludeText. Для вставки поля IncludeText следует выбрать пункт меню Insert -> Quick Parts -> Field -> IncludeText (рисунок 2.3). В поле «FileName or URL» указывается путь к МД в Share Point. Для преобразования Word XML в формат Word необходимо выставить флажок «Text converter», в качестве значения указать «XML». Для обеспечения возможности форматирования МД при обновлении его в документе Microsoft Word необходимо убрать флажок «Preserve formatting during updates».

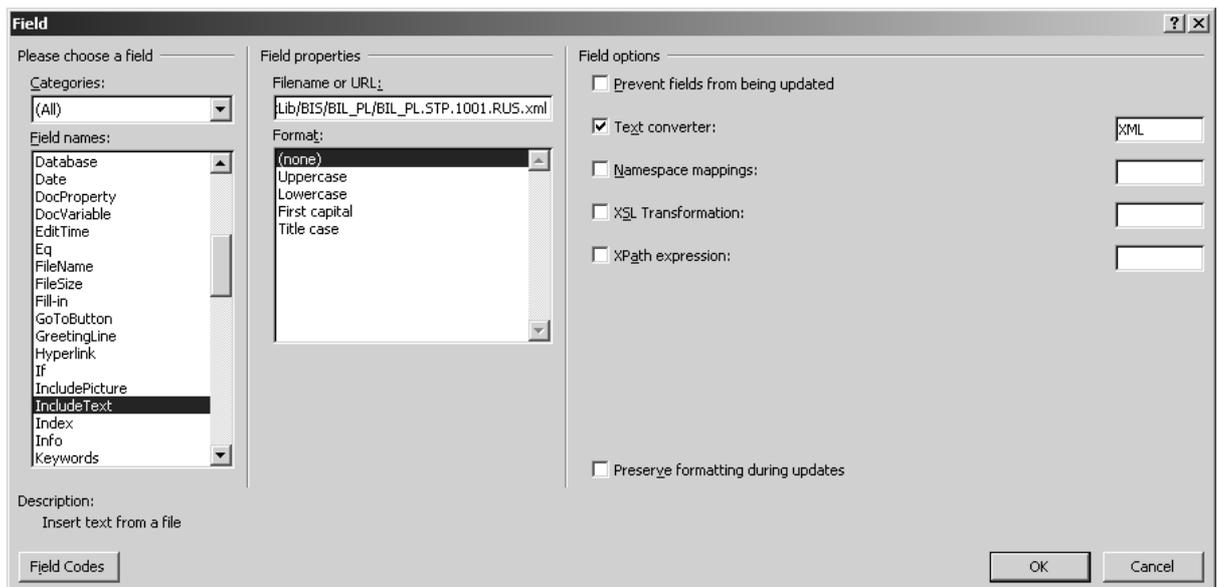


Рисунок 2.3 – Вставка МД в документ

2.2. Комплекс онтологических моделей представления знаний

На рисунке 2.4 представлен фрагмент онтологии информационной системы для настроечного параметра. Основными понятиями данного фрагмента являются: настроечный параметр, функциональность и подсистема, которые являются концептами (классами) онтологии.

Концепт «Настроечный параметр» имеет следующие атрибуты (в скобках указаны тип значения и разрешенные значения атрибута):

- название (строка);
- путь к МД (строка);
- подсистема (экземпляр класса «Подсистема»);
- функциональность (экземпляр класса «Функциональность»);
- язык (строка; возможные значения: русский, английский и т.д.).

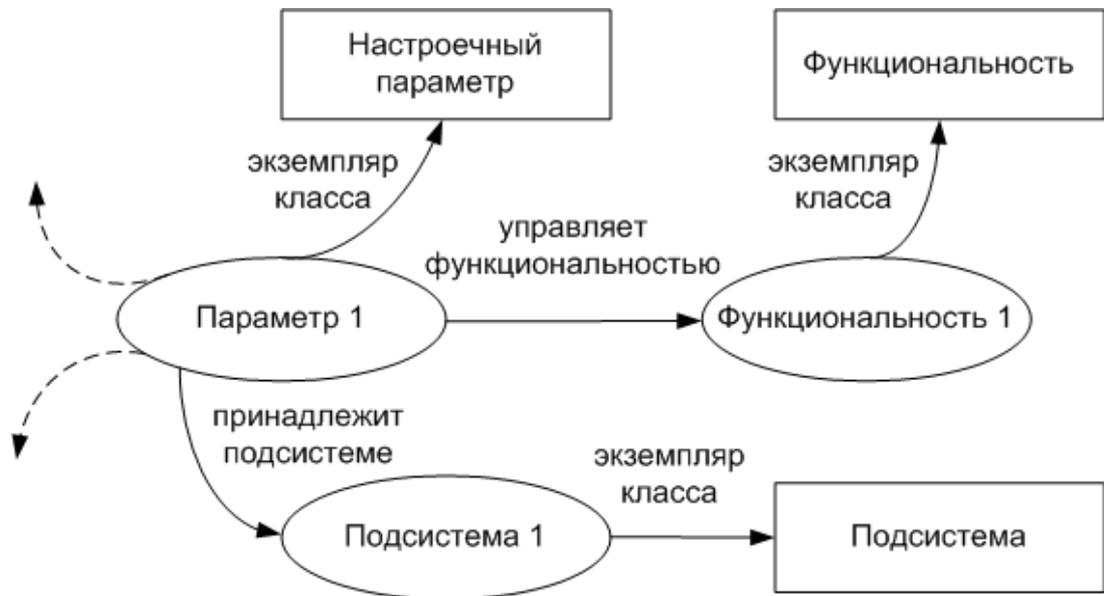


Рисунок 2.4 – Фрагмент онтологии для настроечного параметра

Как уже было отмечено в разделе 1.3, всю совокупность разрабатываемой на предприятии программной продукции можно рассматривать как основной компонент единой информационной системы, в которой информационные процессы и методы работы с информацией осуществляются с применением средств вычислительной техники и средств телекоммуникаций. Онтология информационной системы – формализованное представление такой системы, которое включает описание понятий системы (концептов) и логические связи между ними (отношения). Онтология вместе с набором экземпляров образует базу знаний информационной системы.

При таком подходе онтология может быть представлена в виде следующего кортежа:

$$O = \langle C, R \rangle, \quad (5)$$

где C – совокупность концептов информационной системы; R – совокупность отношений между ними.

Онтология информационной системы содержит три дополняющие друг друга составляющие, которые образуют комплекс онтологических моделей представления знаний в базовых процессах ЖЦ ПС [43]:

$$O = \langle O_D, O_\Phi, O_\Pi \rangle, \quad (6)$$

где O_D – онтология документации на ПС; O_Φ – онтология функциональностей информационной системы; O_Π – онтология продукции.

Онтология документации на программное обеспечение

В онтологию документации на ПС выделены концепты, относящиеся к документации на ПС (виды МД). Предложена следующая модель концепта онтологии документации на ПС:

$$C_D = \langle A_D, F_D, L_D \rangle, \quad (7)$$

где C_D – концепт онтологии документации на ПС; A_D – конечное множество атрибутов, описывающих свойства концептов C_D и отношения R_D между ними; F_D – множество ограничений на значения атрибутов; L_D – МД, связанные с концептом.

Для каждого концепта (класса) выделяется подмножество ключевых атрибутов, служащих для однозначной идентификации экземпляров.

В онтологии документации на ПС выделено около 100 концептов. Как уже было отмечено в подразделе 2.1, невозможно мгновенно создать все МД и определить для них атрибуты. На сегодняшний день создаются МД для следующих концептов (таблица 3).

Таблица 3 – Концепты онтологии документации на ПС

№ п/п	Наименование концепта (вида МД)	Описание	Код вида МД
1.	Настроечный параметр	Код, название и описание настроечного параметра	STP
2.	Назначение подсистемы	Текст описания без заголовка с наименованием	DSC
3.	Назначение продукта	Текст описания без заголовка с наименованием	DSP
4.	Назначение группы продуктов	Текст описания без заголовка с наименованием	DSG
5.	Назначение решения	Текст описания без заголовка с наименованием	DCD
6.	Назначение комплекса	Текст описания без заголовка с наименованием	DCK
7.	Назначение опции	Текст описания без заголовка с наименованием	DCO
8.	Процедура	Описание с заголовком: формат, входные, выходные параметры, алгоритм, сообщения и исключения	PRC
9.	Функция	Описание с заголовком: формат, входные, выходные параметры, алгоритм, сообщения и исключения	FNC
10.	Сообщение об ошибке	Код, текст сообщения и описание	ERR
11.	MIB-файл подсистемы	Описание MIB-файла подсистемы: SNMP-переменные и таблицы	MIB
12.	Роль	Название и описание роли	ROL
13.	Возможность	Описание функциональной возможности с заголовком	FTR
14.	Реализация возможности	Описание реализации функциональной возможности с заголовком	FTD
15.	Описание операции	Описание операции с заголовком	OPR
16.	Краткое описание автономной формы	Описание назначения автономной формы	FRK
17.	Описание автономной формы	Описание внешнего вида автономной формы	FRU
18.	Настройки автономной формы	Описание действий по установке (регистрации) автономной формы, а также настроек, которые необходимо выполнить, чтобы обеспечить работу с автономной формой	FRA
19.	Интеграция автономной формы	Информация, необходимая при разработке интегратора автономных форм: описание интерфейса вызова автономной формы (входные и выходные параметры), точки завершения работы автономной формы	FRA
20.	Не определен	Присваивается МД по умолчанию	-

В онтологии документации на ПС выделены следующие типы отношений:

- концептуальные (отношения между концептами):
 - «принадлежит подсистеме» – отношение принадлежности вида МД к подсистеме;
 - «принадлежит продукту» – отношение принадлежности вида МД к продукту;
 - «принадлежит группе продуктов» – отношение принадлежности вида МД к группе продуктов;
 - «принадлежит решению» – отношение принадлежности вида МД к решению;
 - «принадлежит комплексу» – отношение принадлежности вида МД к функциональному комплексу;
 - «управляет функциональностью» – отношение между видом МД и функциональной возможностью;
- ассоциативное отношение экземпляра с другим экземпляром. Ассоциативным отношением также считается гиперссылка из текста экземпляра МД на другой экземпляр МД.

Например, настроечный параметр 1001 принадлежит подсистеме «Биллинг» (BILL_PL) и управляет функциональностью формирования счетов-фактур. Параметр 1001 связан с параметром 1094 (рисунок 2.5).

<p>1001 – FUCTURIES_AMOUNT</p> <p>Счета-фактуры.</p> <p>Параметр является признаком формирования счета-фактуры.</p> <p>Используется поле VALUE_NUMBER:</p> <ul style="list-style-type: none"> • 1 – формировать счет-фактуру по отгрузке; режим работы со счетами-фактурами определяется значением настроечного параметра 1094 – MODE_FUCTURY; • 4 – не формировать счет-фактуру. <p>Значение по умолчанию – 1.</p>
--

Рисунок 2.5 – Описание настроечного параметра 1001

На рисунке 2.6 представлен фрагмент онтологии документации на ПС для настроечного параметра 1001.

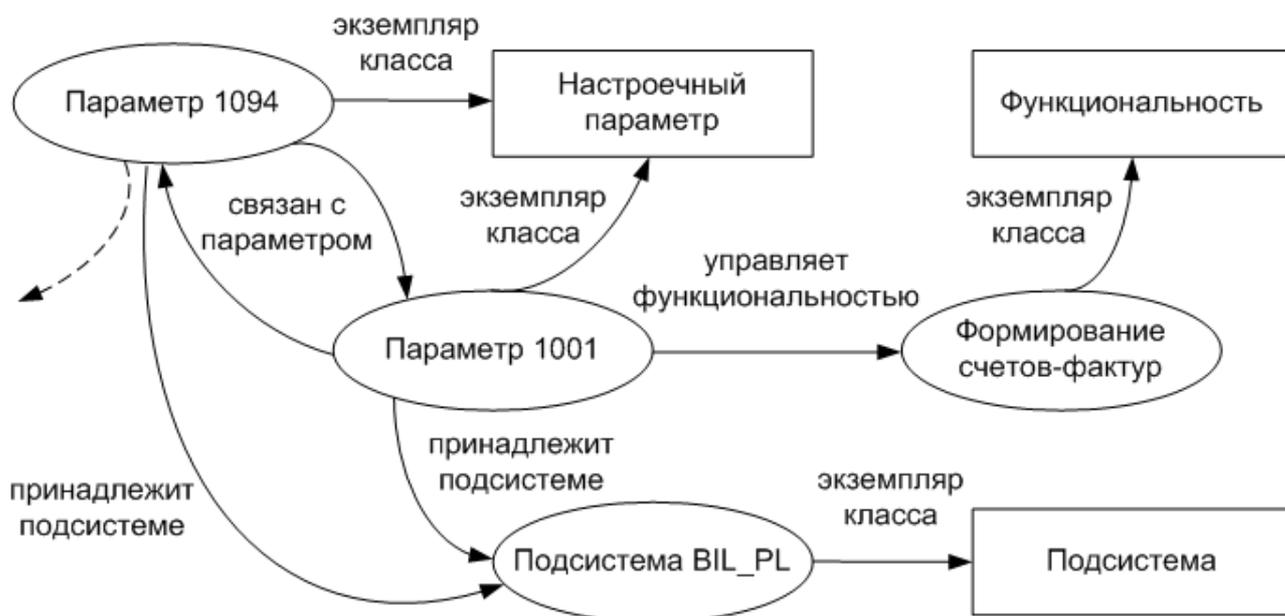


Рисунок 2.6 – Фрагмент онтологии для настроечного параметра 1001

Онтология документации на ПС реализована на основе библиотеки документов Microsoft SharePoint. Для каждого концепта (вида МД) онтологии документации на ПС создан тип контента (Content Type) со своим набором столбцов (атрибутов).

Тип контента в SharePoint представляет собой повторно используемую коллекцию параметров, которые можно применить к определенной категории контента. Типы контента обеспечивают централизованное управление метаданными и поведением типов документов, элементов и папок с возможностью повторного использования параметров.

Для использования типов контента в библиотеке документов необходимо выполнить следующие действия:

- создать тип контента;
- включить функцию «Типы контента» в библиотеке документов;

- добавить свой настраиваемый тип контента в библиотеку документов.

В Microsoft SharePoint используется несколько встроенных типов контента. Все из них, кроме одного, являются производными от других встроенных типов контента и их можно расположить иерархически на основе родительских отношений. Вверху иерархии размещен системный тип контента. За ним следует элемент, который образован от системного типа и от которого образованы все другие типы контента [6]. Пользовательские типы контента создаются на основе встроенных типов контента. Для всех типов контента онтологии документации на ПС создан родительский тип «Модуль документации».

В типе контента можно указывать столбец сайта или списка. Столбец представляет управляемый пользователем атрибут или фрагмент метаданных для элементов в списке или типов контента, в которые добавлен столбец.

Тип контента «Модуль документации» имеет следующие столбцы (атрибуты):

- Name – полный путь и наименование файла МД; тип данных – файл (File); родительский тип – Document;
- Content Type – тип контента;
- Title – название МД; тип данных – строка (Single line of text); родительский тип – Item; формируется автоматически с помощью пользовательского процесса (Workflow) из значения атрибута Name без расширения файла; используется в механизме поиска SharePoint;
- Язык – язык локализации МД; тип данных – выбор (Choice), возможные значения: русский, английский и испанский; родительский тип – нет;
- Шифр (техническое наименование) – уникальный атрибут МД; тип данных – строка (Single line of text); родительский тип – нет;

- Функциональная возможность – наименование функциональной возможности из иерархии управляемых метаданных; тип данных – управляемые метаданные (Managed Metadata); родительский тип – нет;
- Связь с другим МД – ссылка на связанный МД; тип данных – многострочный текст (Multiple lines of text); родительский тип – нет.

Тип контента «Настроечный параметр» (вида МД с описанием настроенных параметров) имеет все атрибуты типа контента «Модуль документации», а также собственный атрибут «Подсистема» – код подсистемы из списка «Подсистемы»; тип данных – внешние данные (External Data); родительский тип – нет. В атрибуте «Шифр» задается номер настроечного параметра.

Значения атрибутов «Подсистема», «Функциональная возможность» и «Связь с другим МД» для конкретного экземпляра O_D являются отношениями данного экземпляра с другими экземплярами O_D , O_Φ и O_Π .

Описание остальных концептов (типов контента) онтологии документации на ПС приведено в Приложении А.

В Microsoft SharePoint для каждого экземпляра библиотеки документов по умолчанию определяются следующие основные стандартные атрибуты:

- ID – идентификатор документа; тип данных – Long Integer;
- Created – дата и время создания документа; тип данных – Date/Time;
- Created By – пользователь, создавший документ; тип данных – Person or Group;
- Modified – дата и время последнего изменения документа; тип данных – Date/Time;
- Modified By – пользователь, изменивший документ; тип данных – Person or Group;
- File Type – тип документа; тип данных – Number;
- File Size – размер документа; тип данных – Int64;

- **Checked Out To** – пользователь, работающий с документом; тип данных – **Person or Group**;
- **Check In Comment** – комментарий к версии документа; тип данных – многострочный текст (**Multiple lines of text**);
- **Version** – версия документа; тип данных – **Microsoft.SharePoint.SPFileVersionCollection**.

Для создания отдельного экземпляра онтологии документации на ПС требуется выполнить следующие действия:

- 1) создать МД в редакторе Microsoft Word;
- 2) загрузить МД в библиотеку документов;
- 3) выбрать для МД соответствующий тип контента (**Content Type**);
- 4) определить значения атрибутов (рисунок 2.7).

API_CLNT_BASE.STP.210396.RUS.XML

Edit

Save Cancel Paste Copy Delete Item

Commit Clipboard Actions

Content Type Настроечный параметр

Name * API_CLNT_BASE.STP.210396.RUS.XML

Язык RUS ENU ES

Шифр 210396

Подсистема API_CLNT_BASE

Функциональная возможность Проведение биллинга за указываемый период;

Связь с другим МД docs/DocLib/BIS/API_CLNT_BASE/API_CLNT_BASE.STP.210111.RUS.XML;

Связь МД с другим МД

Version: 4.0
Created from e-mail at 5/27/2011 10:30 AM by Sher, Elena
Last modified at 6/6/2011 5:42 PM by Dmitriev, Pavel

Save Cancel

Рисунок 2.7 – Создание экземпляра онтологии документации на ПС

МД загружается в библиотеку документов при помощи интерфейса загрузки или электронной почты (e-mail). Также МД может быть создан на основе другого МД и сохранен под своим уникальным именем.

Для загрузки документов через e-mail для библиотеки документов должен быть задан адрес электронной почты и настроены правила загрузки. МД присоединяются к письму в виде вложения.

В онтологии документации на ПС на сегодняшний день создано около 52000 экземпляров (рисунок 2.8). Объем документации на ПС, переведенный на компонентную структуру, составляет примерно 60 % (более 23000 страниц текстовой информации).

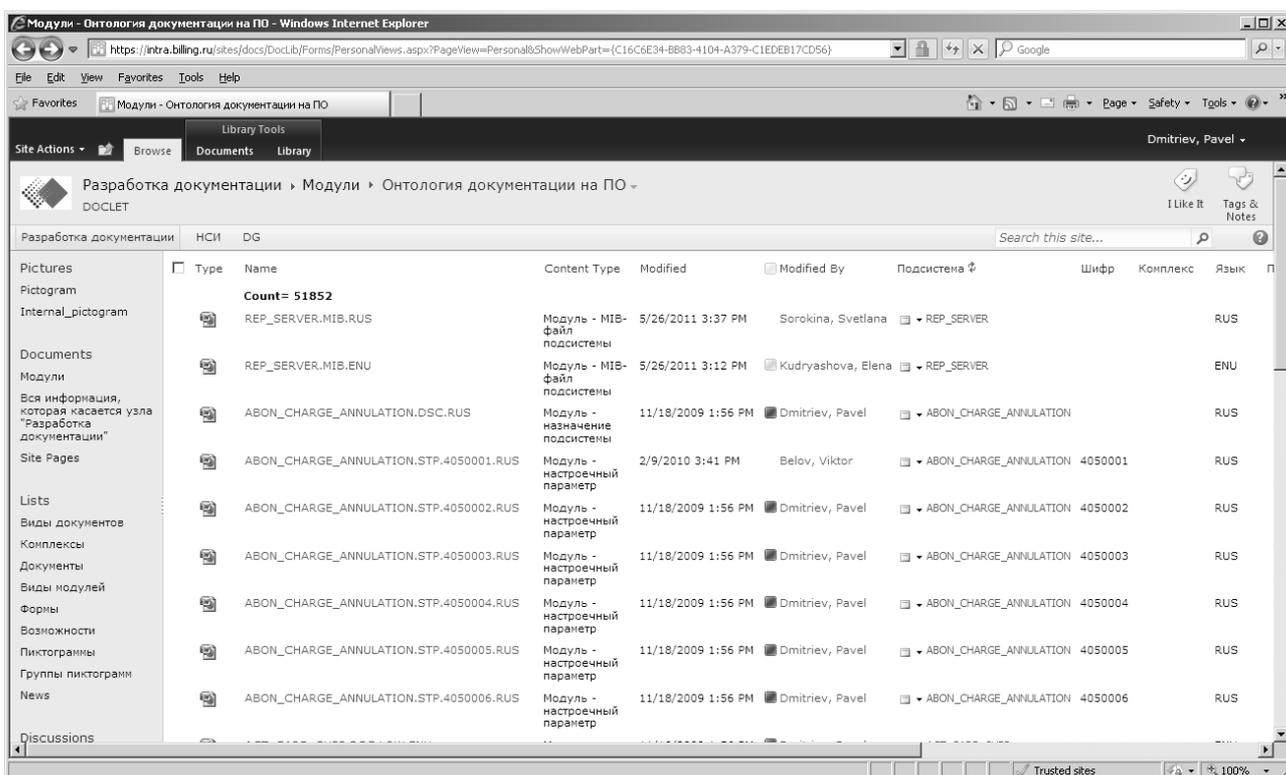


Рисунок 2.8 – Онтология документации на ПС

Онтология функциональностей информационной системы

Понятия онтологии, предназначенные для поддержки решения задач информационного поиска, должны быть связаны со значениями терминов

предметной области. Такого рода онтологию принято называть лингвистической или словарной онтологией [104].

В основе онтологии функциональностей предметной области лежит карта бизнес-процессов модели eTOM (enhanced Telecom Operations Map), разработанной международной некоммерческой организацией TeleManagement Forum (TM Forum) [42]. Организация TM Forum занимается вопросами развития и оптимизации бизнеса операторов связи, а также стандартизацией управления сетями связи и построения системы поддержки операционной деятельности следующего поколения (NGOSS) [126, 129, 132].

Концепция NGOSS, разработанная TM Forum, состоит из следующих составляющих (рисунок 2.9):

- eTOM – расширенная карта процессов деятельности телекоммуникационной компании – это эталонная модель, или архитектура бизнес-процессов, предназначенная для поставщиков услуг связи, а также их партнеров, работающих в телекоммуникационной отрасли. eTOM представляет собой набор процессов-элементов (шагов бизнес-процессов, или операций), организованных в виде иерархической структуры.
- TNA (Technology Neutral Architecture) – технологически нейтральная архитектура NGOSS. Является поставщиком структур и конструкций, необходимых для поддержки процессов анализа, проектирования, разработки и развертывания программных решений. Основная задача технологии – разделить бизнес-логику и компоненты реализации. Для возможности повторного использования бизнес-логики и независимости бизнес-процессов от конкретных программных решений.
- SID (Shared Information and Data Model) – общекорпоративная модель данных, предоставляющая телекоммуникационной индустрии единый набор определений бизнес-объектов участвующих в работе ключевых бизнес-процессов, а также, связи и отношения между этими объектами.

- TAM (Telecom Application Map) – карта приложений телекоммуникационной отрасли. Предоставляет взгляд на карту телекоммуникационных приложений в рамках таких ключевых сущностей NGOSS как eTOM и SID. TAM дополняет картину созданную в рамках TNA и позволяет отобразить составные части процессов на конкретные приложения независимых разработчиков.

Методология eTOM ориентирована на бизнес-процессы операторов услуг связи, описание связей и интерфейсов между этими процессами, на организацию совместного использования информации о заказчиках, предоставляемых услугах, имеющихся ресурсах, поставщиках/партнерах и другой информации в рамках многочисленных процессов. Для разработчиков информационной систем (например, систем класса OSS/BSS) модель eTOM очерчивает потенциальные границы программных компонентов, соответствующие требованиям заказчиков, и выделяет необходимые функции, входные и выходные данные, которые должны поддерживаться программными продуктами [100, 103, 121].

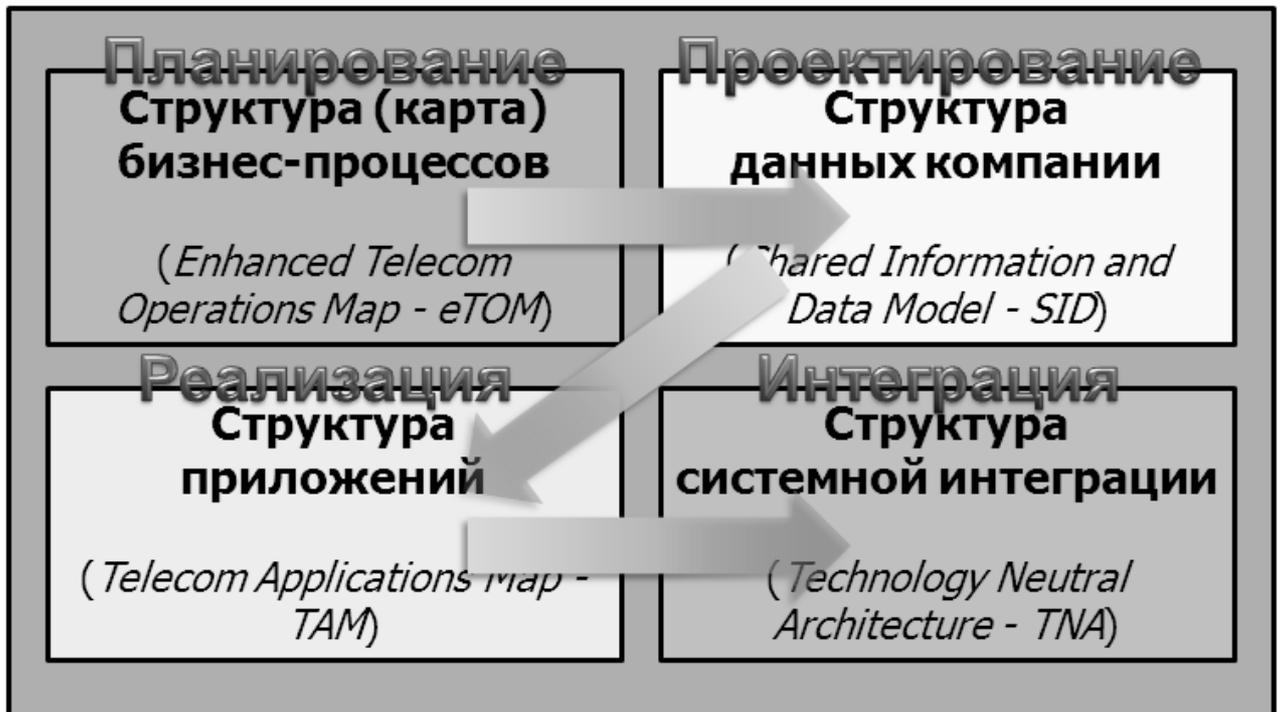


Рисунок 2.9 – Структура концепции NGOSS

Основными функциями eТОМ являются:

- 1) служить стандартной архитектурой бизнес-процессов отрасли связи;
- 2) задавать единый понятийный аппарат для описания элементов процессов поставщика телекоммуникационных услуг;
- 3) определять основные элементы данных, необходимые для выполнения каждого базового бизнес-процесса в рамках некоторой бизнес-операции, с их последующим использованием на уровне всей компании при разработке бизнес-требований и информационной модели для реализации интерфейсов, моделирования элементов совместно используемых данных, создания систем управления и поддержки бизнеса;
- 4) способствовать выявлению процессов и интерфейсов, в наибольшей степени нуждающихся в интеграции и автоматизации и зависящих от единых отраслевых стандартов.

Важными особенностями архитектуры eТОМ являются:

- eТОМ – это эталонная архитектура, учитывающая бизнес-процессы, возможные в деятельности телекоммуникационной компании;
- при разработке eТОМ акцент был сделан на связях между процессами, определении интерфейсов между ними и совместном использовании разными бизнес-процессами информации о клиентах, услугах, ресурсах и т.д.;
- в eТОМ учтены взаимодействия с внешней средой: клиентами, партнерами, поставщиками, регулирующими органами и др.;
- универсальность и открытость: eТОМ применима к любым сетевым технологиям, услугам и типам организации бизнеса компании;
- интеграция с другими широко применяющимися моделями (ITIL и др.);

eТОМ постоянно совершенствуется, в ее основе лежит мировой опыт ведущих компаний отрасли. Согласно рекомендации Международного союза

электросвязи преимущества использования eТОМ в компании состоят в следующем [136]:

- предлагает стандартные структуру, терминологию и систематику для описания бизнес-процессов телекоммуникационной компании и их элементов;
- позволяет применить единый стандарт разработки бизнес-процессов во всех подразделениях компании;
- является основой для понимания и управления набором разнообразных приложений и информационных систем с точки зрения требований бизнес-процессов;
- обеспечивает системное и высококачественное описание целевых сквозных бизнес-процессов с возможностью оптимизации их стоимости и производительности, а также использования существующих процессов и информационных систем;
- в результате широкого применения на различных предприятиях телекоммуникационной отрасли упрощает внедрение готового типового ПО и делает его использование дешевле разработки ПО на заказ.

Декомпозиция процессов для архитектуры бизнес-процессов eТОМ начинается на уровне «Предприятие» и описывает бизнес-процессы в виде некоторого набора групп [101]. Для формирования структуры бизнес-процессов модель eТОМ использует иерархию, в соответствии с которой выполняется последовательная декомпозиция всех процессов предприятия. Определяются описания процессов, входные и выходные данные, а также другие основные элементы.

Важно отметить, что в данной работе иерархия бизнес-процессов рассматривается с точки зрения предприятия – разработчика ПО (поставщика решений) для операторов услуг связи.

Метод декомпозиции применяется при разбиении сложного процесса на составляющие его функции. Суть метода декомпозиции иллюстрируется на рисунке 2.10. Здесь показан некоторый типичный элемент процесса «Процесс X», который обеспечивает некоторую специфическую область функциональных возможностей, например, выставление счетов клиентам. При выполнении анализа охватываемые функции, другие характеристики, связанные с этим процессом, подразделяются на несколько элементов процесса более низкого уровня и т.д.

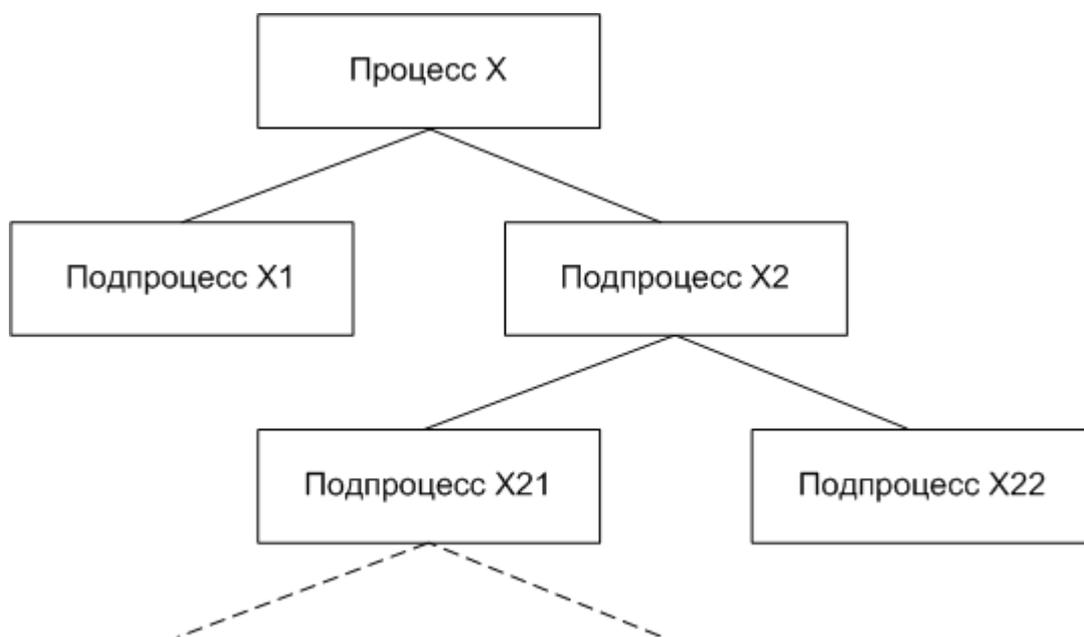


Рисунок 2.10 – Пример декомпозиции процесса

Схема eTOM уровня 2 приведена в Приложении Б.

Концепт C_ϕ онтологии функциональностей информационной системы рассматривается как пара: сигнатура (термин – функциональная возможность) из иерархии бизнес-процессов S и множество W ключевых слов, синонимов и сокращений к сигнатуре:

$$C_\phi = \langle S, W \rangle \quad (8)$$

Иерархия в онтологии функциональностей O_ϕ строится на основе отношения типа «класс-подкласс». Между функциональностями онтологией и

онтологией документации на ПС устанавливается прямая связь при помощи концептуального отношения «управляет функциональностью» – отношение между МД и функциональной возможностью (рисунок 2.11).

Онтология функциональностей реализована на основе набора терминов управляемых метаданных Microsoft SharePoint. Как уже было отмечено в разделе 1.4, управляемые метаданные – это иерархическая коллекция централизованно управляемых терминов. Термин – это слово или фраза, которую можно связать с элементом списка или библиотеки документов Microsoft SharePoint. Набор терминов – это коллекция связанных терминов.



Рисунок 2.11 – Связь онтологии функциональностей и онтологии документации на ПС

SharePoint Server включает средство управления банками терминов, с помощью которого можно создавать наборы терминов и управлять ими. При наличии необходимых разрешений с помощью средства управления банками терминов можно выполнять следующие действия:

- создание и удаление наборов терминов.
- Добавление, изменение и удаление терминов.
- Упорядочение управляемых терминов в наборе в виде иерархии.
- Задание синонимов и сокращений.
- Импорт терминов.
- Преобразование корпоративных ключевых слов в управляемые термины путем их перемещения в набор терминов.

При создании столбца управляемых метаданных в библиотеке документов необходимо указать набор терминов, из которого будут выбираться значения столбца.

Создание экземпляров онтологии функциональностей и формирование отношений между ними выполняется инженером по знаниям по запросу руководителя проектов (постановщика, аналитика). Наименование экземпляра функциональной возможности и её место в иерархии определяется на этапе эскизного или технического проектирования, фиксируется в проектной документации.

Фрагмент онтологии функциональностей для функциональной возможности «Взаиморасчеты» представлен на рисунке 2.12.

В онтологии функциональностей на сегодняшний день создано около 600 экземпляров.

Использование модели eTOM в онтологии функциональностей СУЗ позволяет сделать прозрачными бизнес-процессы, автоматизируемые предприятием – разработчиком ПО для телекоммуникационной отрасли, обеспечивает удобную навигацию и качественный поиск в онтологии документации на ПС.

Альтернативным или дополнительным вариантом построения онтологии функциональностей может являться карта приложений телекоммуникационной отрасли ТАМ. Схема ТАМ уровня 0 приведена в Приложении Б.

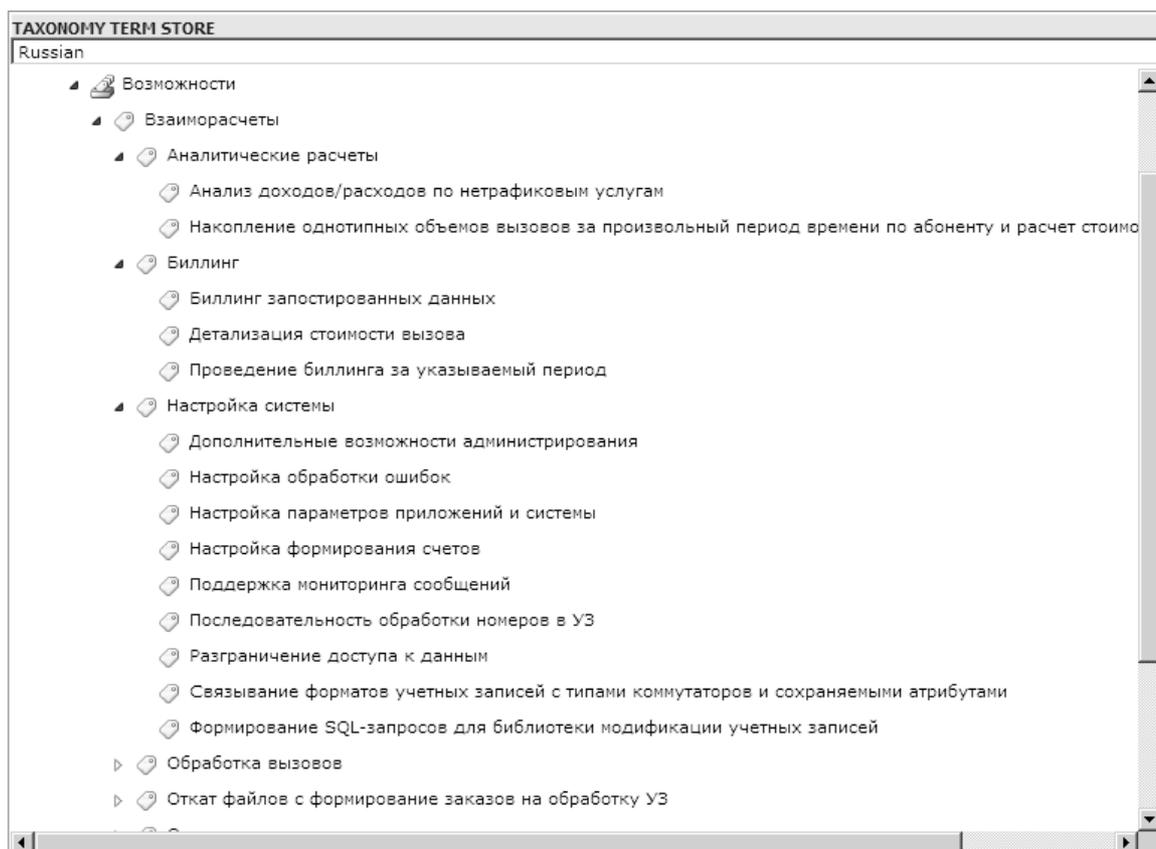


Рисунок 2.12 – Фрагмент онтологии функциональностей для функциональной возможности «Взаиморасчеты»

Онтология продукции

В онтологию продукции выделены концепты, относящиеся к программным продуктам (например, подсистема, продукт, группа продуктов и т.д.). Предложена следующая модель концепта онтологии продукции:

$$C_{П} = \langle A_{П}, F_{П} \rangle, \quad (9)$$

где $C_{П}$ – концепт онтологии продукции; $A_{П}$ – конечное множество атрибутов, описывающих свойства концептов $C_{П}$ и отношения $R_{П}$ между ними; $F_{П}$ – множество ограничений на значения атрибутов.

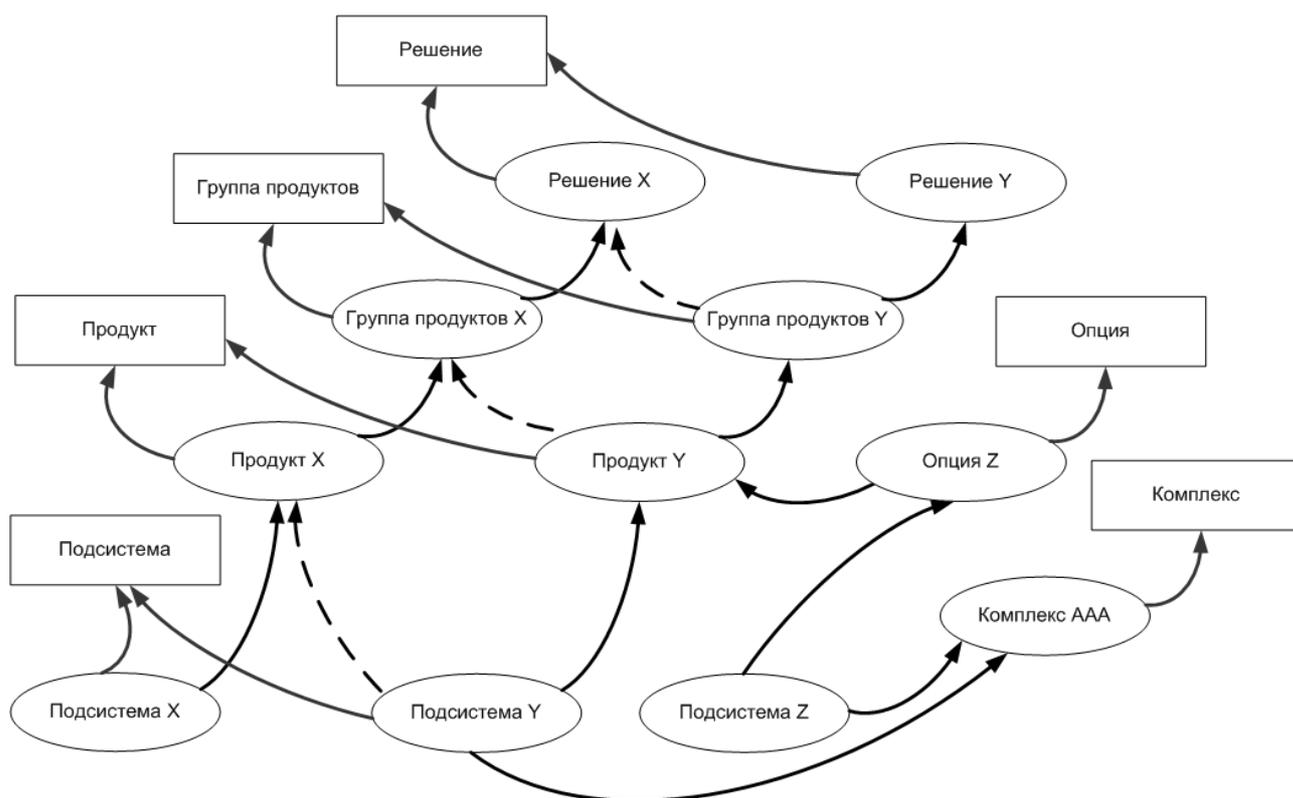
Создание экземпляров онтологии продукции и формирование отношений между ними выполняется менеджерами групп продуктов. Наименование экземпляра продукции и её место в иерархии определяется на этапе эскизного или технического проектирования, фиксируется в проектной документации.

В онтологии продукции выделено 6 концептов:

- «Подсистема» – элементарная единица учета выпускаемой продукции с точки зрения контроля версий и технологического процесса (постановка, разработка, тестирование, установка, эксплуатация, сопровождение).
- «Опция» – функциональность продукта, право использования которой предоставляется потребителям на основании отдельного соглашения, что технически реализуется с помощью регистрационного ключа, анализируемого программным обеспечением в процессе его работы.
- «Продукт» – совокупность взаимосвязанных подсистем, являющаяся единицей учета выпускаемой продукции с точки зрения поставки потребителям.
- «Группа продуктов» – совокупность взаимодополняющих продуктов, развиваемых на основе общего плана и архитектуры и предназначенных для автоматизации взаимосвязанных бизнес-процессов.
- «Решение» – совокупность взаимодополняющих групп продуктов, предназначенных для комплексной автоматизации потребителя.
- «Комплекс» – совокупность двух или более подсистем, совместно реализующих определенную функциональность.

Таксономия в онтологии продукции O_{II} строится на основе отношения типа «часть-целое» (включения) (рисунок 2.13). Прямоугольниками обозначены концепты, эллипсами – экземпляры, линии от экземпляров к концептам – отношения принадлежности экземпляров к концептам (классам), линии между экземплярами – отношения включения.

Одна подсистема может входить в состав нескольких продуктов или одну опцию. Опция входит в состав одного продукта. Один продукт может входить в состав нескольких групп продуктов. Группы продуктов могут объединяться в решение. Подсистемы могут объединяться в комплекс.

Рисунок 2.13 – Таксономия в O_{II}

Онтология продукции реализована на основе списков Microsoft SharePoint. Концепт «Подсистема» (список SharePoint «Подсистемы») имеет следующие атрибуты:

- **Наименование.** Указывается название подсистемы на русском языке; тип данных – строка (Single line of text).
- **Наименование английское.** Указывается название подсистемы на английском языке; тип данных – строка (Single line of text).
- **Код (техническое наименование).** Указывается аббревиатура, присвоенная подсистеме; тип данных – строка (Single line of text); количество символов – не более 20.
- **Регистрационный номер** – номер подсистемы, состоящий из пяти цифровых знаков; тип данных – строка (Single line of text); 00000÷99999;

- Владелец подсистемы. Указывается сотрудник ответственный за подсистему; тип данных – строка (Single line of text).
- Продукты. Продукты, в состав которых входит подсистема; тип данных – внешние данные (External Data).
- Опция. Опция, в состав которой входит подсистема; тип данных – внешние данные (External Data).
- Платформы. Указываются коды операционных систем, на платформе которых проводилось тестирование подсистем; тип данных – многострочный текст (Multiple lines of text).
- Тип подсистемы. Указывается тип подсистемы в соответствии с правилами классификации: универсальная, специализированная, регистрационный ключ; тип данных – выбор (Choice).
- Языки локализации. Указываются языки локализации документации; тип данных – выбор (Choice).
- Языки интерфейса. Отражается существование интерфейса пользователя на указанных языках; тип данных – выбор (Choice).
- Статус поддержки подсистемы. Указывается текущий статус технической поддержки подсистемы; тип данных – выбор (Choice); возможные значения: по умолчанию – «Да», при прекращении поставки и технической поддержки – «Нет», при частичном снятии с поддержки – «Частично».
- Примечание. При необходимости могут приводиться любые дополнительные сведения о подсистеме; тип данных – строка (Single line of text).

Описание структуры остальных концептов онтологии продукции (структуры списков) приведено в Приложении В.

Между онтологией документации на ПС и онтологией продукции устанавливаются прямые связи при помощи концептуальных отношений «принадлежит подсистеме», «принадлежит продукту» и т.д. (рисунок 2.14).

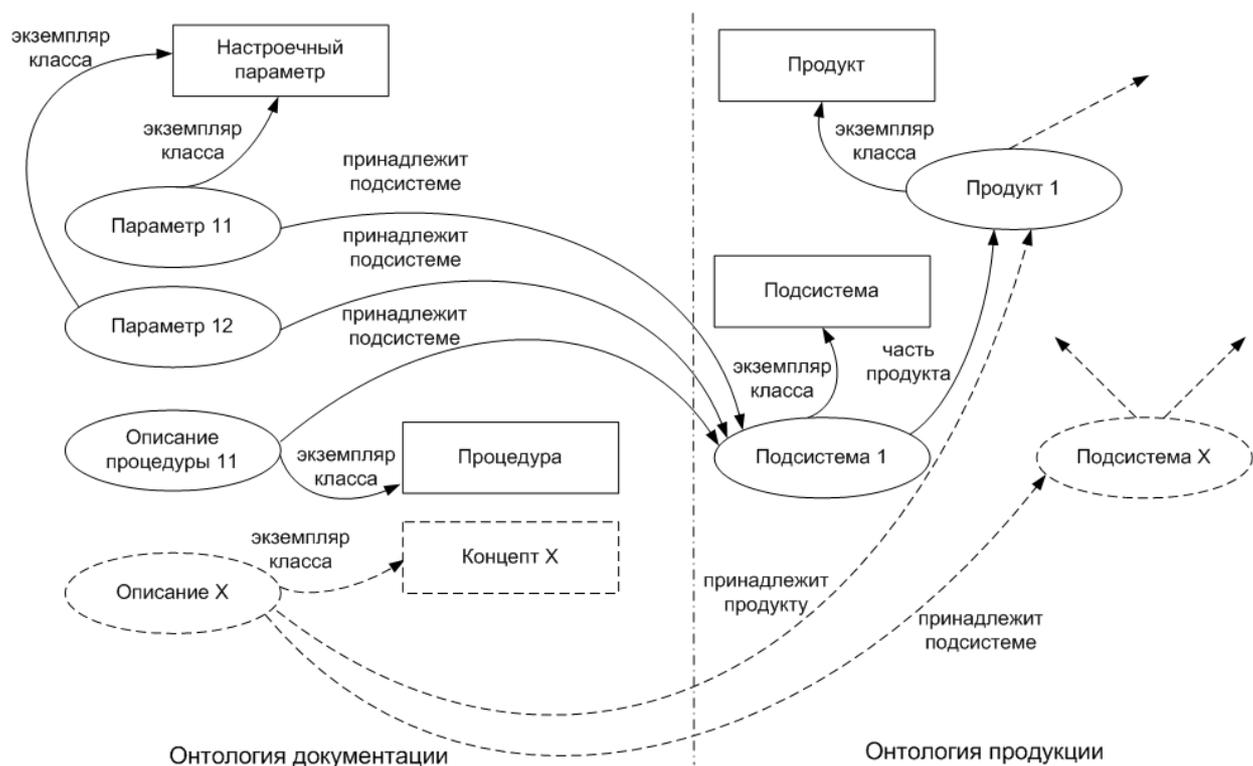


Рисунок 2.14 – Связь онтологии документации на ПС и онтологии продукции

Онтология продукции на сегодняшний день содержит около 5000 экземпляров (таблица 4).

Таблица 4 – Количество экземпляров в онтологии продукции

№ п/п	Наименование концепта	Количество экземпляров
1.	Подсистема	3461
2.	Опция	110
3.	Продукт	579
4.	Группа продуктов	23
5.	Решение	50
6.	Комплекс	752

Онтологии информационной системы являются динамическими. Функциональность продукта Microsoft SharePoint позволяет добавлять новые, удалять или изменять существующие типы контента и столбцы (атрибуты), в том числе и их типы данных.

2.3. Методика создания системы управления знаниями

Разработана следующая методика создания СУЗПП, состоящая из следующих этапов:

1. Предпроектный этап:

- обследование документации на ПС и других ресурсов знаний предприятия:
 - определение основных ресурсов знаний, содержащих явные знания: текстовые документы, базы данных, медиа-файлы и т.д.;
 - анализ процессов создания и использования ресурсов знаний;
 - анализ взаимосвязи ресурсов знаний на разных этапах ЖЦ ПС;
- формирование требований к СУЗПП;
- разработка технического задания.

2. Проектный этап:

- разработка онтологий информационной системы (выделение понятий, свойств и отношений между ними);
- наполнение онтологий:
 - создание экземпляров онтологий, заполнение свойств и формирование отношений между экземплярами;
 - перевод документации на ПС на модульную структуру,
- программная реализация (организация автоматизированной обработки и поиска информации по запросу, разработка шаблонов документов);
- разработка документации на СУЗПП;
- разработка документированных процедур и инструкций для пользователей СУЗПП на основе типового стандарта «Система управления знаниями о программной продукции» (Приложение Г) и предложений по изменению существующих стандартов организации в связи с внедрением СУЗПП.

3. Этап внедрения:

- проведение опытных испытаний;
- обучение персонала;
- ввод в действие стандартов;
- ввод системы в промышленную эксплуатацию.

4. Этап сопровождения (анализ функционирования, выявление проблем, внесение изменений в СУЗПП и стандарты организации).

Так как документаторы теперь не работают с полными текстами документов, а только с МД, дополнительно возникает задача формирования цельных динамических документов, когда, например, по какой-либо подсистеме вся документация переведена на компонентную структуру [41].

В качестве рабочего определения понятия «динамически формируемый документ» в настоящей работе используется следующее:

Динамически формируемый документ – шаблон, который определяет необходимый набор модулей документации.

Реализация данной задачи основана на принципе единого источника, рассмотренного в разделе 1.3.

Часть документов строится на основе predetermined шаблонов документов. Другая часть документов собирается на основе выборки, полученной пользователем на этапе поиска, либо непосредственно им указанной.

На рисунке 2.16 представлен пример схемы формирования динамического документа из экземпляров онтологии.

Рассмотрим пример формирования руководства системного программиста на подсистему и справочника настроечных параметров на группу продуктов на основе predetermined шаблонов документов.

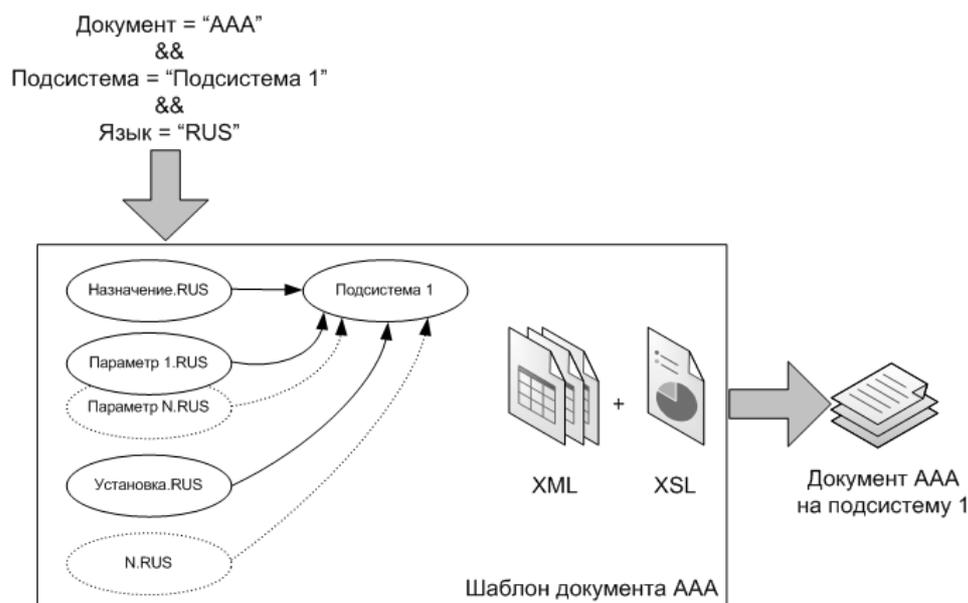


Рисунок 2.16 – Пример схемы формирования динамического документа

Входными данными для шаблона руководства системного программиста на подсистему является наименование подсистемы и язык документа. Данные атрибуты задаются для каждого экземпляра МД в онтологии для конкретной подсистемы. Алгоритм формирования руководства системного программиста на подсистему состоит из следующих этапов:

- получение МД по подсистеме и языку;
- вывод на экран МД в определенной последовательности и стилевом оформлении;
- сохранение документа.

Этапы вывода и сохранения могут быть объединены.

Входными данными для шаблона справочника настроечных параметров на группу продуктов является наименование группы продуктов и язык документа. Алгоритм формирования справочника настроечных параметров на продукт состоит из следующих этапов:

- определение совокупности наименований продуктов входящих в группу продуктов;
- определение совокупности наименований подсистем входящих в продукты;

- получение МД – настроечных параметров по подсистемам и языку;
- вывод на экран;
- сохранение документа.

2.4. Выводы к главе 2

1. Разработан метод структуризации информации, находящейся в документации на ПС, позволяющий выделить из документов концепты (понятия) разрабатываемых ПС, свойства и связи между ними, структурировать их формальным способом в виде МД.
2. Разработан комплекс онтологических моделей представления знаний в процессах определения требований, проектирования и разработки ПС: онтологии документации на ПС, онтологии функциональностей и онтологии продукции.
3. Онтология документации играет решающую роль в модели представления знаний информационной системы. Показано, что использование имеющихся знаний в онтологии документации позволяет автоматизировать выпуск комплектов документов без привлечения дополнительных ПС.
4. Разработана методика создания СУЗПП, позволяющая внедрить процесс управления знаниями в базовые процессы ЖЦ ПС для повышения их качества.

3. НАУЧНО-ТЕХНИЧЕСКИЕ ПРЕДЛОЖЕНИЯ ПО АВТОМАТИЗАЦИИ И СТАНДАРТИЗАЦИИ ПРОЦЕССА УПРАВЛЕНИЯ ЗНАНИЯМИ

3.1. Модель системы управления знаниями

Концепция разрабатываемой СУЗПП состоит в комбинации двух типов СУЗ (с использованием компонент информационно-поисковых систем и систем искусственного интеллекта). Ключевым компонентом СУЗПП является подсистема поиска, обеспечивающая оперативный отбор и выдачу релевантной информации по запросам. Поиск информации производится не в хранилище документов, а в онтологической базе знаний, что позволяет учесть семантику информации, находящейся в документации на ПС, взаимосвязи и взаимозависимости, обеспечить целостность и непротиворечивость знаний, повысить качество поиска. Используется комбинация синтаксического и семантического поиска, т.е. поиск выполняется в экземплярах онтологии с учетом их семантических свойств и связей.

Разработана следующая модель СУЗПП, представленная на рисунке 3.1.

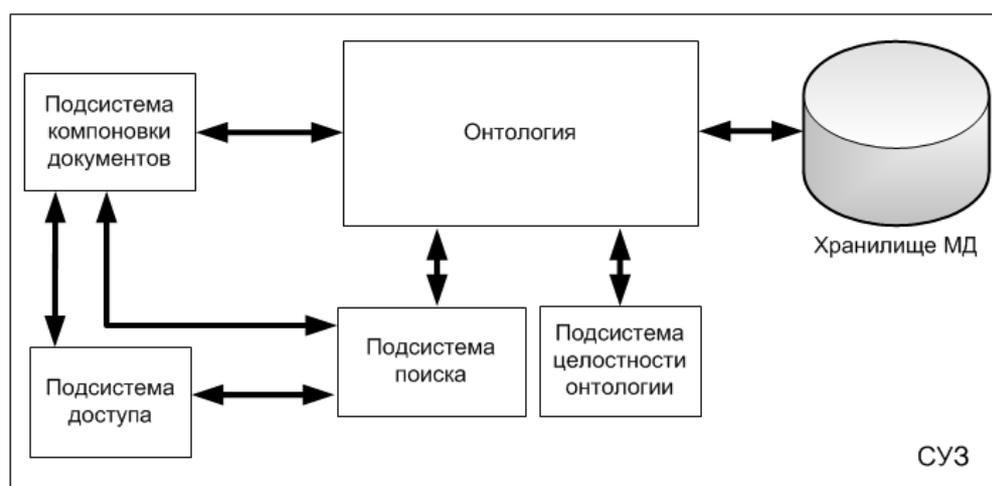


Рисунок 3.1 – Модель СУЗПП

Основными компонентами разрабатываемой СУЗПП являются [46]:

- **Онтология информационной системы.** Используется для описания информационных ресурсов с учетом семантики обрабатываемой информации. Онтология информационной системы разделена на составляющие: онтологию документации на ПС, онтологию функциональностей, онтологию продукции.
- **Подсистема поиска.** Выполняет поиск информации по запросам, навигацию, рубрицирование и визуализацию экземпляров онтологии информационной системы, обеспечивает обмен знаниями между пользователями СУЗПП при помощи механизмов комментирования и тегирования МД
- **Подсистема компоновки документов.** Выполняет формирование требуемых цельных документов в определенных форматах.
- **Подсистема разграничения доступа.** Обеспечивает разграничение доступа к информационным ресурсам.
- **Подсистема целостности онтологии.** Обеспечивает целостность онтологии.

Модель СУЗПП, участники процесса УЗ и их виды деятельности представлены на рисунке 3.2. Модель ЖЦ ПС после внедрения СУЗПП представлена на рисунке 3.3.

Прототип СУЗПП реализован на платформе продукта Microsoft SharePoint. СУЗПП представляет собой Web-портал и создается в пределах одного Web-сайта (узла) SharePoint. Архитектура разрабатываемого Web-портала представлена на рисунке 3.4.

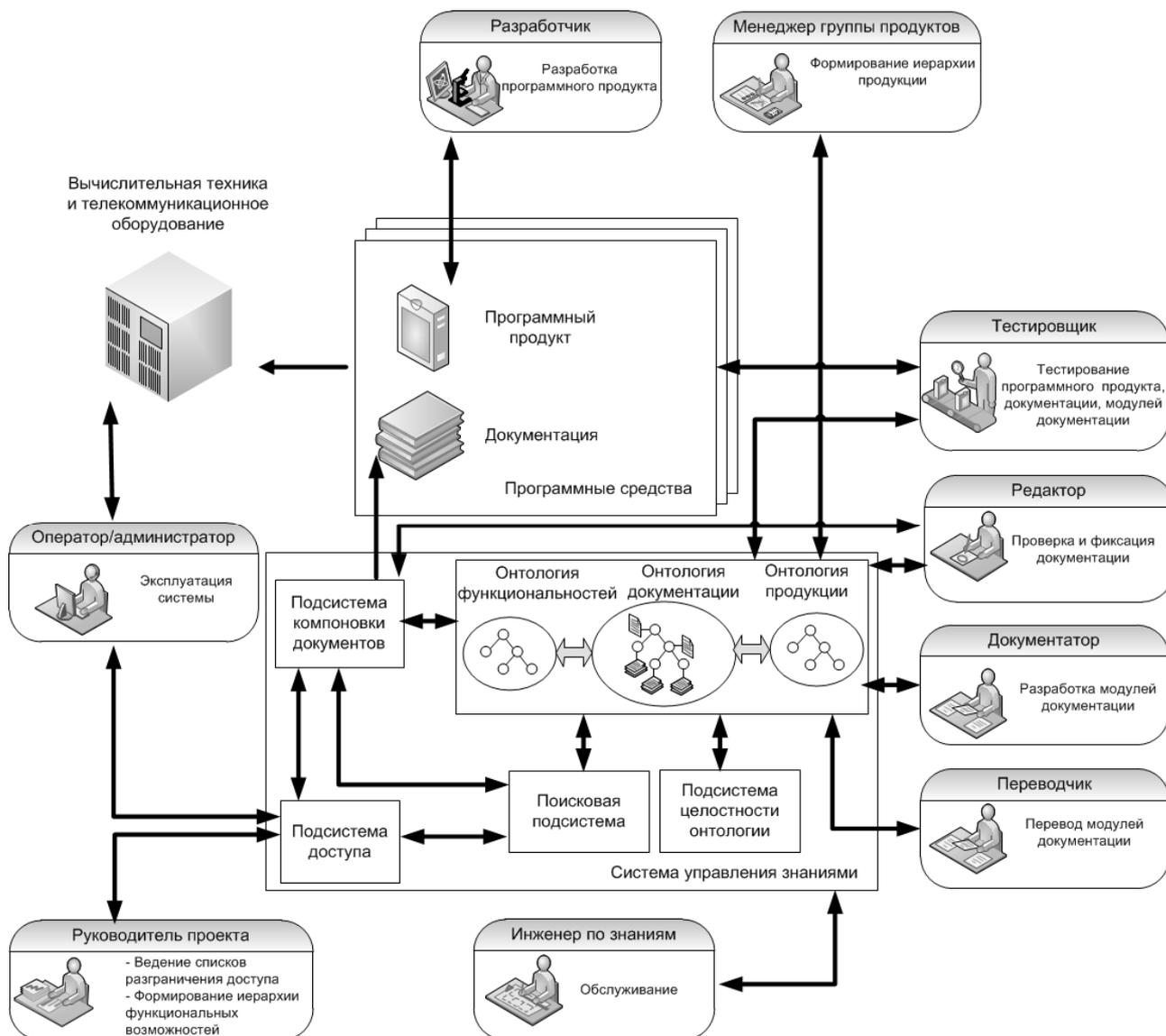


Рисунок 3.2 – Модель СУЗПП и участники процесса УЗ

Интерфейс СУЗПП для эксплуатирующего систему специалиста внешне напоминает интерфейс обычного Web-каталога (рисунок 3.5). В его основе – система рубрик, соответствующая иерархии онтологии функциональностей и онтологии продукции. Пользователь выбирает определённую рубрику и получает список МД, который ей соответствуют. Для наиболее полной и точной формализации информационной потребности пользователь может уточнить вид МД и указать дополнительные слова, которые должны присутствовать в требуемых МД или его атрибутах. При этом результатом

поиска являются МД, для которых дополнительно автоматически добавляются связи с другими МД.

Предприятие-разработчик ПО



Рисунок 3.3 – Модель ЖЦ ПС после внедрения СУЗПП

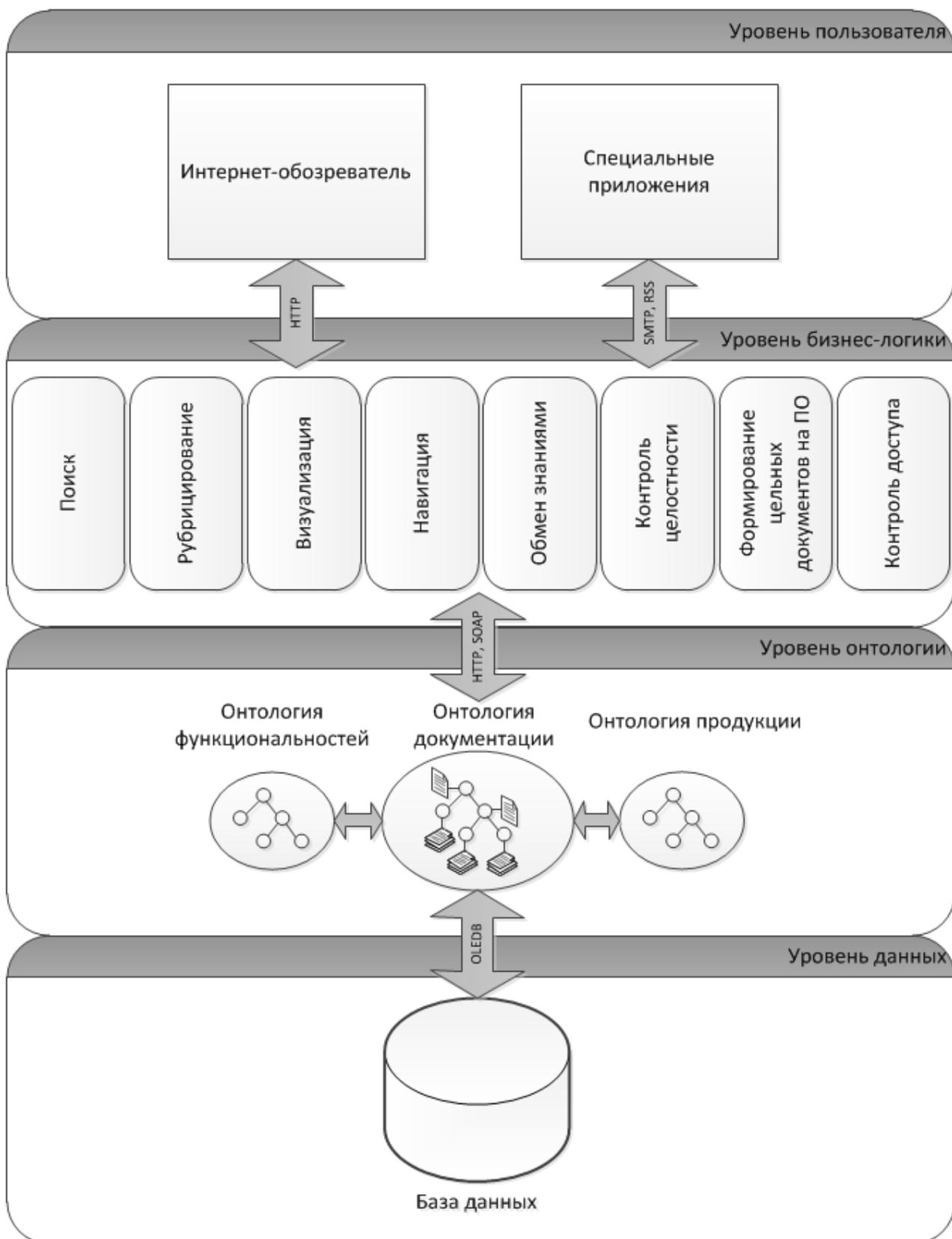


Рисунок 3.4 – Архитектура Web-портала

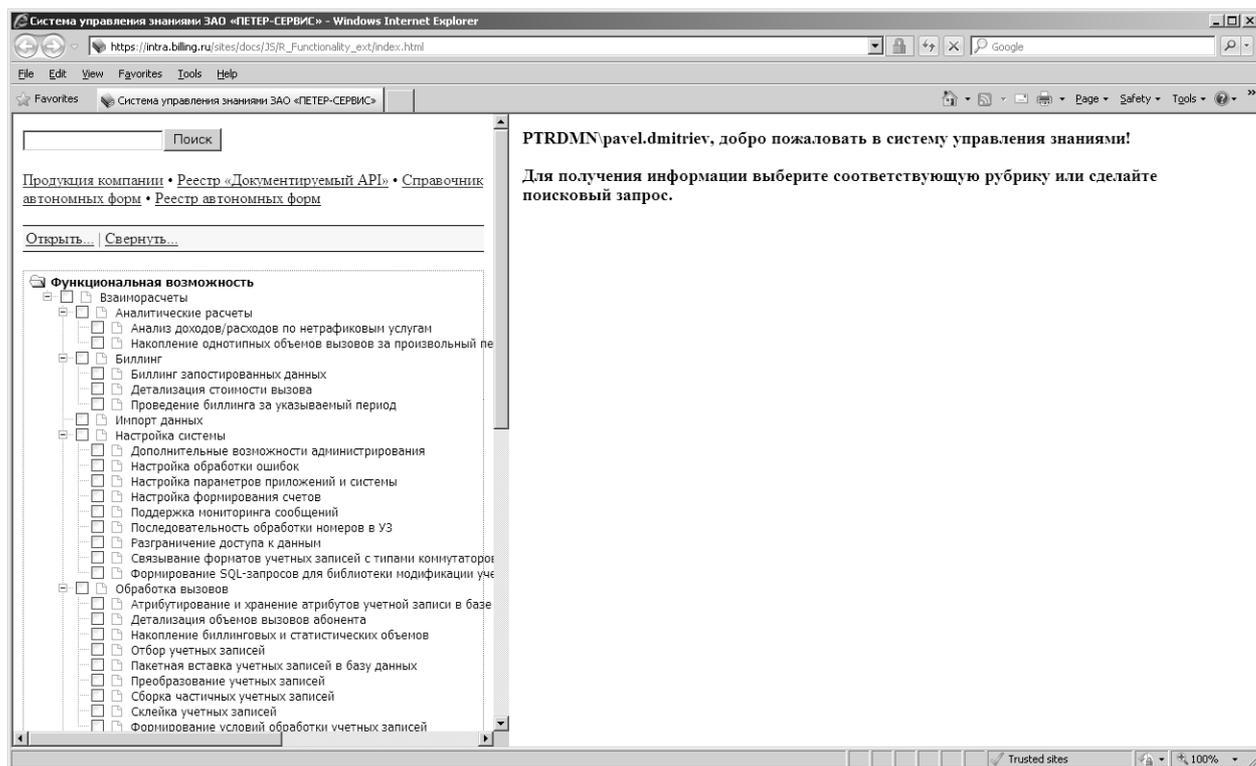


Рисунок 3.5 – Интерфейс СУЗПП

Таким образом, для решения задачи поиска информации соединены два подхода: использование поисковой системы, обеспечивающей релевантную отработку формального запроса, и интерфейса Web-каталога, обеспечивающего пользователю понятный и удобный интерфейс.

Пертинентность информационного поиска достигается за счёт спецификации не только определенной рубрики, в которой ищется информация, но и вида требуемого МД.

Разработан типовой стандарт организации «Система управления знаниями о программной продукции», регламентирующий правила работы с системой, роли пользователей системы и основные операции, которые доступны в рамках роли (Приложение Г).

3.2. Компоненты программно-алгоритмического обеспечения системы управления знаниями

3.2.1. Подсистема поиска

В состав компонентов алгоритмического обеспечения входят алгоритмы функционирования подсистем разрабатываемой СУЗПП. Компоненты программного обеспечения подсистем СУЗПП представляют собой совокупность программных средств, реализованных при помощи языков JavaScript, VBA, XML, XSLT, библиотек jQuery и AJAX, Web-служб Microsoft SharePoint, сценариев WSH.

Подсистема поиска предназначена для поиска информации по запросам, навигации, рубрицирования и визуализации экземпляров онтологии информационной системы. Дополнительно подсистема поиска обеспечивает обмен знаниями между пользователями СУЗПП при помощи механизмов комментирования и тегирования МД.

Подсистема поиска использует штатный механизм поиска, реализованный в продукте Microsoft SharePoint, рассмотренный в разделе 1.4.

Алгоритм поиска построен на основе двух подходов:

1. Поиска по ключевым словам по всем экземплярам онтологии документации на ПС с учетом их семантических свойств и связей;
2. Ограничения объема ресурсов при выборе определенной рубрики онтологии функциональностей или онтологии продукции и поиск в более низких рубриках.

При выполнении поискового запроса текст запроса, введенный в подсистеме поиска, передается в Web-службу поиска SharePoint. Далее запрос разбивается на слова и фразы, отделяемые пробелами. После этого получившиеся в результате слова и фразы обрабатываются парадигматическим модулем для создания зависящих от языка изменяемых грамматических форм конкретного слова. Использование средства разбиения на слова и

парадигматического модуля как при обходе контента, так и в процессах обработки запроса повышает эффективность поиска, так как получаются более релевантные результаты поиска. Свойства для соответствующих МД загружаются из хранилища свойств и проверяются на совпадение.

Результатом поиска является список всех совпадающих элементов, упорядоченный в порядке их релевантности словам запроса.

При выводе описания соответствующего МД поисковая подсистема дополнительно добавляет значения атрибутов МД и связи с другими МД.

Алгоритм поиска состоит из следующих этапов:

1. Выполнение поиска по ключевым словам по всем фрагментам онтологии документации с учетом их семантических свойств и связей.
2. Если задано ограничение по рубрикам, то выполняется фильтрация результатов поиска в соответствии с рубрикой, указанной пользователем.
3. Фильтрация результатов поиска в соответствии с языком интерфейса пользователя.
4. Вывод результатов поиска и видов МД.
5. Если пользователь указывает ограничение по виду МД, то выполняется фильтрация результатов поиска в соответствии с видом МД.

Предложенная структура программного обеспечения подсистемы поиска представлена на рисунке 3.6 и включает формы ввода поискового запроса и визуализации иерархии (онтологии функциональностей и онтологии продукции), модуль поиска, форму результатов поиска и формы вывода МД (для каждого вида или группы МД), значений атрибутов и связей, заданных для МД.

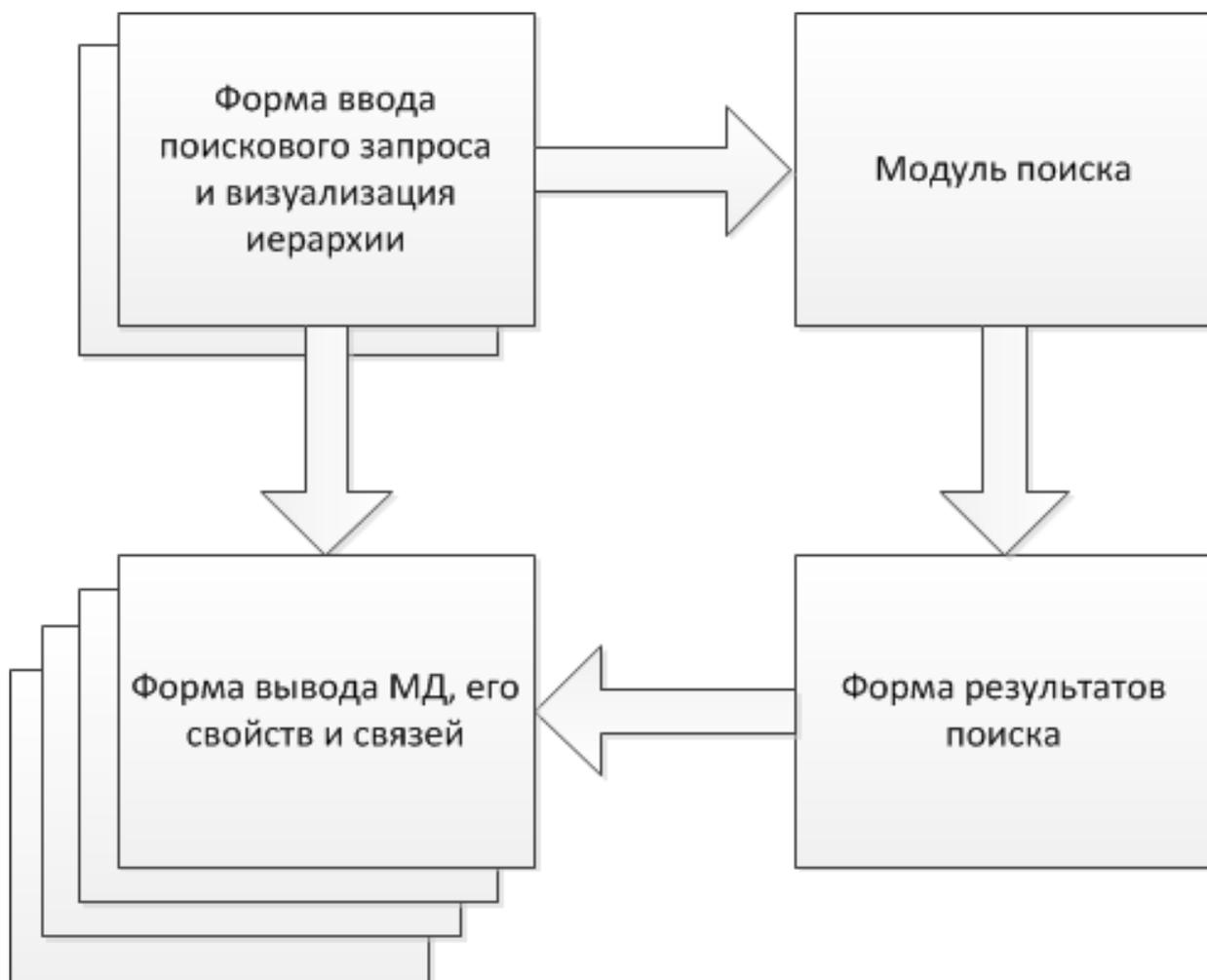


Рисунок 3.6 – Структура программного обеспечения подсистемы поиска

Форма ввода поискового запроса и визуализации иерархии предназначена для передачи поискового запроса и ограничения объема в модуль поиска, построения иерархии онтологии функциональностей или онтологии продукции, а также навигации по иерархии.

Для построения иерархии онтологии функциональностей используется обращение к Web-службе управляемых метаданных (TaxonomyClientService) при помощи методов `GetChildTermsInTermSet` и `GetChildTermsInTerm`. Метод `GetChildTermsInTermSet` возвращает все дочерние термины из набора терминов в виде строки XML-кода [171]. Метод `GetChildTermsInTermSet` возвращает все дочерние термины для заданного родительского термина в виде строки XML-кода [170]. Описание структуры XML-кода приведено в [161].

Для построения иерархии онтологии продукции используется обращение к Web-службе списков (Lists) при помощи метода `GetListItems`. Метод `GetListItems` возвращает все строки списка в соответствии с заданным условием в виде XML-кода [156].

Модуль поиска выполняет обращение к Web-службе поиска (Search) при помощи метода `Query` и разбор результатов поиска.

XML-формат для поискового запроса определяется в схеме `Microsoft.Search.Query`. Фактический поисковый запрос задается в элементе `QueryText` [11].

Метод `Query` возвращает результаты поиска в строке как XML-код. Формат для XML-кода определяется схемами `Microsoft.Search.Response` и `Microsoft.Search.Response.Document`. Полный набор результатов поиска содержится в элементе `Results` схемы `Microsoft.Search.Response`, с каждым отдельным результатом, представляемым элементом `Document` схемы `Microsoft.Search.Response.Document`.

Модуль поиска производит разбор результатов поиска с учетом заданных ограничений. Дополнительно для каждого результата поиска производится проверка доступа при помощи подсистемы разграничения доступа. Если у пользователя нет разрешения на просмотр соответствующего МД, поисковая подсистема отфильтровывает этот документ из возвращаемого списка.

Форма результатов поиска предназначена для вывода результатов поиска после их разбора в модуле поиска, а также для возможности фильтрации результатов по виду МД.

Форма вывода МД выполняет вывод текста самого МД, а также значений его атрибутов и связей с другими МД. Вывод МД выполняется при помощи XSLT-преобразования XML-документа МД. Для получения значений атрибутов и связей с другими МД используется обращение к Web-службе списков (Lists) при помощи метода `GetListItems`.

Дополнительно для каждого связанного МД производится проверка доступа при помощи подсистемы разграничения доступа. Если у пользователя нет разрешения на просмотр соответствующего МД, то он отсутствует в форме вывода.

Примеры реализации обращений к Web-службам управляемых метаданных, списков и поиска приведены в Приложении Д.

Реализация таблицы XSLT-стилей приведена в Приложении Е.

Работа подсистемы поиска показана на рисунке 3.7. В этой начальной форме отображается иерархия онтологии функциональностей (функциональных возможностей ПС) и поле ввода поискового запроса. Из данной формы можно перейти к дополнительным сервисам: форме с иерархией онтологии продукции и формам вывода подсистемы компоновки документов.

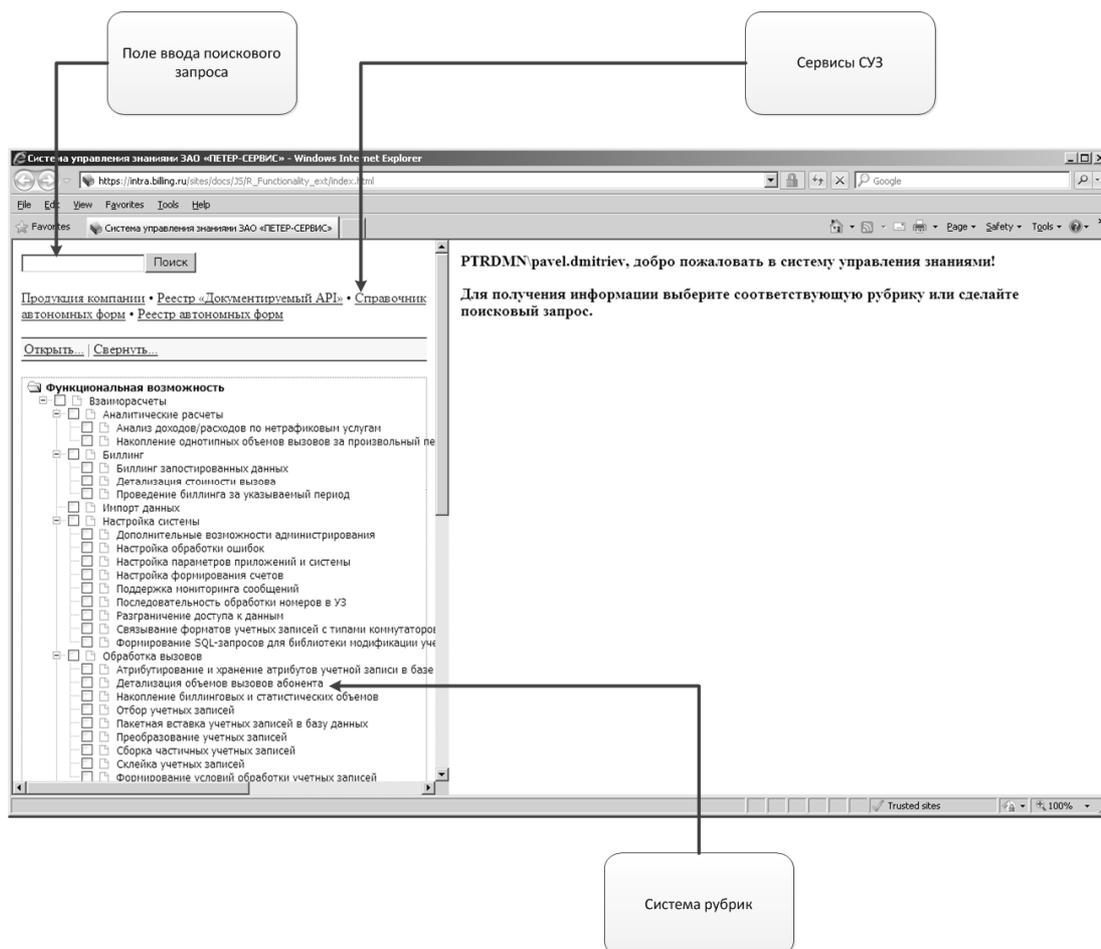


Рисунок 3.7 – Начальная форма подсистемы поиска

Форма с иерархией онтологии продукции имеет аналогичный внешний вид (рисунок 3.8).

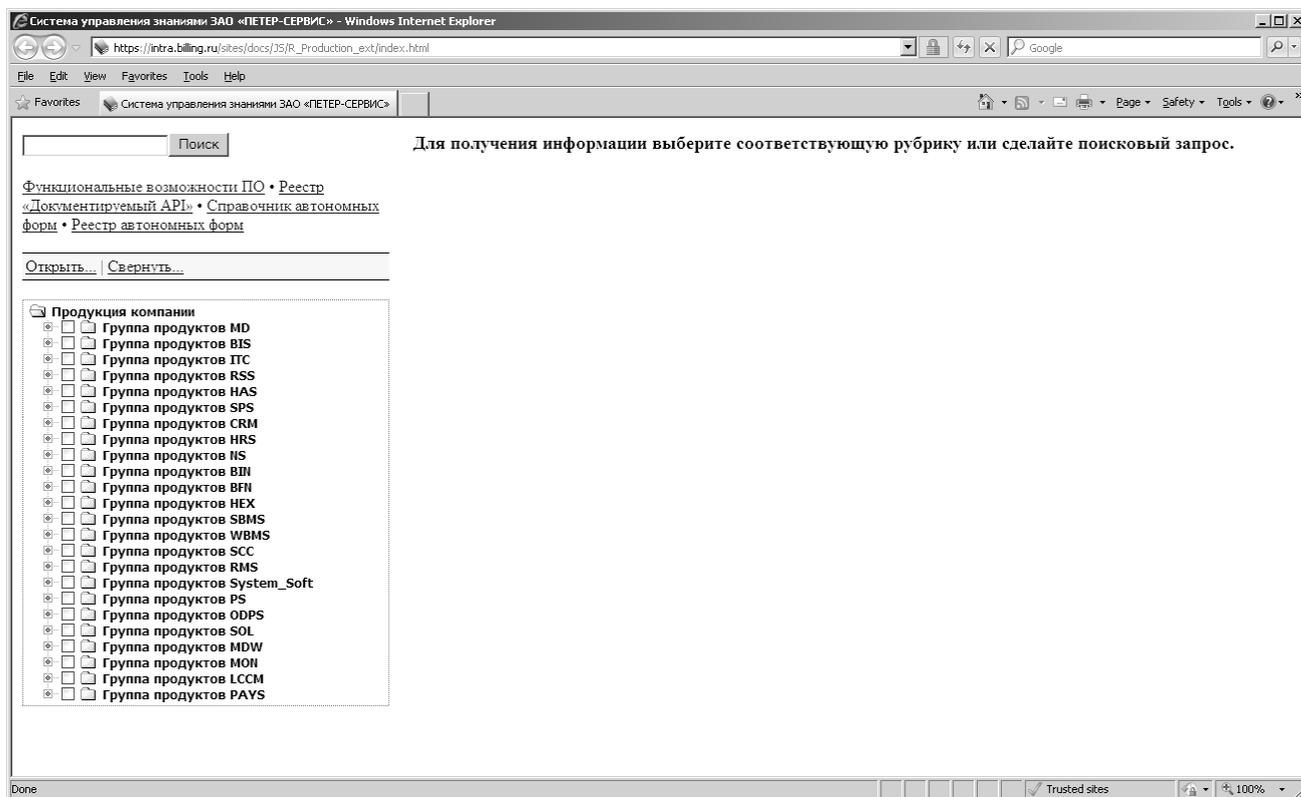


Рисунок 3.8 – Форма подсистемы поиска с иерархией онтологии продукции

После того, как пользователь в начальной форме выбирает определенную рубрику, связанной с данной рубрикой МД с типом контента «Возможность» автоматически отображается на экране. Дополнительно выводятся ссылки на все МД, связанные с данной рубрикой (функциональной возможностью), с группировкой по виду МД (рисунок 3.9).

Пользователь может перейти по ссылке на описание интересующего его МД. Также пользователю предоставляется возможность получить документ с текстом МД, всеми связанными МД и вывести его на печать.

Таким образом, пользователь получает полное описание выбранной функциональной возможности.

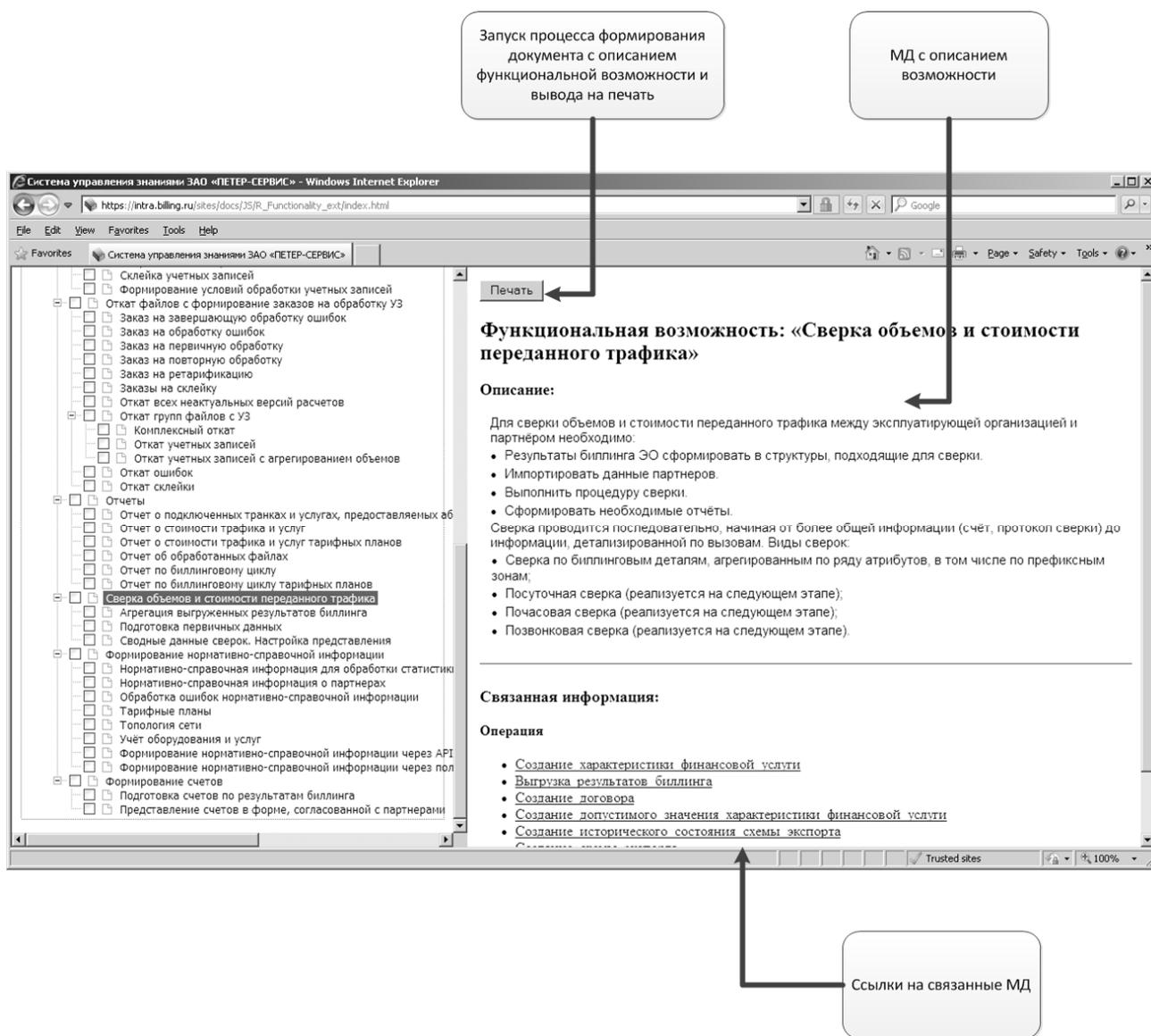


Рисунок 3.9 – Форма вывода МД «Возможность»

Для наиболее полной и точной формализации информационной потребности пользователь может указать дополнительные слова, которые должны присутствовать в требуемых МД или его свойствах.

В поле ввода поискового запроса пользователь вводит ключевое слово (слова) и при необходимости ограничивает выборку путем указания соответствующей рубрики (рубрик). Далее при помощи кнопки «Поиск» запускается процесс поиска, результаты которого выводятся в форме результатов поиска (рисунок 3.10).

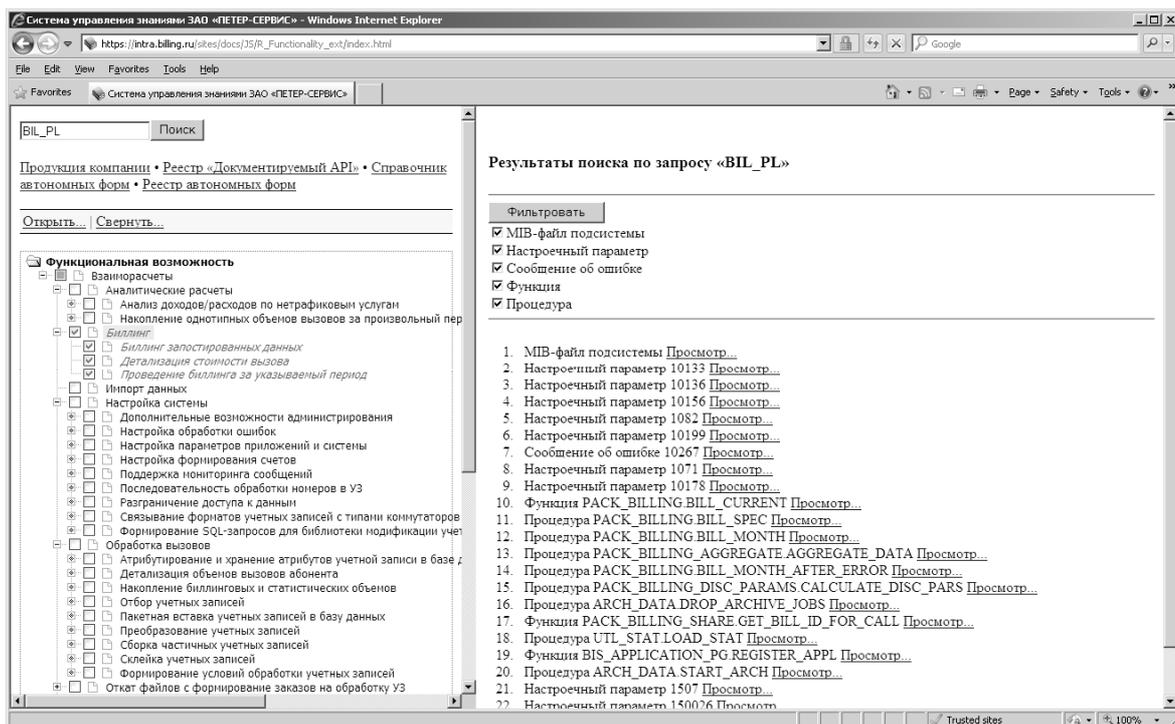


Рисунок 3.10 – Форма результатов поиска

Пользователь может указать дополнительное ограничение по виду МД и получить только МД соответствующего вида (рисунок 3.11).

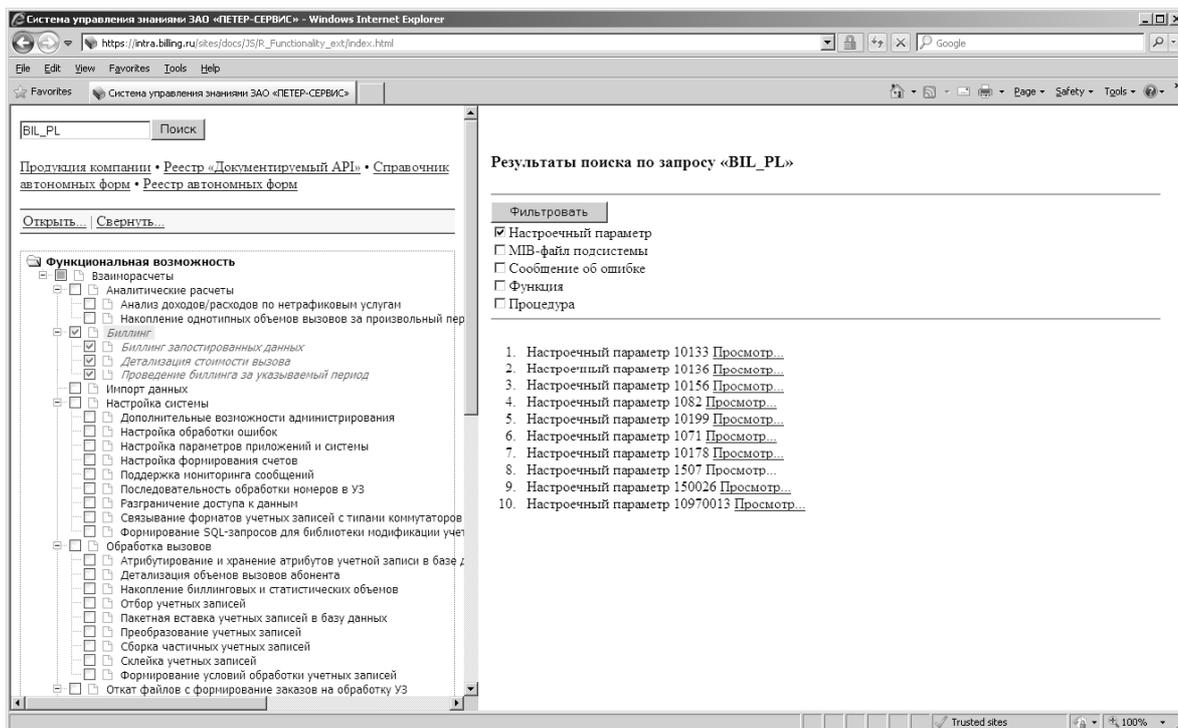


Рисунок 3.11 – Ограничение результатов поиска по виду МД

В форме результатов поиска пользователь может перейти по ссылке для просмотра текста МД, его семантических свойств и связей с другими МД в форму вывода МД (рисунок 3.12).

В данной форме пользователь может добавить комментарий или тег к МД при помощи штатного обработчика SharePoint socialdataframe.aspx. Входными параметрами обработчика являются идентификатор библиотеки документов (онтологии документации на ПС) и идентификатор МД.

Пример запроса:

https://intra.billing.ru/sites/docs/_layouts/socialdataframe.aspx?listid=%7BF1B8B86B%2DEA09%2D4C5F%2D8BD8%2D01F04DEF7937%7D&id=24428&mode=1&IsDlg=1

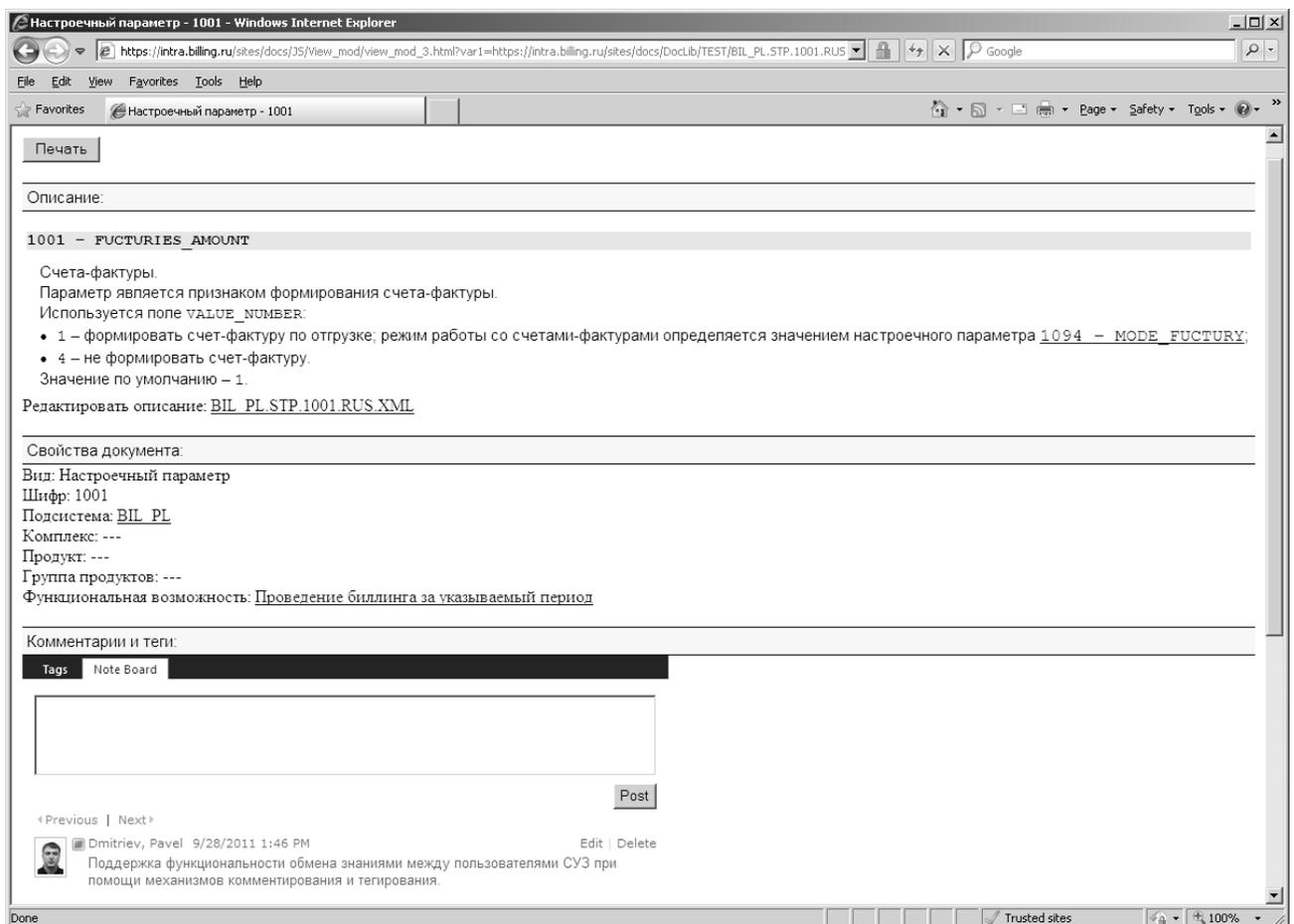


Рисунок 3.12 – Форма вывода МД «Настроенный параметр»

3.2.2. Подсистема компоновки документов

Подсистема компоновки документов предназначена для автоматического формирования требуемых цельных документов на соответствующие экземпляры концептов (классов) онтологии продукции (группу продуктов, подсистему и т.д.) в определенных форматах, а также подготовки различных реестров и справочников на основе свойств и связей МД.

Компоновка цельных документов производится на основе predetermined шаблонов документов. Для каждого вида документа на ПС создается свой шаблон.

Необходимые для формирования документы задаются в списке «Документы», в котором определяется соответствие между экземпляром онтологии продукции и видом документа на ПС (таблица 5). Список заполняется документатором, ответственным за документирование экземпляра продукции.

Таблица 5 – Структура списка «Документы»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Код файла документа	Наименование файла документа	Строка (Single line of text)	-
2.	Вид документа	Наименование вида документа	Выбор (Choice)	«Руководство оператора», «Руководство системного программиста», «Описание комплекса» и т.д.
3.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-
4.	Продукт	Код продукта из списка «Продукты»	Внешние данные (External Data)	-
5.	Группа продуктов	Код группы продуктов из списка «Группы продуктов»	Внешние данные (External Data)	-
6.	Решение	Код решения из списка «Решения»	Внешние данные (External Data)	-

Продолжение таблицы 5

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
7.	Опция	Код опции из списка «Опции»	Внешние данные (External Data)	-
8.	Комплекс	Код комплекса из списка «Комплексы»	Внешние данные (External Data)	-

Алгоритмы формирования руководства системного программиста на подсистему и справочника настроечных параметров на группу продуктов на основе predetermined шаблонов были рассмотрены в разделе 2.3.

Обобщенный алгоритм формирования документа состоит из следующей последовательности шагов:

- получение списка МД, связанных с указанным экземпляром концепта выпускаемой продукции;
- вывод на экран МД в определенной последовательности и стилевом оформлении;
- сохранение документа (вывод на печать).

При формировании документов в формате, отличном от HTML, этап вывода на экран отсутствует.

Для получения списка МД, связанных с экземпляром концепта выпускаемой продукции, используется обращение к Web-службе списков (Lists) при помощи метода GetListItems, рассмотренного в подразделе 3.2.1. Возможность формирования документа определяется подсистемой разграничения доступа.

На рисунок 3.13 показана форма вывода для вида МД «Назначение подсистемы», в которой дополнительно присутствуют элементы управления запуском процесса компоновки документов.

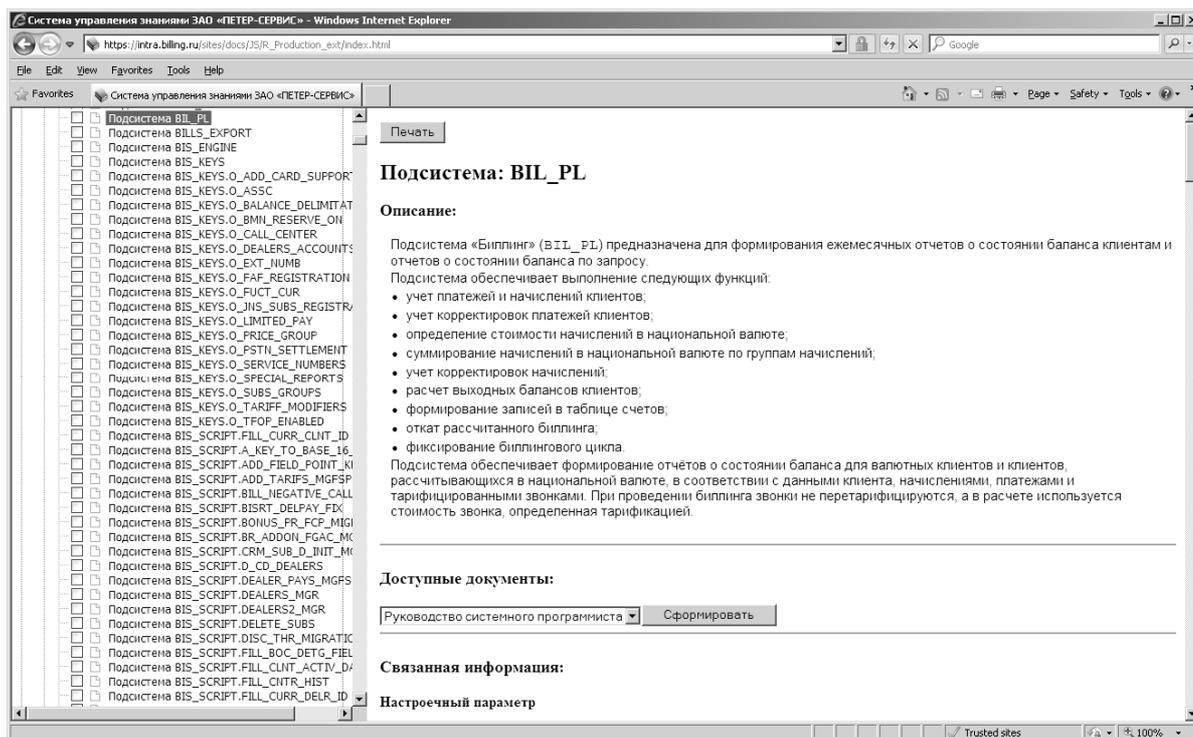


Рисунок 3.13 – Форма вывода для вида МД «Назначение подсистемы»

Пример результата компоновки руководства системного программиста на подсистему в HTML-формате показан на рисунке 3.14.

Реализация шаблона для формирования руководства системного программиста в HTML-формате приведена в Приложении Ж.

Реестры и справочники представляют собой формы, в которые выводятся соответствующие текст из МД или ссылки на них, удовлетворяющий определенному условию. Например, реестр «Документируемый API» представляет собой HTML-страницу, на которую выводятся ссылки на МД вида «Процедура» или «Функция» для выбранной пользователем подсистемы (рисунок 3.15).

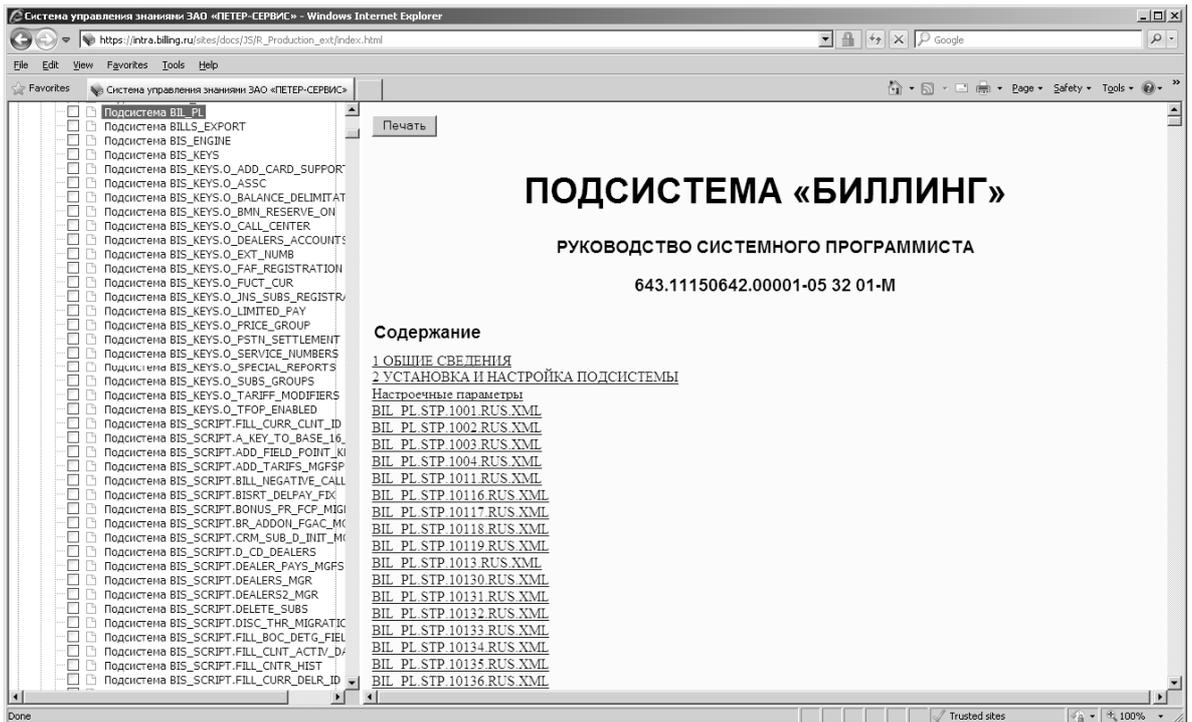


Рисунок 3.14 – Руководство системного программиста на подсистему

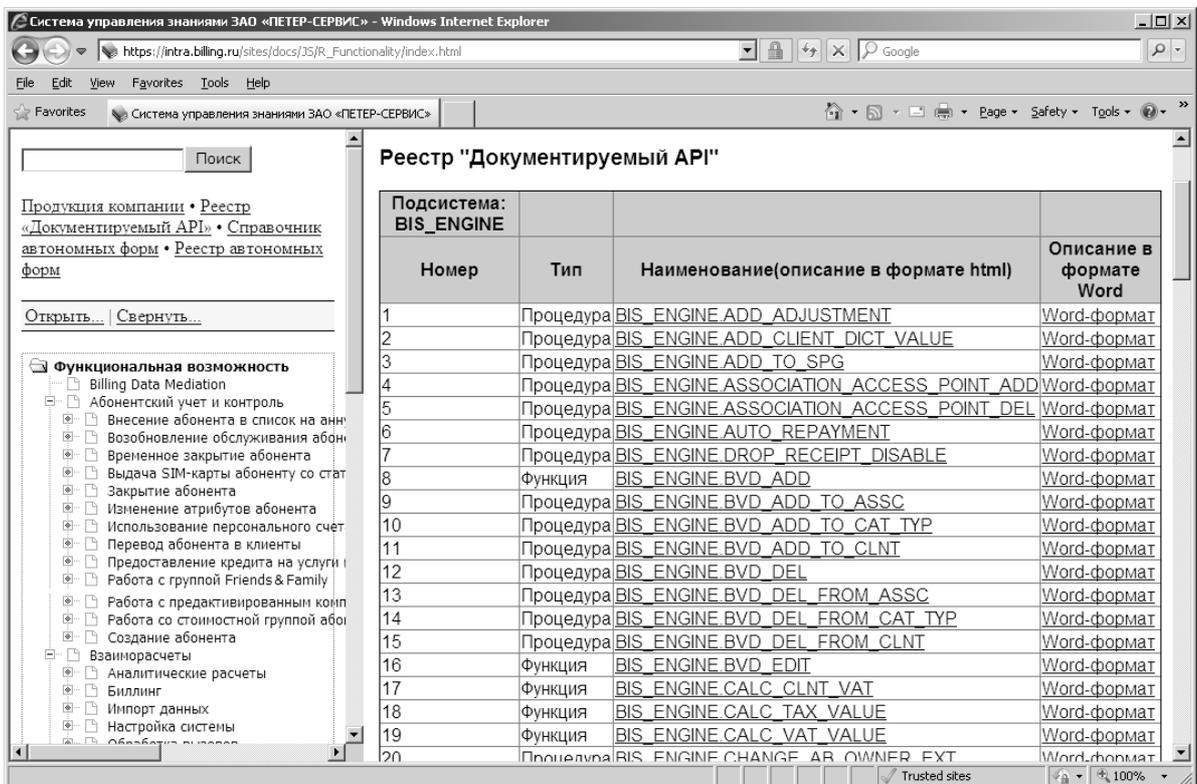


Рисунок 3.15 – Реестр «Документируемый API»

3.2.3. Подсистема разграничения доступа

Подсистема разграничения доступа обеспечивает разграничение доступа к информационным ресурсам.

В основе подсистемы разграничения доступа лежит штатный механизм разграничения доступа Microsoft SharePoint. В SharePoint управление доступом к Web-сайтам, спискам, папкам и элементам списков осуществляется при помощи ролей. Определенным пользователям или группам пользователей назначаются роли, разрешающие доступ к объектам SharePoint.

Определение роли (уровень разрешений) представляет собой список прав, связанных с ролью. Право – это контролируемое действие на Web-сайте SharePoint, предназначенное для некоторого пользователя. Например, пользователь с ролью «Чтение» может просматривать страницы на Web-сайте и элементы в списках. Все разрешения пользователей и групп управляются с помощью ролей. Определение роли – это коллекция прав, связанных с определенным объектом. Определения ролей (например, «Полный доступ», «Чтение», «Участие», «Проектирование» или «Ограниченный доступ») действуют на конкретном Web-сайте и имеют одинаковое значение на всех страницах сайта. Назначение роли – это связь между определением роли, пользователями, группами и областью действия (например, один пользователь может иметь право на чтение списка 1, а другой пользователь может иметь право на чтение списка 2). Связь, выражающаяся путем назначения роли, является ключевым объектом процесса ролевого управления безопасностью SharePoint [82].

Все разрешения управляются с помощью ролей, права никогда не назначаются пользователям напрямую. Назначаются только осмысленные коллекции прав (определения ролей), которые хорошо определены и согласованы. Управление уникальными разрешениями происходит путем добавления пользователей и групп в определения ролей или их удаления.

Таким образом, все пользователи СУЗПП объединяются в группы пользователей. Группе пользователей назначается соответствующая роль (роли).

Как уже было отмечено в разделе 3.1, СУЗПП располагается в пределах одного узла SharePoint. Полный перечень пользователей, групп пользователей и соответствующие им роли на узле представлены в таблице 6.

Таблица 6 – Пользователи и группы пользователей СУЗПП

№ п/п	Пользователь/ группа пользователей	Наименование роли	Объекты узла «Разработка документации»	Права
1.	Инженер по знаниям (документатор-эксперт)	«Полный доступ»	Весь узел, в том числе библиотека документов (онтология документации на ПС), иерархия управляемых метаданных (онтология функциональностей) и т.д.	Все действия в пределах узла
2.	Сотрудники группы документации	«Группа документации»	Библиотека документов (онтология документации на ПС)	Добавление, изменение и удаление
3.	Менеджеры групп продуктов	«Продукция»	Списки SharePoint (онтология продукции)	Добавление, изменение и удаление
4.	Руководители проектов	«Доступ»	Списки SharePoint (списки подсистемы разграничения доступа)	Добавление, изменение и удаление
5.	Сотрудники предприятия – разработчика ПО (внутренние пользователи)	«Предприятие»	Весь узел	Чтение
6.	Сотрудники предприятия – заказчика «1»	«Предприятие 1»	Библиотека документов (онтология документации на ПС), иерархия управляемых метаданных (онтология функциональностей), списки SharePoint (онтология продукции)	Удаленный доступ на чтение
7.	Сотрудники предприятия – заказчика «N»	«Предприятие N»	Библиотека документов (онтология документации на ПС), иерархия управляемых метаданных (онтология функциональностей), списки SharePoint (онтология продукции)	Удаленный доступ на чтение

Самый большой уровень разрешений имеет пользователь – инженер по знаниям. Самый низкий – сотрудники предприятия – заказчика. Более высокие группы имеют все права более низких групп. Например, сотрудники группы документации предприятия – разработчика ПО, объединены в одну группу с ролью «Группа документации», имеют права на добавления, изменение и удаление ресурсов к библиотеке документов SharePoint (онтологии документации на ПС), а также права на чтение остальных ресурсов узла.

Так как сотрудники предприятия – заказчика должны иметь доступ только к той части ресурсов онтологии информационной системы, которые включены в комплект поставки согласно договору между предприятием – разработчиком ПО и предприятием – заказчиком, то необходим дополнительный механизм разграничения доступа. Т.е. сотрудник предприятия – заказчика должен иметь доступ только к тем МД, которые входят в состав купленного данным предприятием ПС.

Для организации дополнительного механизма разграничения доступа к ресурсам онтологии информационной системы для конкретного пользователя созданы списки SharePoint «Заказчики» и «Продукты заказчиков».

В списке «Заказчики» задается соответствие между наименованием заказчика и его группой пользователей в SharePoint. В списке «Продукты заказчиков» задается соответствие между наименованием заказчика и продуктами, доступ к которым ему разрешен.

С целью ознакомления заказчиков с ПС, доступ к которому закрыт, создан список «Исключения». В данном списке задаются виды МД, доступ к которым разрешен, например, «Назначение продукта», «Назначение подсистемы» и т.д.

Ведение списков «Заказчики» и «Продукты заказчиков» подсистемы разграничения доступа выполняется руководителями проектов. Ведение списка «Исключения» выполняется инженером по знаниям.

Структура списков подсистемы разграничения доступа приведена в Приложении 3.

Алгоритм организации доступа для конкретного пользователя при взаимодействии с подсистемой поиска состоит из следующей последовательности этапов:

1. Проверка вхождения вида МД, полученного в результате поиска, в список «Исключения».
2. Если вид МД входит в список, то пользователю предоставляется доступ к МД.
3. Иначе выполняется определение имени учетной записи пользователя.
4. Определение группы пользователя.
5. Определение заказчика в списке «Заказчики» по группе пользователя.
6. Проверка вхождения экземпляра концепта выпускаемой продукции, заданного для МД, в список «Продукты заказчиков» для данного заказчика и записей со статусом «Используется».
7. Если для МД задан экземпляр концепта выпускаемой продукции отличный от концепта «Продукт», то выполняется дополнительный запрос к онтологии продукции для определения экземпляра концепта «Продукт», в который входит экземпляр концепта МД.
8. Если данному заказчику доступ к данному экземпляру концепта выпускаемой продукции разрешен, то пользователю предоставляется доступ к МД.
9. Иначе доступ к МД пользователю не предоставляется.

При этом подразумевается, что МД, для которых задан экземпляр концепта выпускаемой продукции выше концепта «Продукт» в онтологии продукции («Группа продуктов», «Решение») или задан концепт «Комплекс», не имеют ограничений на доступ. Т.е. проверка доступа выполняется для экземпляров концептов «Подсистема», «Опция» и «Продукт».

Алгоритм организации доступа для конкретного пользователя при взаимодействии с подсистемой компоновки документов состоит из следующей последовательности этапов:

1. Определение имени учетной записи пользователя.
2. Определение группы пользователя.
3. Определение заказчика в списке «Заказчики» по группе пользователя.
4. Проверка вхождения экземпляра концепта выпускаемой продукции, указанно пользователем для генерации соответствующего вида документа, в список «Продукты заказчиков» для данного заказчика и записей со статусом «Используется».
5. Если указан экземпляр концепта выпускаемой продукции отличный от концепта «Продукт», то выполняется дополнительный запрос к онтологии продукции для определения экземпляра концепта «Продукт».
6. Если данному заказчику доступ к данному экземпляру концепта выпускаемой продукции разрешен, то пользователю предоставляется доступ к генерации документа.
7. Иначе доступ к генерации документа пользователю не предоставляется.

При этом проверка вхождения экземпляра концепта выпускаемой продукции, указанного пользователем для генерации соответствующего вида документа производится только для экземпляров концептов «Подсистема», «Опция» и «Продукт». Для экземпляров концептов «Группа продуктов», «Решение» и «Комплекс» проверка доступа не производится.

Для определения группы пользователя используется обращение к Web-службе работы с пользователями и группами (Users and Groups) при помощи метода `GetGroupCollectionFromUser` [176].

Для получения значений из списков «Заказчики», «Продукты заказчиков», «Исключения» и списков онтологии продукции используются

обращения к Web-службе списков (Lists) при помощи метода GetListItems, рассмотренного в подразделе 3.2.1.

Примеры реализации обращений к Web-службам приведены в Приложении Д.

3.2.4. Подсистема целостности онтологии

Подсистема целостности онтологии обеспечивает целостность онтологии информационной системы.

В качестве рабочего определения понятия «целостность онтологии» в настоящей работе используется следующее:

Целостность онтологии – свойство онтологии, означающее, что онтология содержит непротиворечивые значения атрибутов (свойств концептов), необходимые для корректного функционирования СУЗ.

В разрабатываемой СУЗПП целостность онтологии поддерживается при помощи механизма ограничений целостности на значения атрибутов онтологии информационной системы и рабочих процессов подсистемы целостности онтологии.

Ограничения целостности на значения атрибутов онтологии информационной системы реализуются с помощью ограничения типа «Required (Must contain information)», т.е. часть атрибутов соответствующего концепта онтологии информационной системы не могут содержать неопределенное значение. Например, для родительского концепта (типа контента) «Модуль документации» определены ограничения целостности для атрибутов «Язык» и «Функциональная возможность». Для концепта «Настроечный параметр» определены ограничения целостности для атрибутов «Язык» и «Функциональная возможность» (от родительского типа «Модуль документации»), а также для атрибутов «Шифр» и «Подсистема».

Аналогичным образом определены ограничения целостности для остальных концептов онтологии информационной системы.

В состав подсистемы целостности онтологии входят рабочие процессы (РП) для обеспечения и контроля целостность онтологии. РП обеспечивают способ выполнения ряда действий, которые соответствуют тому или иному процессу. РП могут контролировать большинство аспектов сайта, списков, библиотек, типов контента или элементов списков SharePoint, включая ЖЦ элементов. РП может включать как действия, выполняемые пользователями (или участниками РП), так и действия, выполняемые самим РП. При создании РП задаются события, действия, условия и шаги [87]. Для создания РП использовано приложение Microsoft SharePoint Designer.

На сегодняшний день создано 3 РП, которые обеспечивают и контролируют целостность онтологии: «Формирование значения поля Title», «Нотификация при долгом Checkout» и «Нотификация при ошибке заполнения». Данные РП являются базовыми. По мере развития и обслуживания СУЗПП число РП должно увеличиться.

РП «Формирование значения поля Title»

РП «Формирование значения поля Title» создан для библиотеки документов онтологии документации на ПС. РП выполняет установку значения атрибута «Title» в значение атрибута «Name» без расширения файла (описание атрибутов приведено в разделе 2.2). Поле «Title» используется в механизме поиска SharePoint.

РП автоматически запускается в момент создания или изменения МД. Условия и действия РП приведены на рисунке 3.16.

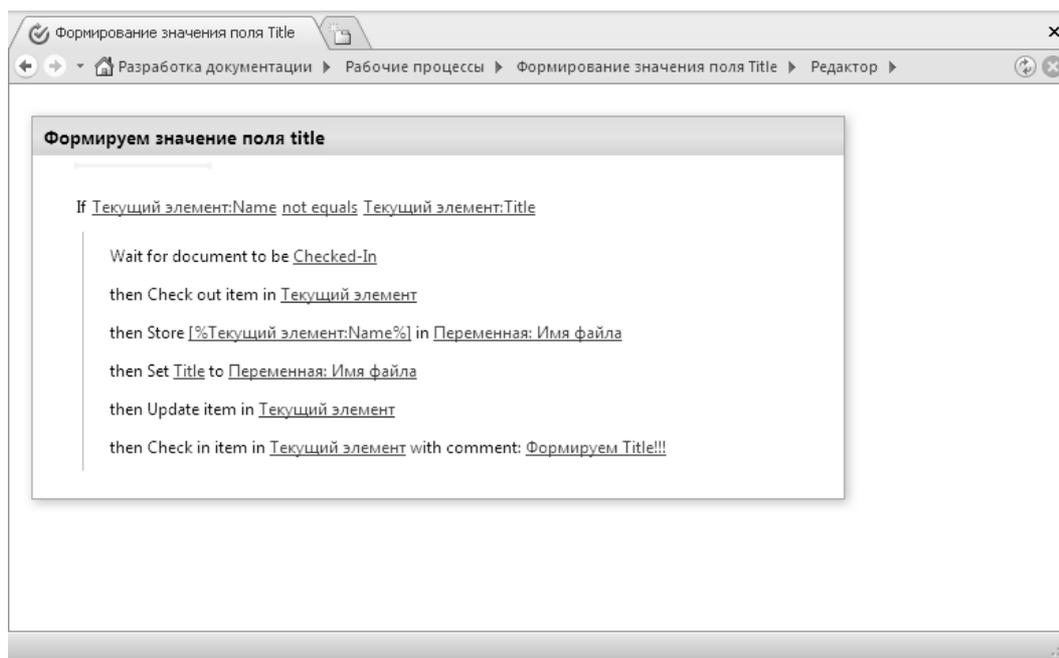


Рисунок 3.16 – Реализация РП «Формирование значения поля Title»

Диаграмма РП «Формирование значения поля Title» приведена на рисунке 3.17.

РП «Нотификация при долгом Checkout»

РП «Нотификация при долгом Checkout» создан для библиотеки документов онтологии документации на ПС. РП выполняет нотификацию документатора создавшего/изменившего МД, если МД не был возвращен на сервер в течение длительного промежутка времени (в текущей реализации – 30 дней).

РП автоматически запускается в момент создания или изменения. Условия и действия РП приведены на рисунке 3.18.

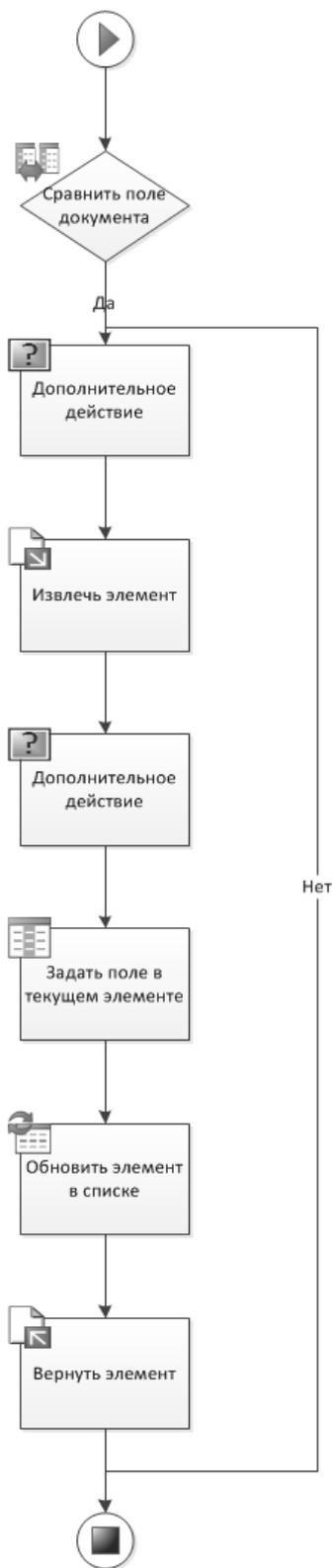


Рисунок 3.17 – Диаграмма РП «Формирование значения поля Title»

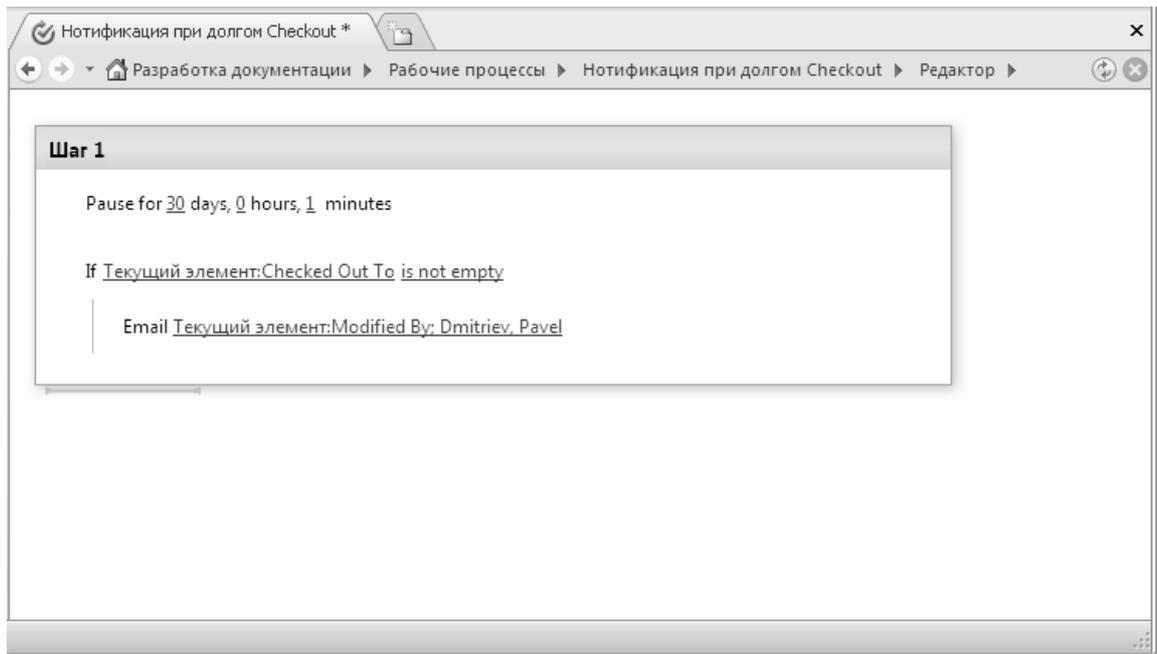


Рисунок 3.18 – Реализация РП «Нотификация при ошибке заполнения»

Диаграмма РП «Нотификация при ошибке заполнения» приведена на рисунке 3.19.



Рисунок 3.19 – Диаграмма РП «Нотификация при ошибке заполнения»

РП «Нотификация при ошибке заполнения»

РП «Нотификация при ошибке заполнения» создан для библиотеки документов онтологии документации на ПС. РП выполняет нотификацию документатора, создавшего/изменившего МД, если для МД не был задан его вид – тип контента (Content Type), т.е. МД не был отнесен ни к одному концепту онтологии документации на ПС (по умолчанию – «Не определен»).

РП автоматически запускается в момент создания или изменения МД. Условия и действия РП приведены на рисунке 3.20.

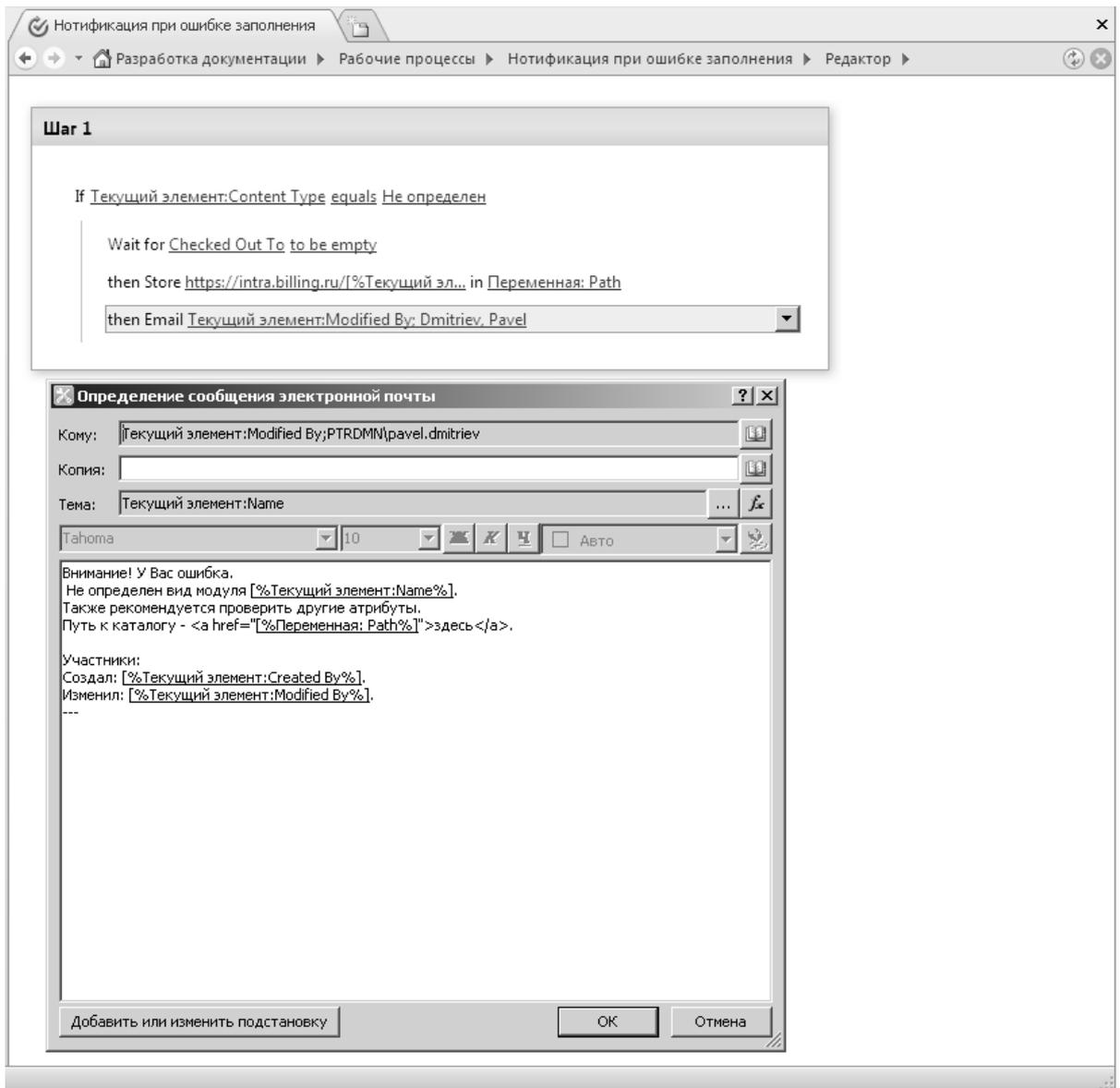


Рисунок 3.20 – Реализация РП «Нотификация при ошибке заполнения»

Диаграмма РП «Нотификация при ошибке заполнения» приведена на рисунке 3.21.

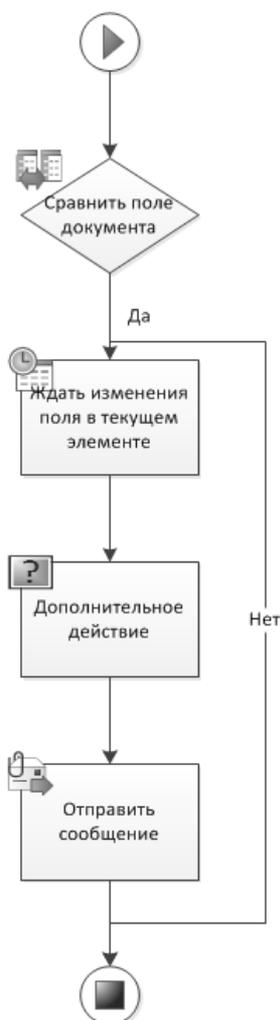


Рисунок 3.21 – Диаграмма РП «Нотификация при ошибке заполнения»

3.3. Методика оценки качества базовых процессов жизненного цикла программных средств при внедрении системы управления знаниями

3.3.1. Расчет комплексного показателя качества процесса

Опытная проверка и внедрение СУЗПП на предприятии ЗАО «Петер-Сервис» выполнялось 2013 году. Перевод документации на модульную структуру выполняется поэтапно с 2010 года. Для расчета показателей качества были выбраны данные из системы учета трудозатрат и

единой системы учета и обработки дефектов на платформе Serena Business Mashup, в которой фиксируются все дефекты, найденные внутри компании на этапах тестирования и контрольной инсталляции, а также сообщения потребителей (заказчиков).

Анализ данных выполнялся за одинаковые периоды (первое полугодие 2012 года (Т_1) и первое полугодие 2013 года (Т_2)) до и после внедрения СУЗПП в разрезе аналогичных по сложности и структуре двух групп продуктов, имеющих примерно равные проектные команды: группа продуктов (ГП_1), в которой документация переведена на модульную структуру, и группа продуктов (ГП_2), в которой документация продолжала разрабатываться в виде цельных документов. В таблице 7 представлены значения основных показателей качества процесса проектирования и разработки в части документации на ПС до и после внедрения СУЗПП.

Комплексный показатель качества рассчитан по следующей формуле:

$$Q = - \sum_{i=1}^n \alpha_i Q_i, \quad (10)$$

где Q_i – значение i -го частного показателя качества; α_i – весовой коэффициент i -го частного показателя; n – число показателей.

Весовые коэффициенты определены на основе метода экспертных оценок (Приложение И). Пусть x_{ij} – оценка показателя i , данная j -ым экспертом, m – число экспертов. Весовой коэффициент i -го частного показателя, подсчитанный по оценкам всех экспертов, равен:

$$\alpha_i = \frac{\sum_{j=1}^m \alpha_{ij}}{m}, \quad i = \overline{1, n}, \quad (11)$$

где α_{ij} – вес i -го показателя, подсчитанный по оценкам j -го эксперта, равен:

$$\alpha_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}}, \quad i = \overline{1, n}, \quad j = \overline{1, m}. \quad (12)$$

Таблица 7 – Основные показатели качества процесса проектирования и разработки в части документации на ПС до и после внедрения СУЗПП

Наименование показателя	Единица измер.	ГП_1				ГП_2				Весовой коэффициент
		Т_1	Т_2	Величина изменения, %	Показатель качества	Т_1	Т_2	Величина изменения, %	Показатель динамики	
Среднее количество ошибок, найденных на этапе тестирования	Шт. в месяц	94,5	66,5	-30	-0,30	69,5	66	-5	-0,05	0,12
Среднее количество ошибок, найденных на этапе контрольной инсталляции	Шт. в месяц	8,5	6,75	-21	-0,21	4,5	5	11	0,11	0,24
Среднее количество сообщений об ошибках в документации, полученных от заказчиков	Шт. в месяц	4,5	3,5	-22	-0,22	5,75	5	-13	-0,13	0,40
Среднее количество сообщений с вопросами по эксплуатации ПС, полученных от заказчиков	Шт. в месяц	26,5	18,5	-30	-0,30	19,5	18,5	-5	-0,05	0,24
Комплексный показатель качества	0,25					0,04				

Состав частных показателей качества (Q_i) определен на основе критериев оценки соответствующего базового процесса ЖЦ ПС, рассмотренных в разделе 1.3.

Для точности экспертных оценок была определена согласованность мнений экспертов с помощью коэффициента конкордации:

$$W = \frac{12 \sum_{i=1}^n (s_j - \bar{s})^2}{m^2 \times (n^3 - n)}, \quad (13)$$

где $\sum_{i=1}^n (s_j - \bar{s})^2$ – сумма квадратов разницы между суммой рангов по показателю и средней суммой рангов; m – число экспертов; n – число показателей. Полученный коэффициент конкордации свидетельствует о согласованности мнений группы экспертов.

Как видно из таблицы 7, после внедрения СУЗПП по группе продуктов ГП_1 наблюдается повышение качества процесса проектирования и разработки в части документации на ПС – значение комплексного показателя качества увеличилось на 25%. Тогда как для группы продуктов ГП_2 значение показателя осталось практически на том же уровне. При этом других кардинальных мер по улучшению процесса по группе продуктов ГП_1 не предпринималось.

Повышение качества процесса проектирования и разработки в части документации на ПС также подтвердил экспертный опрос сотрудников компании, занимающихся разработкой и поддержанием в актуальном состоянии документации (таблица 8). В результате опроса значение показателя результативности внедрения СУЗПП составило 5 баллов по шкале от 0 до 5 при достаточной согласованности мнений экспертов (значение коэффициента вариации равно 0,14).

Таблица 8 – Результаты опроса документаторов

Наименование показателя	Единица измерения	Значение показателя
Результативность внедрения СУЗПП	Шкала от 0 до 5	5
Среднее время увеличения затрат, связанное с переводом существующей документации на модульную структуру	%	17
Среднее время уменьшения затрат после внедрения СУЗПП	%	20

Помимо разработки новых программных продуктов большинство предприятий – разработчиков ПО предоставляют услуги по внедрению (поставке) и сопровождению. Для многих компаний на рынке информационных технологий процессы внедрения и сопровождения является главным или единственным источником доходов. Так как процесс сопровождения наиболее долгосрочный и прибыльный, то улучшение данного процесса является одной из наиболее приоритетных задач.

В таблице 9 представлены значения основных показателей качества процесса сопровождения в части мониторинга и анализа сообщений потребителей до и после внедрения СУЗПП. Как видно из таблицы 9, после внедрения СУЗПП по группе продуктов ГП_1 наблюдается повышение качества процесса сопровождения – значение комплексного показателя качества увеличилось на 21% (снижение количества сообщений от заказчиков и времени на их анализ). Тогда как для группы продуктов ГП_2 значение показателя осталось практически на том же уровне. При этом других мер по улучшению процесса не предпринималось.

Оценка качества по остальным базовым процессам ЖЦ ПС не проводилась ввиду небольших изменений в данных процессах, связанных с внедрением СУЗПП. Однако положительный результат применения СУЗПП при решении задач в остальных процессах ЖЦ ПС достигается за счет более быстрого и полного поиска информации о уже реализованной программной продукции.

Таблица 9 – Основные показатели качества процесса сопровождения в части мониторинга и анализа сообщений потребителей

Наименование показателя	Единица измер.	ГП_1				ГП_2				Весовой коэффициент
		T_1	T_2	Величина изменения, %	Показатель качества	T_1	T_2	Величина изменения, %	Показатель динамики	
Среднее время на анализ и выработку решения по сообщению с замечанием по постановке задачи (категория Р)	Час	6	5,5	-8	-0,08	4	3,5	-13	-0,13	0,24
Среднее время на анализ и выработку решения по сообщению с ошибкой в компонентах ПС (категория S)	Час	15	13	-13	-0,13	13	12,5	-4	-0,04	0,33
Среднее время на анализ и выработку решения по сообщению с ошибкой в документации (категория D)	Час	6	3,3	-45	-0,45	4	4,5	13	0,13	0,23

Продолжение таблицы 9

Наименование показателя	Единица измер.	ГП_1				ГП_2				Весовой коэффициент
		T_1	T_2	Величина изменения, %	Показатель качества	T_1	T_2	Величина изменения, %	Показатель динамики	
Среднее время на анализ и ответ по сообщению с вопросами по эксплуатации ПС (категория Q)	Час	24	14	-42	-0,42	16	15	-6	-0,06	0,08
Среднее время на анализ и выработку решения по сообщению с замечанием, связанным с процессом инсталляции подсистем продукта (категория I)	Час	12	11	-8	-0,08	8	7,5	-6	-0,06	0,12
Комплексный показатель качества	0,21				0,03					

3.3.2. Оценка трудозатрат

Экономическая эффективность от применения СУЗПП связана с уменьшением размера издержек в базовых процессах ЖЦ ПС, а также изменением качественных характеристик данных процессов.

В общем случае оценка экономического эффекта от применения СУЗПП в базовых процессах ЖЦ ПС требует проведения комплекса мероприятий по накоплению, анализу и обобщению статистических данных по затратам и факторам повышения качества в организационно – технических условиях конкретного предприятия – разработчика ПО.

Можно выделить следующие основные параметры, определяющие совокупный эффект от применения СУЗПП:

- качественное улучшение решения задач;
- уменьшение трудоемкости их решения;

Качественное улучшение решения задач в базовых процессах ЖЦ ПС было показано выше в подразделе 3.3.1.

Снижение затрат в процессе проектирования и разработки в части документации на ПС достигается за счет использования единого источника, создания генераторов документов и справочной системы.

Анализ данных о трудозатратах сотрудников, занимающихся разработкой и поддержанием в актуальном состоянии документации на ПС, показал снижение затрат на разработку документации по подсистемам до 20%, на разработку полностью генерируемой общей документации по группам продуктов до 90%. Пример снижения затрат на разработку справочника настроечных параметров на группу продуктов ГП_1 представлен на рисунке 3.22. В рамках 029.00 версии был создан шаблон для автоматического формирования документа и выполнена его генерация.

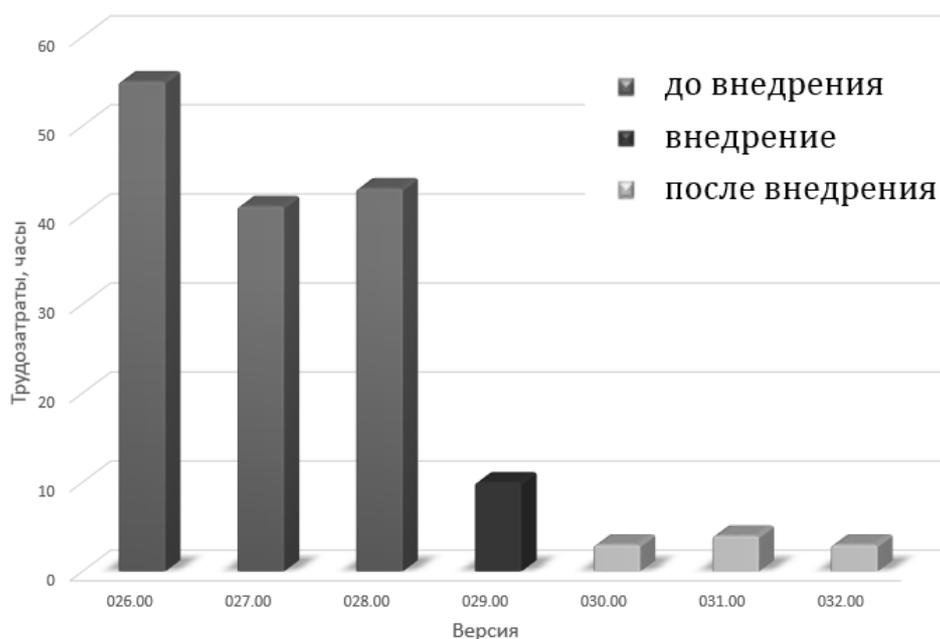


Рисунок 3.22 – Трудозатраты на разработку справочника настроечных параметров

Снижение затрат на сопровождение подтверждается уменьшением времени, которое требуется сотруднику отдела технического сопровождения, на анализ, выработку решения и ответ по сообщениям потребителя. Например, за анализируемый период по данным группы продуктов ГП_1 уменьшение среднего времени на анализ и выработку решения по сообщению с ошибкой в компонентах ПС составило 13%, среднего время на ответ по сообщению с вопросами по эксплуатации ПС – 42% (таблица 9).

3.3.3. Оценка качества процесса поиска информации о программной продукции

Для оценки качества процесса поиска информации о программной продукции выполнены следующие эксперименты с прототипом СУЗПП:

- полнотекстовый поиск по ключевым словам;
- поиск по значениям семантических свойств и связей МД;
- автоматическое формирование требуемых цельных документов.

В качестве экспериментального материала использованы:

- архив с цельными 132 документами на 100 подсистем;
- 2736 экземпляров МД по данным подсистемам.

С целью получения корректных результатов экспериментов указанные МД были скопированы в дубликат библиотеки документов SharePoint.

Трем экспертам было предложено вручную оценить результаты поиска и автоматического формирования требуемых цельных документов.

Результаты поиска оценивались по релевантности. Всем пользователям были предложены одинаковые запросы, которые были подобраны таким образом, чтобы алгоритм подсистемы поиска прототипа СУЗПП возвращал непустое множество результатов. Оценка релевантности результатов поиска выполнялась по шкале [0;1].

Для получения результатов поиска в архиве с цельными документами использованы система поиска Microsoft Windows Search и компонент поиска Microsoft Word (для поиска внутри цельных документов).

Для получения результатов поиска по МД использованы система поиска Microsoft Windows Search и подсистема поиска прототипа СУЗПП.

Для оценки результатов полнотекстового поиска по ключевым словам использовано 6 запросов (3 запроса с одним ключевым словом и 3 запроса с двумя ключевыми словами). Примеры запросов:

- «INSERT_BILL_DETAILS» – наименование процедуры;
- «подключение услуги» – процесс подключения услуги абоненту клиента.

Для оценки результатов поиска по значениям семантических свойств и связей использовано 6 запросов (3 запроса с одним значением атрибута онтологии документации на ПС и 3 запроса с двумя значениями). Пример запроса: «Сообщение об ошибке 12830006» – сообщение об ошибке с идентификатором 12830006 (значения атрибутов «Content Type» и «Шифр»).

Результаты экспериментов представлены в таблице 10, где А – количество документов, найденных системой и релевантных с точки зрения

экспертов; В – количество вхождений в документе Microsoft Word; С – количество документов, найденных системой, но не релевантных с точки зрения экспертов; D – количество релевантных документов, не найденных системой.

Алгоритм поиска, реализованный в прототипе СУЗПП, показал более высокие результаты. Для составных запросов по ключевым словам и запросов по семантическим свойствам и связям в архиве с цельными документами в системе поиска Windows Search было получено для каждого запроса всего 1 и 2 результата в виде цельного документа, а сам поиск внутри документа не дал результатов. При этом в подсистеме поиска прототипа СУЗПП были получены результаты в виде нескольких релевантных МД.

Таблица 10 – Результаты экспериментов

№ п/п	Эксперимент	Эксперт	Поиск в архиве: Windows Search, А (В)			Поиск по МД: Windows Search, А (С, D)			Поиск по МД: прототип СУЗПП, А (С, D)		
			Запрос			Запрос			Запрос		
			1	2	3	1	2	3	1	2	3
1.	Полнотекстовый поиск по ключевым словам: запрос одно ключевое слово	1	7 (23)	4 (20)	4 (16)	13 (1, 2)	13 (2, 4)	11 (1, 3)	15 (1, 1)	15 (1, 2)	14 (1, 1)
		2	7 (23)	4 (20)	4 (16)	13 (1, 2)	12 (3, 4)	11 (1, 3)	15 (1, 1)	15 (1, 2)	13 (1, 2)
		3	7 (23)	4 (20)	4 (16)	12 (2, 2)	13 (2, 4)	11 (1, 3)	15 (1, 1)	15 (1, 2)	14 (1, 1)
2.	Полнотекстовый поиск по ключевым словам: запрос два ключевых слова	1	2 (0)	2 (0)	2 (0)	4 (1, 2)	5 (1, 1)	4 (1, 1)	5 (1, 1)	5 (1, 1)	5 (1, 1)
		2	2 (0)	2 (0)	2 (0)	4 (1, 2)	4 (1, 2)	4 (1, 1)	5 (1, 1)	5 (1, 1)	5 (1, 1)
		3	2 (0)	2 (0)	2 (0)	4 (1, 2)	5 (1, 1)	4 (1, 1)	5 (1, 1)	5 (1, 1)	5 (1, 1)
3.	Поиск по значениям семантических свойств и связей: одно значение атрибута	1	2 (2)	1 (3)	2 (4)	4 (1, 2)	9 (1, 20)	6 (1, 1)	6 (1, 0)	23 (2, 0)	8 (1, 0)
		2	2 (2)	1 (3)	2 (4)	4 (1, 2)	8 (2, 20)	6 (1, 1)	6 (1, 0)	23 (2, 0)	8 (1, 0)
		3	2 (2)	1 (3)	2 (4)	5 (1, 1)	9 (1, 20)	6 (1, 1)	6 (1, 0)	23 (2, 0)	8 (1, 0)

Продолжение таблицы 10

№ п/п	Эксперимент	Эксперт	Поиск в архиве: Windows Search, A (B)			Поиск по МД: Windows Search, A (C, D)			Поиск по МД: прототип СУЗПП, A (C, D)		
			Запрос			Запрос			Запрос		
			1	2	3	1	2	3	1	2	3
4.	Поиск по значениям семантических свойств и связей: два значения атрибутов	1	1 (0)	1 (0)	1 (0)	1 (0, 1)	1 (0, 1)	1 (0, 1)	2 (0, 0)	2 (0, 0)	2 (0, 0)
		2	1 (0)	1 (0)	1 (0)	1 (0, 1)	1 (0, 1)	1 (0, 1)	2 (0, 0)	2 (0, 0)	2 (0, 0)
		3	1 (0)	1 (0)	1 (0)	1 (0, 1)	1 (0, 1)	1 (0, 1)	2 (0, 0)	2 (0, 0)	2 (0, 0)

Для оценки качества процесса поиска информации были вычислены показатели качества поисковых систем: полнота и точность поиска (таблица 11).

Полнота (R) – отношение найденных релевантных документов к общему количеству релевантных документов:

$$R = \frac{A}{A + D}, \quad (14)$$

где A – количество документов, найденных системой и релевантных с точки зрения экспертов; D – количество релевантных документов, не найденных системой.

Таблица 11 – Показатели полноты и точности поиска по МД

№ п/п	Эксперимент	Эксперт	Windows Search, полнота/точность			Прототип СУЗПП, полнота/точность		
			Запрос			Запрос		
			1	2	3	1	2	3
1.	Полнотекстовый поиск по ключевым словам: запрос одно ключевое слово	1	0,867/ 0,929	0,765/ 0,867	0,786/ 0,917	0,938/ 0,938	0,882/ 0,938	0,933/ 0,933
		2	0,867/ 0,929	0,750/ 0,800	0,786/ 0,917	0,938/ 0,938	0,882/ 0,938	0,867/ 0,929
		3	0,857/ 0,857	0,765/ 0,867	0,786/ 0,917	0,938/ 0,938	0,882/ 0,938	0,933/ 0,933

Продолжение таблицы 11

№ п/п	Эксперимент	Эксперт	Windows Search, полнота/точность			Прототип СУЗПП, полнота/точность		
			Запрос			Запрос		
			1	2	3	1	2	3
2.	Полнотекстовый поиск по ключевым словам: запрос два ключевых слова	1	0,667/ 0,800	0,833/ 0,833	0,800/ 0,800	0,833/ 0,833	0,833/ 0,833	0,833/ 0,833
		2	0,667/ 0,800	0,667/ 0,800	0,800/ 0,800	0,833/ 0,833	0,833/ 0,833	0,833/ 0,833
		3	0,667/ 0,800	0,833/ 0,833	0,800/ 0,800	0,833/ 0,833	0,833/ 0,833	0,833/ 0,833
3.	Поиск по значениям семантических свойств и связей: одно значение атрибута	1	0,667/ 0,800	0,310/ 0,900	0,857/ 0,857	1/ 0,857	1/ 0,920	1/ 0,889
		2	0,667/ 0,800	0,286/ 0,800	0,857/ 0,857	1/ 0,857	1/ 0,920	1/ 0,889
		3	0,833/ 0,833	0,310/ 0,900	0,857/ 0,857	1/ 0,857	1/ 0,920	1/ 0,889
4.	Поиск по значениям семантических свойств и связей: два значения атрибутов	1	0,500/ 1	0,500/ 1	0,500/ 1	1/1	1/1	1/1
		2	0,500/ 1	0,500/ 1	0,500/ 1	1/1	1/1	1/1
		3	0,500/ 1	0,500/ 1	0,500/ 1	1/1	1/1	1/1
Общее значением показателя			0,67/ 0,85	0,58/ 0,87	0,79/ 0,89	0,97/ 0,90	0,94/ 0,93	0,97/ 0,91
			0,67/0,87			0,97/0,91		

Точность (P) – отношение найденных релевантных документов к общему количеству найденных документов:

$$P = \frac{A}{A + C}, \quad (15)$$

где A – количество документов, найденных системой и релевантных с точки зрения экспертов; C – количество документов, найденных системой, но не релевантных с точки зрения экспертов.

Полученные показатели полноты и точности оказались выше у подсистемы поиска прототипа СУЗПП и не зависят от числа экспертов, так как

в механизме поиска учитываются значения семантических свойств и связей МД. Для запросов по семантическим свойствам и связям показатели качества подсистемы поиска прототипа СУЗПП равны или близки к 1. При этом показатель полноты системы поиска Windows Search уменьшился. Общее значением показателя полноты и точности поиска увеличилось на 44% и 5% соответственно.

Таким образом, подсистема поиска прототипа СУЗПП является более качественной.

Для проверки автоматического формирования документов было сгенерировано 10 руководств системного программиста на 10 подсистем, выполнена их проверка на соответствие цельным документам. Все сгенерированные документы были оценены пользователями как полностью соответствующими цельным документам.

Таким образом, результаты проведенных экспериментов позволяют сделать вывод о достоверности основных положений диссертационной работы.

3.4. Выводы к главе 3

1. Разработана модель СУЗПП о программной продукции на основе синтеза компонент информационно-поисковых систем и систем искусственного интеллекта.
2. Показано, что автоматизация процесса управления знаниями на предприятии – разработчике ПО может быть эффективно реализована с использованием предлагаемой четырехуровневой архитектуры СУЗПП в виде Web-портала.
3. Для решения задачи поиска информации соединены два подхода: использование поисковой системы, обеспечивающей релевантную отработку формального запроса, и интерфейса Web-каталога, обеспечивающего пользователю понятный и удобный интерфейс. Система рубрик основана на иерархии онтологии функциональностей и онтологии

продукции. Пертинентность информационного поиска достигается за счет спецификации не только определенной рубрики, в которой ищется информация, но и вида требуемой информации.

4. Разработана методика оценки качества базовых процессов ЖЦ ПС при внедрении СУЗПП, основанная на применении методов оценки качества процессов ЖЦ ПС, экспертных оценок и оценки качества поисковых систем.
5. Разработан типовой стандарт организации «Система управления знаниями о программной продукции», регламентирующий правила работы с системой, роли пользователей системы и основные операции, которые доступны в рамках роли.
6. Разработано информационное и программно-алгоритмическое обеспечение подсистем СУЗПП. Информационное обеспечение реализовано на основе библиотеки документов и списков Microsoft SharePoint. В состав программно-алгоритмического обеспечения подсистем СУЗПП входят алгоритмы и совокупность реализующих их программных средств.

ЗАКЛЮЧЕНИЕ

В диссертационной работе достигнута цель, имеющая важное хозяйственное значение – повышение качества базовых процессов ЖЦ ПС на основе разработки и применения методов и средств УЗ.

В работе получены следующие новые научные результаты:

1. Разработан метод структуризации информации в документации на ПС на основе разработанных онтологических моделей представления знаний в базовых процессах ЖЦ ПС, позволяющий развить известный аппарат инженерии знаний путем разработки понятийно-смысловой структуры документации на ПС и обеспечить интеграцию знаний в базовых процессах ЖЦ ПС.
2. Разработана и апробирована методика создания СУЗПП, основанная на разработанных методе и моделях, позволяющая внедрить процесс управления знаниями в базовые процессы ЖЦ ПС для повышения их качества.
3. Разработана модель СУЗПП, отличающаяся тем, что в её основе лежит комбинация двух известных типов СУЗ (с использованием компонент информационно-поисковых систем и систем искусственного интеллекта), а также составом компонент, что позволяет учесть семантику информации, находящейся в документации на ПС, взаимосвязи и взаимозависимости, обеспечить целостность и непротиворечивость знаний, повысить качество поиска информации, обеспечивая тем самым повышение качества базовых процессов ЖЦ ПС.
4. Разработана и апробирована методика оценки качества базовых процессов ЖЦ ПС при внедрении СУЗПП, основанная на применении методов оценки качества процессов ЖЦ ПС, экспертных оценок и оценки качества поисковых систем.

Опытная проверка и применение результатов диссертационного исследования на предприятиях ЗАО «Петер-Сервис» и ОАО «НТЦ ЕЭС» дало практические результаты в виде роста показателей качества базовых процессов ЖЦ ПС и снижения трудозатрат, что подтверждается соответствующими актами.

СПИСОК ЛИТЕРАТУРЫ

1. Авдошин, С.М. Информационные технологии онтологического инжиниринга / С.М. Авдошин, М.П. Шатилов // Информационные технологии. – 2008. – №10. – С. 28-37.
2. Авдошин, С.М. Онтологический инжиниринг / С.М. Авдошин, М.П. Шатилов // Бизнес информатика. – 2007. – № 2. – С. 3-13.
3. Автоматизация разработки технической документации с применением AuthorIT [Электронный ресурс] // authorit.ru. – 2007. – Режим доступа: <http://authorit.ru/pdf/book.pdf>.
4. Агеев, М.С. Официальные метрики РОМИП'2004 [Электронный ресурс] / М.С. Агеев, И.Е. Кураленок // Российский семинар по оценке методов информационного поиска (РОМИП 2004). – Пущино. – 2004. – Режим доступа: http://www.cir.ru/docs/ips/publications/2004_romip_metrix.pdf.
5. Афонин, С.А. Использование онтологий в поисковых системах [Электронный ресурс] / С.А. Афонин, А.С. Козицын // Материалы всероссийской конференции с международным участием «Зания-Онтологии-Теории» (ЗОНТ-09). – Новосибирск: ИМ СО РАН. – 2009. – Режим доступа: <http://math.nsc.ru/conference/zont09/reports/29Afonin-Kozicyn.pdf>.
6. Базовая иерархия типов контента [Электронный ресурс] // Microsoft Corporation. – 2010. – Режим доступа: <http://technet.microsoft.com/ru-ru/library/ms452896.aspx>.
7. Брайчевский, С.М. Современные информационные потоки: актуальная проблематика / С.М. Брайчевский, Д.В. Ландэ // НТИ. Сер. 1. – 2005. – № 11. – С. 21-33.
8. Булатицкий, Д.И. Управление знаниями в системе менеджмента качества организации: дис. ... кан. тех. наук : 05.02.23. Брянский государственный технический университет. – Брянск. – 2010. – 209 с.
9. Валькман, Ю.Р. Метазнания — средство эффективного управления распределенными знаниями корпоративных систем / Ю.Р. Валькман, С.А. Майстренко // Труды XX национальной конференции по Искусственному Интеллекту с международным участием – КИИ-2006. – Т.3. – М.: Физматлит, 2006. – С. 1062-1070.
10. Введение в разработку приложений на платформе Atom/MeeGo. Реализация качества разработки программных приложений [Электронный ресурс] // Intel® Software Network. – 2010. – Режим доступа: <http://software.intel.com/ru-ru/articles/atom-meego-courseware-part12/>.
11. Веб-служба запросов Windows SharePoint Services [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: [http://msdn.microsoft.com/ru-ru/library/ms470518\(v=office.12\).aspx](http://msdn.microsoft.com/ru-ru/library/ms470518(v=office.12).aspx).

12. Гавриленко, Т.В. Представление знаний о динамической предметной области методами теоретико-множественного анализа : автореф. дис. ... канд. тех. наук : 05.13.01. Сургутский государственный университет. – Сургут. – 2004. – 20 с.
13. Гаврилова, Т.А. Активные индивидуальные методы извлечения знаний и данных [Электронный ресурс] / Т.А. Гаврилова // Enterprise Partner. – №21 (38). – 2001. – Режим доступа: http://bigc.ru/publications/bigspb/km/active_metod_izvl_knowlg_data.php.
14. Гаврилова, Т.А. Введение в онтологический инжиниринг / Т.А. Гаврилова // Искусственный интеллект – проблемы и перспективы. Политехнические чтения. Выпуск 7. – 2006. – С. 109-116.
15. Гаврилова, Т.А. Онтологический подход к управлению знаниями при разработке корпоративных информационных систем / Т.А. Гаврилова // Новости искусственного интеллекта. – 2003. – № 2. – С. 24-30.
16. Гаврилова, Т.А. Формирование прикладных онтологий / Т.А. Гаврилова // Труды XX национальной конференции по Искусственному Интеллекту с международным участием – КИИ-2006. – Т.2. – М.: Физматлит, 2006.
17. Гаврилова, Т.А. Использование онтологий в системах управления знаниями [Электронный ресурс] / Т.А. Гаврилова // «Бизнес Инжиниринг Групп» (БИГ-СПб). – 2003. – Режим доступа: http://bigc.ru/publications/bigspb/km/use_ontology_in_suz.php.
18. Гаврилова, Т.А. Извлечение знаний: лингвистический аспект [Электронный ресурс] / Т.А. Гаврилова // Enterprise Partner. – №10 (27). – 2001. – Режим доступа: http://bigc.ru/publications/bigspb/km/izvl_znaniy_ling_asp.php.
19. Гаврилова, Т.А. Извлечение знаний: "пассивные" методы [Электронный ресурс] / Т.А. Гаврилова // Enterprise Partner. – №11 (28). – 2001. – Режим доступа: http://bigc.ru/publications/bigspb/km/izvl_knowlg_passiv_metod.php.
20. Гаврилова, Т.А. Извлечение знаний: психологический аспект [Электронный ресурс] / Т.А. Гаврилова // Enterprise Partner. – №8 (25). – 2001. – Режим доступа: http://bigc.ru/publications/bigspb/km/izvl_knowlg_psych_aspect.php.
21. Гаврилова, Т.А. Разработка корпоративных систем управления знаниями [Электронный ресурс] / Т.А. Гаврилова, Л.Ю. Григорьев // «Бизнес Инжиниринг Групп» (БИГ-СПб). – 2003. – Режим доступа: http://bigc.ru/publications/bigspb/km/create_kms.php.
22. Гаврилова, Т.А. Модели структурирования и инструменты управления знаниями / Т.А. Гаврилова // Сборник докладов XXI научно-практической конференции «Реинжиниринг бизнес-процессов на основе современных информационных технологий. Системы управления знаниями». – М.: МЭСИ, 2008. – С. 47-50.

23. Гаврилова, Т.А. Информационные технологии управления знаниями / Т.А. Гаврилова, Д.В. Кудрявцев // В кн.: Инновационное развитие: экономика, интеллектуальные ресурсы, управление знаниями. Под ред. Б.З. Мильнера. – М.: ИНФРА-М. – 2009. – С. 500-516.
24. Гаврилова, Т.А. Управление знаниями как корпоративная стратегия автоматизации / Т.А. Гаврилова, В.Л. Веремьев // Сборник научных трудов 6-ой научно-практической конференции «Реинжиниринг бизнес-процессов на основе современных информационных технологий. Системы управления знаниями». – М.: МЭСИ, 2002. – С. 218-224.
25. Гаврилова, Т.А. Онтологический инжиниринг / Т.А. Гаврилова // Труды конференции "КИИ' 2002". – М.: Физматлит, 2002. – С. 845-853.
26. Гаврилова, Т.А. Управление знаниями: от слов к делу / Т.А. Гаврилова, Л.Ю. Григорьев, Д.В. Кудрявцев // "Intelligent Enterprise". – 2004. – №12-13 (101). – С.43-48.
27. Гладун, А.Я. Онтологии в корпоративных системах / А.Я. Гладун, Ю.В. Рогушина // Корпоративные системы. – 2006. – № 1. – С. 41-47.
28. Гладун, А.Я. Семантическая Википедия как источник онтологий для интеллектуальных поисковых систем А.Я. Гладун, Ю.В. Рогушина // XIVth International Conference "Knowledge-Dialogue-Solution" KDS. – 2008. – Varna. – Режим доступа: http://www.foibg.com/ibs_isc/ibs-02/IBS-02-p23.pdf.
29. Гличев, А.В. Основы управления качеством продукции / А.В. Гличев. – М.: Стандарты и качество, 2001. – 424 с.
30. Грибкова, О.В. Системы управления знаниями [Электронный ресурс] / О.В. Грибкова // Университет «Дубна». – 2006. – Режим доступа: http://www.uni-dubna.ru/~mazny/students/upr_zn/3_0.html.
31. Горленко, О.А. Создание системы менеджмента качества в организации: монография / О.А. Горленко. – М.: Машиностроение-1, 2003. – 126 с.
32. ГОСТ 2.601-95. Единая система конструкторской документации. Эксплуатационные документы.
33. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
34. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы стадии создания.
35. ГОСТ Р ИСО 9000-2011. Система менеджмента качества Основные положения и словарь.
36. ГОСТ Р ИСО 9001-2011. Системы менеджмента качества. Требования.
37. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Процессы жизненного цикла программных средств.

38. ГОСТ Р 51904-2002. Программное обеспечение встроенных систем. Общие требования к разработке и документированию.
39. ГОСТ Р ИСО/МЭК 15910-2002. Информационная технология. Процесс создания документации пользователя программного средства.
40. Джанетто, К. Управление знаниями. Руководство по внедрению корпоративной системы управления знаниями / К. Джанетто, Э. Уиллер. – М.: Добрая книга, 2007. – 192 с.
41. Дмитриев, П.И. Автоматизация процесса формирования комплектов технической документации на основе онтологического подхода / П.И. Дмитриев // Информация и космос. – 2011. – № 1. – С. 90-93.
42. Дмитриев, П.И. Использование модели eTOM в системе управления знаниями инфокоммуникационной системы / П.И. Дмитриев // Сборник научных трудов IV международного научного конгресса «Нейробиотелеком 2010». – СПб.: СПбГУТ, 2010. – С. 55-58.
43. Дмитриев П.И. Онтологическая модель системы управления знаниями инфокоммуникационной системы / П.И. Дмитриев // Телекоммуникации. – 2011. – № 12. – С. 8-13.
44. Дмитриев, П.И. Основные возможности Microsoft SharePoint 2010 для построения корпоративной системы управления знаниями / П.И. Дмитриев // Молодой ученый. – 2011. – №4. – Т.1. – С. 86-88.
45. Дмитриев, П.И. Основные ресурсы системы управления знаниями инфокоммуникационной системы / П.И. Дмитриев // Труды Международной научно-технической конференции «Системы и процессы управления и обработки информации» I ч. – СПб.: СЗТУ, 2010. – С. 182-189.
46. Дмитриев, П.И. Разработка системы управления знаниями инфокоммуникационной системы / П.И. Дмитриев // Труды XI Международной научно-практической конференции молодых ученых, студентов и аспирантов «Анализ и прогнозирование систем управления» I ч. – СПб.: СЗТУ, 2010. – С. 135-140.
47. Дмитриев, П.И. Система управления знаниями предприятия – разработчика программного обеспечения для телекоммуникационной отрасли / П.И. Дмитриев // Актуальные вопросы технических наук: материалы международной заочной научной конференции. – Пермь: Меркурий, 2011. – С. 37-41.
48. Добров, Б.В. Тезаурус RuTез как ресурс для решения задач информационного поиска [Электронный ресурс] / Б.В. Добров, Н.В. Лукашевич // Материалы всероссийской конференции с международным участием «Зания-Онтологии-Теории» (ЗОНТ-09). Новосибирск: ИМ СО РАН. – 2009. – Режим доступа: <http://www.math.nsc.ru/conference/zont09/reports/93Dobrov-Lukashevich.pdf>.

49. Ермаков, А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста / А.Е. Ермаков // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции "Диалог". Вып. 7 (14). – М.: РГГУ, 2008. – С. 154-159
50. Ермаков, А.Е. Извлечение знаний из текста и их обработка: состояние и перспективы / А.Е. Ермаков // Информационные технологии. – 2009. – №7. – С. 50-55.
51. Ермолаев, К.А. Использование онтологии для управления знаниями предприятия / К.А. Ермолаев // Исслед. по информ. Казань: Отечество. – 2007. – №12. – С.65-78.
52. Загорулько, Ю.А. Способы автоматического построения онтологии для задач анализа текстов [Электронный ресурс] / Ю.А. Загорулько, Е.А. Сидорова, О.И. Боровикова // Материалы всероссийской конференции с международным участием «Зания-Онтологии-Теории» (ЗОНТ-09). Новосибирск: ИМ СО РАН. – 2009. – Режим доступа: <http://www.math.nsc.ru/conference/zont09/reports/91Zagorulko-Sidorova-Borovikova.pdf>.
53. Захарова, И.В. Способы автоматического построения онтологии для задач анализа текстов [Электронный ресурс] / И.В. Захарова, М.С. Тимченко // Материалы всероссийской конференции с международным участием «Зания-Онтологии-Теории» (ЗОНТ-09). Новосибирск: ИМ СО РАН. – 2009. – Режим доступа: <http://math.nsc.ru/conference/zont09/reports/67Zaharova-Timchenko.pdf>.
54. Ивлиев, В. Применение программных средств в системе менеджмента качества / В. Ивлиев // Стандарты и качество. – 2004. – № 1. – С. 80-84.
55. Интеллектуальные системы принятия проектных решений / А.В. Алексеев [и др.]. – Рига: Зинатне, 1997. – 320 с.
56. Искусственный интеллект. Кн. 2: Модели и методы: Справочник / под ред. Э.В. Попова. – М.: Радио и связь, 1990. – 303 с.
57. Использование онтологий в системах управления знаниями организаций / С.В. Козлов [и др.] // Известия Томского политехнического университета. – 2006. – Т.309. – № 3. – С. 180-184.
58. Кане, М.М. Инструменты менеджмента качества/ М.М. Кане. – СПб: Питер, 2008. – 560 с.
59. Классификация моделей представления знаний [Электронный ресурс] // Портал искусственного интеллекта. – 2009. – Режим доступа: <http://www.aiportal.ru/articles/knowledge-models/classification.html>.
60. Клещев, А.С. Классификация свойств онтологий. Онтологии и их классификации / А.С. Клещев, Е.А. Шалфеева // Научно-техническая информация. Сер. 2. – 2005. – №9. – С. 16-22.

61. Клещев, А.С. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия "онтология" / А.С. Клещев, И.Л. Артемьева // Научно–техническая информация. Сер. 2. – 2001. – № 2. – С. 20-27.
62. Клещев, А.С., Математические модели онтологий предметных областей. Часть 2. Компоненты модели. / А.С. Клещев, И.Л. Артемьева // Научно–техническая информация. Сер. 2. – 2001. – № 3. – С. 19-28.
63. Клещев, А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Часть 3. Сравнение разных классов моделей онтологий. / А.С. Клещев, И.Л. Артемьева // Научно–техническая информация. Сер. 2. – 2001. – № 4. – С. 10-15.
64. Колмогоров, А.Н. Математическая логика / А.Н. Колмогоров, А.Г. Драгалин. – М.: КомКнига, 2006. – 240 с.
65. Краткий словарь терминов современного менеджмента [Электронный ресурс] // «Бизнес Инжиниринг Групп» (БИГ-СПб). – 2002. – Режим доступа: http://bigc.ru/publications/glossary/part11_km.php.
66. Круглов, М.Г. Менеджмент качества как он есть / М.Г. Круглов. – М.: Эксмо, 2007. – 544 с.
67. Кузнецов, И.П. Семантические представления / И.П. Кузнецов. – М.: Наука, 1986. – 296 с.
68. Ландэ, Д.В. Поиск знаний в Internet / Д.В. Ландэ. – М.: Диалектика, 2005. – 272 с.
69. Лapidус, В.А. Всеобщее качество (TQM) в российских компаниях / В.А. Лapidус. – М.: Новости, 2000. – 432 с.
70. Левин, Р. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на Бейсике / Р. Левин, Д. Дранг, Б. Эделсон. – М.: Финансы и статистика, 1990. – 239 с.
71. Липаев, В.В. Документирование сложных программных средств / В.В. Липаев. – М.: СИНТЕГ, 2005. – 216 с.
72. Липаев, В.В. Обеспечение качества программных средств. Методы и стандарты / В.В. Липаев. – М.: СИНТЕГ, 2001. – 380 с.
73. Липаев, В.В. Процессы и стандарты жизненного цикла сложных программных средств / В.В. Липаев. – М.: СИНТЕГ, 2006. – 262 с.
74. Липаев, В.В. Экономика производства сложных программных продуктов / В.В. Липаев. – М.: СИНТЕГ, 2011. – 352 с.
75. Лицензионные данные, определение потребностей в лицензировании [Электронный ресурс] // Microsoft Corporation. – 2010. – Режим доступа: <http://sharepoint.microsoft.com/ru-ru/buy/Pages/Licensing-Details.aspx>.
76. Люггер, Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. 4-е изд.: Пер. с англ / Д.Ф. Люггер. – М.: Издательский дом «Вильяме», 2003. – 864 с.

77. Международный стандарт ISO 9000:2005 Система менеджмента качества. Основные положения и словарь.
78. Международный стандарт ISO 9001:2008 Системы менеджмента качества. Требования.
79. Минский, М. Фреймы для представления знаний: Пер. с англ / М. Минский. – М.: Энергия, 1979. – 152 с.
80. Мирошников, В.В. Онтологическая модель системы управления знаниями в области качества / В.В. Мирошников, Д.И. Булатицкий // Вестник Брянского государственного технического университета. – 2009. – № 4. – С. 100-106.
81. Мирошников, В.В. Система управления знаниями в области качества / В.В. Мирошников, Д.И. Булатицкий // Информационные технологии. – 2006. – №7. – С. 16-22.
82. Назначения, определения и наследование ролей [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/ms414036.aspx>.
83. Никоненко, А.А. Обзор баз знаний онтологического типа [Электронный ресурс] / А.А. Никоненко // Искусственный интеллект №4. – 2009. – Режим доступа: http://www.nbuu.gov.ua/portal/natural/ii/2009_4/4%5C00_Nikonenko_AA.pdf.
84. Нильсон, Н. Принципы искусственного интеллекта / Н. Нильсон. – М.: Радио и связь, 1990. – 376 с.
85. Новичков, А. Система генерации проектной документации Rational SoDA [Электронный ресурс] / А. Новичков // КомпьютерПресс. – 2002. – Режим доступа: <http://www.compress.ru/Article.aspx?id=12028>.
86. Новые возможности корпоративного поиска (SharePoint Server 2010) [Электронный ресурс] // Microsoft Corporation. – 2010. – Режим доступа: <http://technet.microsoft.com/ru-ru/library/ee667266.aspx>.
87. Общие сведения о разработке и настройке рабочих процессов [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: <http://office.microsoft.com/ru-ru/sharepoint-designer-help/HA101859249.aspx>.
88. Окрепилов, В.В. Управление качеством / В.В. Окрепилов. – СПб: Наука, 2000. – 912 с.
89. Онтологии и тезаурусы: модели, инструменты, приложения / Б.В. Добров [и др.]. – М.: Бином. Лаборатория знаний. Интернет-Университет Информационных Технологий, 2009. – 176 с.
90. Онтологический подход к построению базы знаний «Сверхтвердые материалы» / В.Н. Кулаковский [и др.] // Искусственный интеллект. – 2008. – №4. – С. 91-102.
91. Основные понятия представления знаний [Электронный ресурс] // Справочные материалы по информационным технологиям. – Режим

- доступа: <http://itteach.ru/predstavlenie-znaniy/osnovnie-ponyatiya-inzhenerii-znaniy>.
92. Острогорский, М. Разработка технической документации на основе единого источника [Электронный ресурс] / М. Острогорский // «Философт». – 2008. – Режим доступа: <http://www.philosoft.ru/ssth-01.zhtml>.
 93. Острогорский, М. DocBook/XML: открытая платформа для разработки технической документации [Электронный ресурс] / М. Острогорский // «Философт». – 2007. – Режим доступа: <http://www.philosoft.ru/docbook-basics.zhtml>.
 94. Пальчунов, Д.Е. Решение задачи поиска информации на основе онтологий / Д.Е. Пальчунов // Бизнес информатика. – 2008. – № 1. – С. 8-13.
 95. Планирование управления корпоративным контентом (SharePoint Server 2010) [Электронный ресурс] // Microsoft Corporation. – 2010. – Режим доступа: <http://technet.microsoft.com/ru-ru/library/ff602190.aspx>.
 96. Планирование управляемых метаданных (SharePoint Server 2010) [Электронный ресурс] // Microsoft Corporation. – 2010. – Режим доступа: <http://technet.microsoft.com/ru-ru/library/ee530389.aspx>.
 97. Поиск в корпоративной среде SharePoint [Электронный ресурс] // Microsoft Corporation. – 2010. – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/ee556856.aspx>.
 98. Продукты SharePoint [Электронный ресурс] // Microsoft Corporation – 2010. – Режим доступа: <http://technet.microsoft.com/ru-ru/library/ee428287.aspx>.
 99. Разработка онтологии системы управления знаниями организации с использованием методологии концептуального проектирования [Электронный ресурс] // Успехи современного естествознания. – 2006. – Режим доступа: <http://www.rae.ru/use/pdf/2006/9/42.pdf>.
 100. Райли, Д. NGOSS. Построение эффективных систем поддержки и эксплуатации сетей для оператора связи / Д. Райли, М. Кринер. – М.: Альпина Бизнес Букс, 2007. – 192 с.
 101. Расширенная карта процессов оператора связи (eTOM®) [Электронный ресурс] // Jet Info. – 2007. – Режим доступа: http://www.jetinfo.ru/Sites/info/Uploads/2007_7.66B18310AEEF48B386233A330458F8F2.pdf.
 102. РД 50-34.698-90 Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
 103. Резникова, Н.П. Менеджмент в телекоммуникациях / Н.П. Резникова, Е.В. Демина, В.Б. Буогак. – М.: Эко-Трендз, 2005. – 392 с.
 104. Рубашкин, В.Ш. Онтологии: от информационно-поисковых тезаурусов к инженерии знаний / В.Ш. Рубашкин // Труды XX национальной

- конференции по Искусственному Интеллекту с международным участием – КИИ-2006, Т. 2. – М.: Физматлит, 2006.
105. Савельева, Н. Системы управления контентом [Электронный ресурс] / Н. Савельева // «Открытые системы», № 04, 2004. – Режим доступа: <http://www.osp.ru/os/2004/04/184166>.
 106. Соммервилл, И. Инженерия программного обеспечения: 6-е издание / И. Соммервилл. – М.: Вильямс, 2002. – 624 с.
 107. Тельнов, Ю.Ф. Реинжиниринг бизнес-процессов / Ю.Ф. Тельнов. – М.: Финансы и статистика, 2005. – 320 с.
 108. Терехов, А.Н. Технология программирования / А.Н. Терехов. – М.: Бином. Лаборатория знаний. Интернет-Университет Информационных Технологий, 2006. – 148 с.
 109. Тузовский, А.Ф. Архитектура семантического Web-портала / А.Ф. Тузовский // Известия Томского политехнического университета. – 2006. – Т.309. – № 7. – С. 142-145.
 110. Тузовский, А.Ф. Интеллектуальное пространство в системах управления знаниями / А.Ф. Тузовский, В.З. Ямпольский // Известия высших учебных заведений. Физика. – 2004. – № 7. – С. 23-29.
 111. Тузовский, А.Ф. Онтолого-семантические модели в корпоративных системах управления знаниями : дис. ... док. тех. наук : 05.13.01. Томский политехнический университет (ТПУ). – Томск, 2007. – 370 с.
 112. Тузовский, А.Ф. Построение модели знаний организации с использованием системы онтологий / А.Ф. Тузовский, С.В. Козлов // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог 2006». – М.: РГГУ, 2006. – С. 508-512.
 113. Тузовский, А.Ф. Разработка систем управления знаниями на основе единой онтологической базы знаний / А.Ф. Тузовский // Известия Томского политехнического университета. – 2007. – Т.310. – № 2. – С. 182-185.
 114. Тузовский, А.Ф. Системы управления знаниями (методы и технологии) / А.Ф. Тузовский, С.В. Чириков, В.З. Ямпольский. – Томск: Изд-во НТЛ, 2005. – 260 с.
 115. Тузовский, А.Ф. Формирование семантических метаданных для объектов системы управления знаниями / А.Ф. Тузовский // Известия Томского политехнического университета. – 2007. – Т.310. – № 3. – С. 108-112.
 116. Управление знаниями / Э.В. Попов [и др.] // Вестник РФФИ. – 2004. – № 4. – С. 5-29.
 117. Управление знаниями – одна из основных концепций управления [Электронный ресурс] // «Бизнес Инжиниринг Групп» (БИГ-СПб). – 2003. – Режим доступа: http://bigc.ru/publications/bigspb/km/km_main_concept.php.
 118. Управление знаниями: хрестоматия / под ред. Т. Е. Андреевой. – СПб.: Высшая школа менеджмента, 2009. – 514 с.

119. Управление метаданными [Электронный ресурс] // Microsoft Corporation – 2010. – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/ee559337.aspx>.
120. Фатрелл, Р. Управление программными проектами: достижение оптимального качества при минимуме затрат / Р. Фатрелл, Д. Шафер, Л. Шафер. – М.: Вильямс, 2003. – 986 с.
121. Чаадаев, В.К. Бизнес-процессы в компаниях связи / В.К. Чаадаев. – М.: Эко-Трендз, 2004. – 176 с.
122. Чайка, И.И. Стратегия качества и развития организации – производная от её амбиций и внешних условий / И.И. Чайка // Стандарты и качество. – 2002. – №10. – С. 85-87.
123. Aarnes, A. Sharepoint 2010 leads the Magic Quadrant for Enterprise Content Management [Электронный ресурс] / A. Aarnes // Gartner Research. –2010. – Режим доступа: <http://sharepoint247.com/sharepoint2010/sharepoint-2010-leads-the-magic-quadrant-for-enterprise-content-management>.
124. Abecker, A. Handbook on Ontologies / A. Abecker, L. van Elst // Chapter 22: Ontologies for Knowledge Managemen. International Handbooks on Information Systems. – Springer Verlag, 2003 – P. 436-454.
125. Andrews, W. Magic Quadrant for Information Access Technology [Электронный ресурс] / W. Andrews // Gartner Research, 2007. – Режим доступа: http://www.asknet.ru/IS/Analytics/Gartner_MQ.pdf.
126. Application Framework (TAM), Release 4.5 – GB929 [Электронный ресурс] // TeleManagement Forum. – 2011. – Режим доступа: <http://www.tmforum.org/Guidebooks/GB929ApplicationFramework/45327/article.html>.
127. Baril, F. DITA - The mechanics of a single sourcing project [Электронный ресурс] / F. Baril // Proceedings of Extreme Markup Languages. – 2004. – Режим доступа: <http://conferences.idealliance.org/extreme/html/2004/Baril01/EML2004Baril01.html>.
128. Bell, T. Magic Quadrant for Enterprise Content Management [Электронный ресурс] / T. Bell, K. Shegda, M. Gilbert, K. Chin// Gartner Research. – 2010. – Режим доступа: <http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/industryinsights/solutions/pdfs/magic-quadrant-for-enterprise-content-management.pdf>.
129. Best Practices & Standards [Электронный ресурс] // TeleManagement Forum. – 2011. – Режим доступа: <http://www.tmforum.org/BestPracticesStandards/1669/home.html>.
130. Brazhnik, O. Databases and the geometry of knowledge / O. Brazhnik // Data & Knowledge Engineering. – 2007. – V. 61. – № 2. – P. 207-227.
131. Bughin, J. How businesses are using Web 2.0: A McKinsey Global Survey [Электронный ресурс] / J. Bughin, J. Manyika // McKinsey Quarterly. – 2007.

- Режим доступа:
http://www.mckinsey.de/downloads/publikation/mck_on_bt/2007/mobt_12_How_Businesses_are_Using-Web_2_0.pdf.
132. Business Process Framework, Release 8.0 – GB921 [Электронный ресурс] // TeleManagement Forum. – 2008. – Режим доступа: <http://www.tmforum.org/Guidebooks/GB921BusinessProcess/36651/article.html>.
133. Collins, H. Enterprise knowledge portals: next generation portal solutions for dynamic information access, better decision making and maximum results / H. Collins. – New York: AMACOM, 2003. – 222 p.
134. Davenport, T. Working Knowledge: how organizations manage what they know / T. Davenport, L. Prusak. – Boston: Harvard Business School Press, 1998. – 200 p.
135. Day, D. Introduction to the Darwin Information Typing Architecture [Электронный ресурс] / D. Day // IBM Corporation. – 2005. – Режим доступа: <https://www.ibm.com/developerworks/xml/library/x-dita1/>.
136. Enhanced Telecom Operations Map (eTOM) [Электронный ресурс] // ITU. – 2004. – Режим доступа: <http://www.itu.int/rec/T-REC-M.3050.0-200407-S>.
137. European Guide to good Practice in Knowledge Management - Part 1: Knowledge Management Framework [Электронный ресурс] // CEN. – 2004. – Режим доступа: <ftp://cenftp1.cenorm.be/PUBLIC/CWAs/e-Europe/KM/CWA14924-01-2004-Mar.pdf>.
138. European Guide to good Practice in Knowledge Management - Part 3: SME Implementation [Электронный ресурс] // CEN. – 2004. – Режим доступа: <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/KM/CWA14924-03-2004-Mar.pdf>.
139. European Guide to good Practice in Knowledge Management - Part 4: Guidelines for Measuring KM [Электронный ресурс] // CEN. – 2004. – Режим доступа: <ftp://cenftp1.cenorm.be/PUBLIC/CWAs/e-Europe/KM/CWA14924-04-2004-Mar.pdf>.
140. European Guide to good Practice in Knowledge Management - Part 5: KM Terminology [Электронный ресурс] // CEN. – 2004. – Режим доступа: <ftp://cenftp1.cenorm.be/PUBLIC/CWAs/e-Europe/KM/CWA14924-05-2004-Mar.pdf>.
141. Fensel, D. Ontologies: a silver bullet for Knowledge Management and Electronic Commerce / D. Fensel. – Berlin: Springer Verlag, 2001. – 187 p.
142. Gardenfors, P. Conceptual spaces. The geometry of thought / P. Gardenfors. – Cambridge: The MIT Press, 2004. – 307 p.
143. Gilbert, M. Magic Quadrant for Enterprise Content Management [Электронный ресурс] / M. Gilbert, K. Shegda, K. Chin, G. Tay // Gartner Research. – 2011. – Режим доступа: http://kennynguyen.us/wp-content/uploads/2012/04/1_16918_magic_quadrant_for_enterpris_219745.pdf.

144. Gilbert, M. Magic Quadrant for Enterprise Content Management [Электронный ресурс] / M. Gilbert, K. Shegda, K. Chin, G. Tay, H. Koehler-Kruener // Gartner Research. – 2012. – Режим доступа: <http://www.webcitation.org/6IPZcp4qk>.
145. Gomez-Perez, A. Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web / A. Gomez-Perez. – Springer Verlag, 2002. – 402 p.
146. Gruber, T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing / T.R. Gruber // International Journal Human-Computer Studies – Vol. 43, Issues 5-6. – 1995. – P.907–928.
147. Gruber, T.R. Ontology to appear in the Encyclopedia of Database Systems [Электронный ресурс] / T.R. Gruber // Ling Liu and M. Tamer Özsu (Eds.). Springer-Verlag. – 2008. – Режим доступа: <http://tomgruber.org/writing/ontologydefinition-2007.htm>.
148. IDEF5 Ontology Description Capture Method [Электронный ресурс] // Knowledge Based Systems, Inc. – Режим доступа: <http://idef.com/IDEF5.htm>.
149. ISO/IEC 19506 Information technology - Object Management Group Architecture-Driven Modernization (ADM) - Knowledge Discovery Meta-Model (KDM) systems [Электронный ресурс] // ISO/IEC. – 2012. – С. 331. – Режим доступа: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=35&ics2=080&ics3=&csnumber=32625.
150. ISO 15489-1 Information and documentation - Records management [Электронный ресурс] // ISO. – 2001. – 19 p. – Режим доступа: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=31908.
151. ISO 23081-2 Information and documentation - Managing metadata for records - Part 2: Conceptual and implementation issues [Электронный ресурс] // ISO. – 2009. – 33 p. – Режим доступа: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=31908.
152. ISO 26162 Systems to manage terminology, knowledge and content - Design, implementation and maintenance of terminology management systems [Электронный ресурс] // ISO. – 2012. – 69 p. – Режим доступа: http://www.iso.org/iso/catalogue_detail.htm?csnumber=43427.
153. ISO/IEC 27001:2013 Information technology - Security techniques - Information security management systems – Requirements [Электронный ресурс] // ISO. – 2013. – 23 p. – Режим доступа: http://www.iso.org/iso/ru/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54534.
154. ISO/IEC 27002:2013 Information technology - Security techniques - Code of practice for information security controls [Электронный ресурс] // ISO. –

2013. – 80 р. – Режим доступа: http://www.iso.org/iso/ru/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54533.
155. Kühn, O. Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges [Электронный ресурс] / O. Kühn, A. Abecker // Journal of Universal Computer Science. – 1998. – Режим доступа: http://www.jucs.org/jucs_3_8/corporate_memories_for_knowledge/Kuehn_O.pdf.
156. Lists.GetListItems Method [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: <http://msdn.microsoft.com/en-us/library/lists.lists.getlistitems.aspx>.
157. Maarten van Engers. Knowledge Management: The Role of Mental Models in Business Systems Design. PhD thesis. [Электронный ресурс] // Vrije Universiteit, Amsterdam, 2001. – Режим доступа: http://www.leibnizcenter.org/docs/vanengers/PhDThesis_T_M_van_Engers.pdf.
158. Maier, R. Knowledge management systems: Information and communication technologies for knowledge management / R. Maier. – Berlin Heidelberg: Springer Verlag, 2004. – 635 p.
159. McGuinness, D.L. Ontologies Come of Age [Электронный ресурс] D. McGuinness, Fensel D., Hendler J. et al. (eds.), Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press. – 2003. – Режим доступа: [http://www.ksl.stanford.edu/people/dlm/papers/ontologies-come-of-age-mit-press-\(with-citation\).htm](http://www.ksl.stanford.edu/people/dlm/papers/ontologies-come-of-age-mit-press-(with-citation).htm).
160. Mescan, S. Single-Source Content Management Can Take You Further [Электронный ресурс] / Mescan S. // Techwhirl: The Internet forum for technical communicators. – 2005. – Режим доступа: <http://www.vasont.com/vasont/literature/2005-03TECHWR-L.pdf>.
161. Microsoft Enterprise Managed Metadata Web Service Protocol Specification [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: [http://msdn.microsoft.com/en-us/library/dd962601\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/dd962601(v=office.12).aspx).
162. Noy, N.F. Ontology Development 101: A Guide to Creating Your First Ontology [Электронный ресурс] / N.F. Noy, D.L. McGuinness // Stanford University, Stanford, 2001. – Режим доступа: http://protege.stanford.edu/publications/ontology_development/ontology101-old.html.
163. Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces [Электронный ресурс] // Oracle. – 2011. – Режим доступа: http://download.oracle.com/docs/cd/E21764_01/webcenter.1111/e10149/toc.htm.
164. Paulk, M.C. A History of the Capability Maturity Model for Software / M.C. Paulk // ASQ Software Quality Professional. Vol. 12. №. 1, 2009. – P. 5-19.

165. Paulk, M.C. How ISO 9001 Compares With the CMM / M.C. Paulk // IEEE Software. – 1995. – Vol. 12. – № 1. – P. 74-83.
166. Paulk, M.C. The Capability Maturity Model: Guidelines for Improving the Software Process / Paulk M.C. [et al]. – Addison-Wesley, 1995. – 442 p.
167. Process Classification Framework (PCF) - Cross Industry [Электронный ресурс] // APQC's. – 2014. – 26 p. – Режим доступа: http://www.apqc.org/knowledge-base/download/268712/a%3A1%3A%7Bi%3A1%3Bs%3A1%3A%222%22%3B%7D/K03785_PCF_Cross%20Industry_v6_201_1.pdf.
168. Sowa, J.F. Knowledge representation: logical, philosophical and computational foundations / J.F. Sowa. – CA: Brooks Cole Publishing Co, 2000. – 512 p.
169. Staab, S. Knowledge Portals: Ontologies at Work / S. Staab, A. Maedche // AI Magazine. – 2001. – Vol. 22. – №2. – P. 63–75.
170. TaxonomyClientService.GetChildTermsInTerm Method [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.taxonomy.webservices.taxonomyclientservice.getchildtermsinterm.aspx>.
171. TaxonomyClientService.GetChildTermsInTermSet Method [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.taxonomy.webservices.taxonomyclientservice.getchildtermsintermset.aspx>.
172. The Knowledge Management Process: a Practical Approach. – IDC. – 2000.
173. The Knowledge Management Scenario: Trends and Directions for 1998–2003. – Gartner Group. – 1999.
174. TickIT Shema [Электронный ресурс] // BSI Group. – 2009. – Режим доступа: <http://www.tickit.org/scheme.htm>.
175. Uschold, M. Ontologies: principles, methods and applications / M. Uschold, M. Gruninger // Knowledge Engineering Review. – 1996. – Vol. 11. – № 2. – P. 93–113.
176. UserGroup.GetGroupCollectionFromUser Method [Электронный ресурс] // Microsoft Corporation. – 2011. – Режим доступа: <http://msdn.microsoft.com/en-us/library/ms772552.aspx>.
177. Van Heijst G. Using Explicit Ontologies in KBS Development / G. Van Heijst, A.T. Schreiber, B.J. Wielinga // International Journal of Human and Computer Studies. – 1996. – №46 (2-3). – P. 183–292.
178. Van Loon H. Process Assessment and ISO/IEC 15504: a Reference Book / H. Van Loon. – New York: Springer, 2007. – 280 p.

179. Weber, R. Knowledge-based knowledge management / R. Weber, R. Kaplan // Innovations in Knowledge Engineering (ed. Faucher C. et al.). – Heidelberg: Physica-Verlag, 2003. – P. 125–143.

Приложение А

Описание концептов онтологии документации на программные средства

Описание концептов (типов контента) онтологии документации на программные средства приведено с описанием атрибутов родительского типа контента «Модуль документации».

1. Настраиваемый параметр

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Номер параметра	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

2. Назначение подсистемы

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский
4.	Шифр	-	-	-

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

3. Назначение продукта

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Продукт	Код продукта из списка «Продукты»	Внешние данные (External Data)	-

4. Назначение группы продуктов

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Группа продуктов	Код группы продуктов из списка «Группы продуктов»	Внешние данные (External Data)	-

5. Назначение решения

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Решение	Код решения из списка «Решения»	Внешние данные (External Data)	-

6. Назначение комплекса

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Комплекс	Код комплекса из списка «Комплексы»	Внешние данные (External Data)	-

7. Назначение опции

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Опция	Код опции из списка «Опции»	Внешние данные (External Data)	-

8. Процедура

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Наименование процедуры	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

9. Функция

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Наименование функции	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

10. Сообщение об ошибке

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Номер сообщения	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

11. МПВ-файл подсистемы

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

12. Роль

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Наименование роли	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

13. Возможность

Имеет только атрибуты родительского типа контента «Модуль документации».

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-

14. Реализация возможности

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

15. Описание операции

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

16. Краткое описание автономной формы

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Код автономной формы	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

17. Описание автономной формы

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Код автономной формы	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

18. Настройки автономной формы

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Код автономной формы	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

19. Интеграция автономной формы

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Код автономной формы	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-
8.	WebX	Версия интегратора	Выбор (Choice)	1, 2, 3, 4, 5

20. Не определен

Имеет только атрибуты родительского типа контента «Модуль документации».

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	-	-	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-

Приложение Б – Схемы eTOM и TAM концепции NGOSS

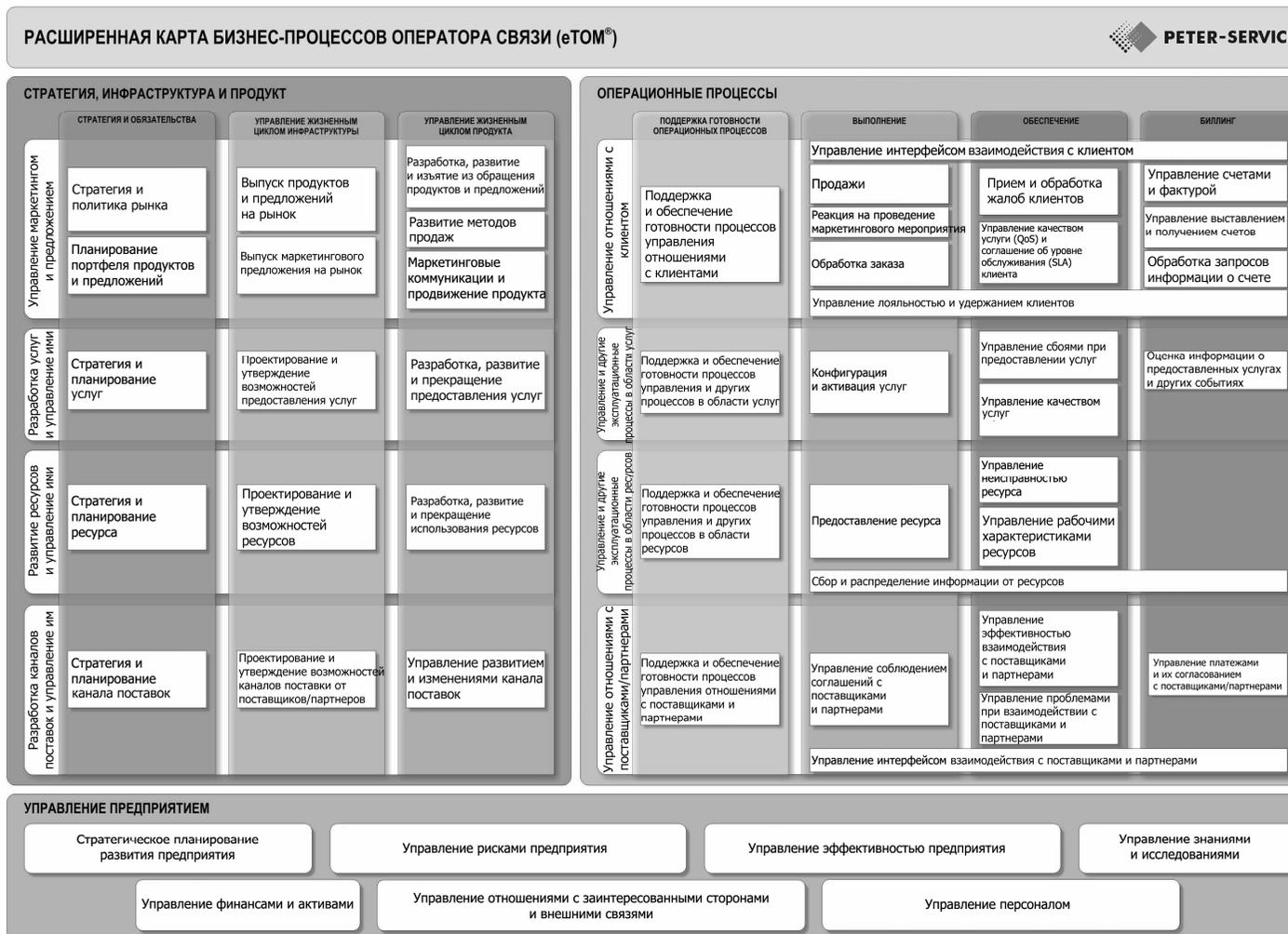


Рисунок Б.1 – Схема eTOM уровня 2

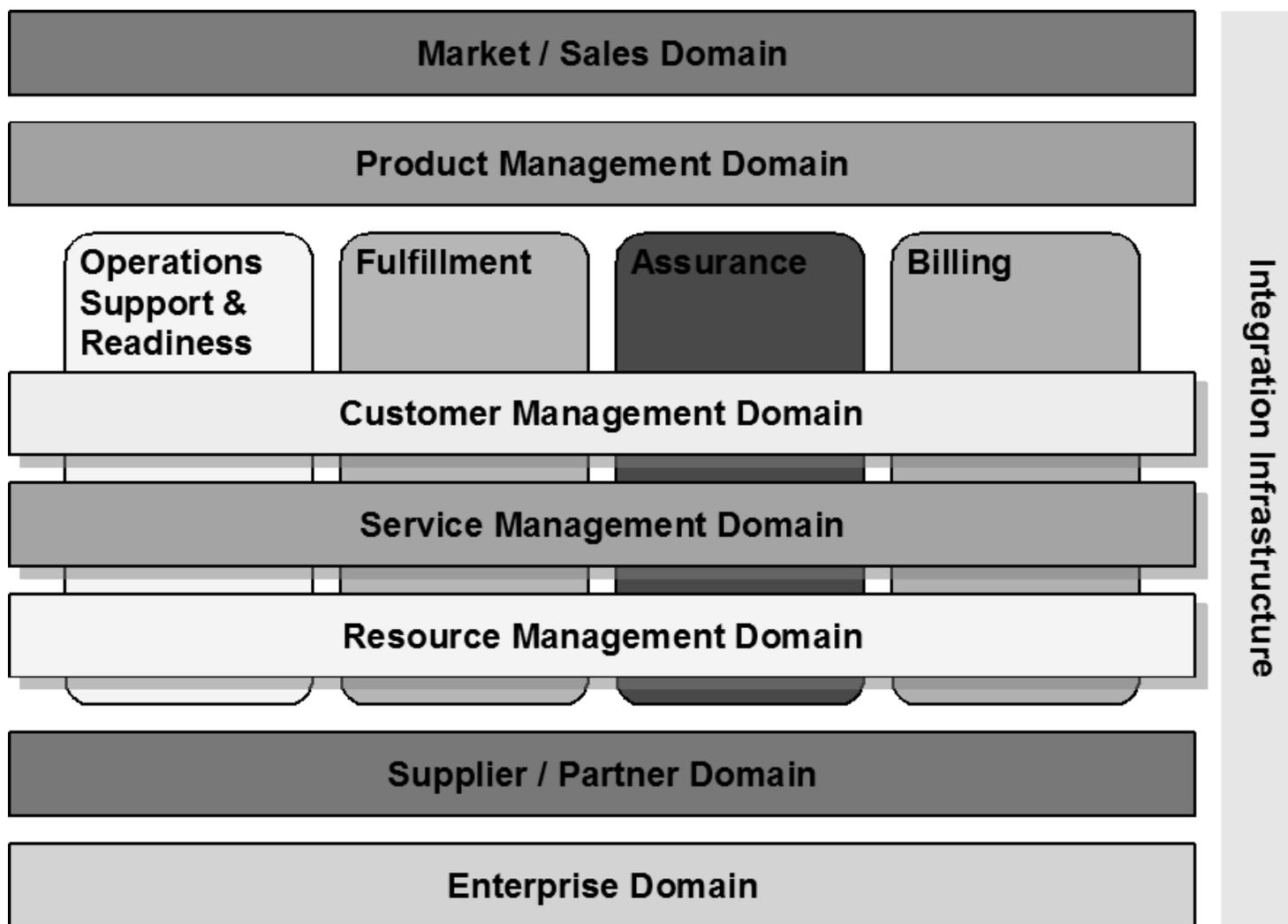


Рисунок Б.2 – Схема ТАМ уровня 0

Приложение В – Описание концептов онтологии продукции

1. Подсистема (список «Подсистемы»)

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название подсистемы на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название подсистемы на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная подсистеме	Строка (Single line of text)	-
4.	Регистрационный номер	Номер подсистемы, состоящий из пяти цифровых знаков	Строка (Single line of text)	00000÷99999
5.	Владелец подсистемы	Сотрудник ответственный за подсистему	Строка (Single line of text)	-
6.	Продукты	Продукты, в состав которых входит подсистема	Внешние данные (External Data)	-
7.	Опция	Опция, в состав которой входит подсистема	Внешние данные (External Data)	-
8.	Платформы	Коды операционных систем, на платформе которых проводилось тестирование подсистем	Многострочный текст (Multiple lines of text)	
9.	Тип подсистемы	Тип подсистемы в соответствии с правилами классификации	Выбор (Choice)	Универсальная; специализированная; регистрационный ключ
10.	Языки локализации	Языки локализации документации	Выбор (Choice)	Русский; английский; испанский
11.	Языки интерфейса	Существование интерфейса пользователя на указанных языках	Выбор (Choice)	Русский; английский; испанский

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
12.	Статус поддержки подсистемы	Указывается текущий статус подсистемы	Выбор (Choice)	«Да»; при прекращении поставки и технической поддержки – «Нет»; при частичном снятии с поддержки – «Частично»
13.	Примечание	Дополнительные сведения о подсистеме	Строка (Single line of text)	-

2. Опция (список «Опции»)

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название опции на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название опции на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная опции	Строка (Single line of text)	-
4.	Продукт	Код продукта, в состав которого входит опция	Внешние данные (External Data)	-
5.	Статус использования	Указывается текущий статус опции	Выбор (Choice)	Используется; Не используется
6.	Примечание	Дополнительные сведения об опции	Строка (Single line of text)	-

3. Продукт (список «Продукты»)

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название продукта на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название продукта на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная продукту	Строка (Single line of text)	-
4.	Регистрационный номер	Номер продукта в группе	Строка (Single line of text)	

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
5.	Группы продуктов	Коды групп продуктов, в состав которых входит продукт	Внешние данные (External Data)	-
6.	Статус поддержки	Текущий статус технической поддержки продукта	Выбор (Choice)	«Да»; «Нет»; «Частично», если осуществляется только техническое сопровождение продукта, без модификации функций.
7.	Языки локализации	Название продукта на русском языке	Выбор (Choice)	Русский; английский; испанский
8.	Планируемая локализация	Название продукта на английском языке	Выбор (Choice)	Русский; английский; испанский
9.	Примечание	Дополнительные сведения о продукте	Строка (Single line of text)	-

4. Группа продуктов (список «Группы продуктов»)

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название группы продуктов на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название группы продуктов на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная группе продуктов	Строка (Single line of text)	-
4.	Регистрационный номер	Номер группы продуктов, состоящий из пяти цифровых знаков	Строка (Single line of text)	00000÷00099
5.	Статус использования	Указывается текущий статус группы продуктов	Выбор (Choice)	Используется; Не используется
6.	Примечание	Дополнительные сведения о группе продуктов	Строка (Single line of text)	-

5. Решение (список «Решения»)

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название решения на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название решения на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная решению	Строка (Single line of text)	-
4.	Регистрационный номер	Номер решения, состоящий из пяти цифровых знаков	Строка (Single line of text)	00000÷00999
5.	Группы продуктов	Коды групп продуктов, входящие в решение	Внешние данные (External Data)	-
6.	Статус использования	Указывается текущий статус решения	Выбор (Choice)	Используется; Не используется
7.	Примечание	Дополнительные сведения о решении	Строка (Single line of text)	-

6. Комплекс (список «Комплексы»)

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название комплекса на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название комплекса на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная комплексу	Строка (Single line of text)	-
4.	Подсистемы	Коды подсистем, входящие в комплекс	Внешние данные (External Data)	-
5.	Регистрационный номер	Номер комплекса, состоящий из пяти цифровых знаков	Строка (Single line of text)	00000÷09999
6.	Статус использования	Указывается текущий статус комплекса	Выбор (Choice)	Используется; Не используется
7.	Примечание	Дополнительные сведения о комплексе	Строка (Single line of text)	-

Приложение Г – Стандарт организации «Система управления
знаниями о программной продукции»

СТО 00000-DD-MM-YYYY

СТАНДАРТ ОРГАНИЗАЦИИ

Система менеджмента качества

**СИСТЕМА УПРАВЛЕНИЯ ЗНАНИЯМИ
О ПРОГРАММНОЙ ПРОДУКЦИИ**

Наименование организации

Предисловие

1 РАЗРАБОТАН И ВНЕСЕН

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Приказом Генерального директора от DD.ММ.YYYY. №XX

3 ВВЕДЕН ВПЕРВЫЕ

СТАНДАРТ ОРГАНИЗАЦИИ
Система менеджмента качества

СИСТЕМА УПРАВЛЕНИЯ ЗНАНИЯМИ О ПРОГРАММНОЙ ПРОДУКЦИИ

Дата введения YYYY-MM-DD

1. Область применения

Настоящий стандарт устанавливает правила работы с системой управления знаниями о программной продукции.

Настоящий стандарт распространяется на деятельность сотрудников Дирекции по разработке программного обеспечения, осуществляющих разработку, оформление и сопровождение документации на программные средства, постановочные работы, тестирование.

Настоящий документ является справочными для сотрудников групп, занимающихся контрольной инсталляцией при проведении завершающих испытаний, поставкой и сопровождением поставляемой продукции.

2. Определения, обозначения и сокращения

ПС – программное средства.

Документация – печатные и электронные руководства, справочная система, справочный текст, описывающие как пользоваться программным продуктом.

МД – модуль документации. Дискретный фрагмент текста документа в электронном формате, который может содержать как изображения, так и текст с форматированием и гиперссылками. Из наборов модулей могут формироваться конкретные электронные копии.

СУЗ – система управления знаниями о программной продукции, которая поддерживает процессы работы со знаниями/информацией, находящимися в документации на ПС.

ФС – функциональная спецификация.

ТС – техническая спецификация.

3. Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

СТО 00001-YYYY Система менеджмента качества. Постановка. Требования к процессу.

СТО 00002-YYYY Система менеджмента качества. Документация на поставляемую продукцию. Требования к оформлению, содержанию и переводу.

СТО 00003-YYYY Система менеджмента качества. Тестирование выпускаемой продукции. Требования к процессу.

СТО 00004-YYYY Система менеджмента качества. Правила идентификации выпускаемой продукции. Общие положения.

СТО 00005-YYYY Система менеджмента качества. Документация на поставляемую продукцию. Порядок регистрации, обозначения и фиксации.

СТО 00006-YYYY Система менеджмента качества. Непрерывное усовершенствование. Требования к процессу.

4. Общие положения

Основным ресурсом знаний СУЗ является документация на ПС.

Под знаниями понимается информация, подвергаясь преобразованию в части выделения понятий некоторой предметной области, свойств и связей между ними.

Для решения задачи поиска информации соединены два подхода: использование поисковой системы, обеспечивающей релевантную отработку формального запроса, и интерфейса Web-каталога, обеспечивающего пользователю понятный и удобный интерфейс.

Пертигентность информационного поиска достигается за счёт спецификации не только определенной рубрики, в которой ищется информация, но и вида требуемого МД.

Поиск выполняется в экземплярах базы знаний (МД) с учетом их семантических свойств и связей. Результатом такого поиска являются знания, которые используются эксплуатирующим систему специалистом.

С точки зрения производства и сопровождения обеспечивается:

- повышение качества базовых процессов жизненного цикла программных средств;
- повышение качества документации (гипертекст, метаинформация, автоматическое формирование документов);
- повышение качества процесса поиска информации о программной продукции;
- активизация обмена знаниями между сотрудниками;
- уменьшение трудозатрат на разработку документации на ПС;
- уменьшение трудозатрат на сопровождение ПС.

С точки зрения потребителя (заказчика), при предоставлении доступа к portalу извне:

- повышение качества процесса поиска информации о продукции;
- получение всевозможных справочных документов по различным информационным срезам, создание информационных витрин, визуализации информации о продукции;
- единое информационное пространство для сведений о продукции;
- упрощенное, адресное формирование рекламаций по документации.

Система имеет Web-интерфейс, реализованный на базе отдельной коллекции Microsoft SharePoint.

Раздел «Онтологии документации на ПС» предназначен для ведения базы МД и работы документаторов. В разделе реализованы механизмы создания и атрибутирования метаинформацией МД. Содержание и оформление МД регламентируется СТО 00002-YYYY. Порядок проведения тестирования МД регламентирован СТО 00003-YYYY.

Раздел «Документы» предназначен для регистрации экземпляров документов на разрабатываемую продукцию (идентификации документа определенного вида на единицу продукции). Регистрация документации регламентирована СТО 00005-YYYY.

В разделе «Система управления знаниями» реализованы интерфейсы для просмотра и поиска информации по документации. В его основе – система рубрик, соответствующая иерархии онтологии функциональностей и онтологии продукции. Пользователь выбирает определённую рубрику и получает список МД, который ей соответствуют. Для наиболее полной и точной формализации информационной потребности пользователь может уточнить вид МД и указать дополнительные слова, которые должны присутствовать в требуемых МД или его атрибутах. При этом результатом поиска являются МД, для которых дополнительно автоматически добавляются связи с другими МД.

Раздел «Функциональные возможности» предназначен для ведения терминов управляемых метаданных – функциональных возможностей. Порядок регистрации регламентирован СТО 00001-YYYY.

Система поддерживается для двух языков локализации – русский и английский.

5. Роли и доступность операций

Базовые роли пользователей системы и основные операции, которые доступны в рамках выбранной роли.

Роль «Полный доступ»

Пользователь/группа пользователей: инженер по знаниям (документатор-эксперт).

Доступно:

- все операции.

Роль «Группа документации»

Пользователь/группа пользователей: сотрудники группы документации.

Доступно:

- добавление, изменение и удаление МД в онтологии документации на ПС;
- регистрация документов в списке «Документы».

Роль «Доступ»

Пользователь/группа пользователей: руководители проектов.

Доступно:

- добавление, изменение и удаление экземпляров в списках подсистемы разграничения доступа.

Роль «Продукция»

Пользователь/группа пользователей: менеджеры групп продуктов.

Доступно:

- добавление, изменение и удаление экземпляров онтологии продукции.

Роль «Предприятие» (внутренний пользователь)

Доступно:

- работа с представлением информации;
- поиск;
- формирование документов.

Роль «Предприятие N» (внешний пользователь)

Доступно:

- работа с представлением информации, доступ к которой открыт;
- поиск;
- формирование документов.

6. Операции

Работа с представлением информации

Предусмотрены два дерева предопределенных поисковых запросов: функциональных возможностей ПС и продукции.

После того, как пользователь выбирает определенную рубрику, связанный с данной рубрикой МД с типом контента «Возможность» или «Назначение ...» автоматически отображается на экране. Дополнительно выводятся ссылки на все МД, связанные с данной рубрикой и группировкой по виду МД.

Пользователь может перейти по ссылке на описание интересующего его МД. Также пользователю предоставляется возможность получить документ с текстом МД, всеми связанными МД и вывести его на печать. Таким образом, пользователь получает полное описание выбранной функциональной возможности или экземпляра продукции.

Поиск

Подсистема поиска использует штатный механизм поиска, реализованный в продукте Microsoft SharePoint.

Алгоритм поиска построен на основе двух подходов:

- поиска по ключевым словам по всем экземплярам онтологии документации на ПС с учетом их семантических свойств и связей;
- ограничения объема ресурсов при выборе определенной рубрики онтологии функциональностей или онтологии продукции и поиск в более низких рубриках.

Формирование документов

Формирование (компоновка) цельных документов производится на основе предопределенных шаблонов документов. Для каждого вида документа на ПС создается свой шаблон.

Необходимые для формирования документы задаются в списке «Документы», в котором определяется соответствие между экземпляром онтологии продукции и видом документа на ПС.

Работа с онтологией продукции

Работа с онтологией продукции выполняется в разделе «Продукция». Создание экземпляров онтологии продукции и формирование отношений между ними выполняется менеджерами групп продуктов. Порядок регистрации экземпляров продукции регламентирован СТО 00004-YYYY.

Работа со списками разграничения доступа

В основе подсистемы разграничения доступа лежит штатный механизм разграничения доступа Microsoft SharePoint. Определенным пользователям или группам пользователей назначаются роли, разрешающие доступ к объектам SharePoint.

Для организации дополнительного механизма разграничения доступа к ресурсам для потребителей созданы списки SharePoint «Заказчики» и «Продукты заказчиков».

В списке «Заказчики» задается соответствие между наименованием заказчика и его группой пользователей в SharePoint. В списке «Продукты заказчиков» задается соответствие между наименованием заказчика и продуктами, доступ к которым ему разрешен.

С целью ознакомления заказчиков с ПС, доступ к которому закрыт, создан список «Исключения». В данном списке задаются виды МД, доступ к которым разрешен, например, «Назначение продукта», «Назначение подсистемы» и т.д.

Ведение списков «Заказчики» и «Продукты заказчиков» выполняется руководителями проектов. Ведение списка «Исключения» выполняется инженером по знаниям.

Регистрация документа в списке «Документы»

Список «Документы» предназначен для регистрации экземпляров документов на поставляемую продукцию. Регистрация документации регламентирована СТО 00005-YYYY.

Зарегистрированные в списке документы становятся доступными для формирования в СУЗ.

Работа с МД

Библиотека «Онтологии документации на ПС» является ядром системы. В библиотеке сохраняются все существующие и вновь создаваемые МД. Атрибуты являются метаинформацией для МД. Типы контента определяют виды МД. Каждому типу контента соответствует свой набор атрибутов. Содержание и оформление МД регламентируется СТО 00002-YYYY. Порядок проведения тестирования МД регламентирован СТО 00003-YYYY.

Регистрация МД в библиотеке выполняется с обязательным определением семантических свойств и связей.

Порядок действий:

- выбрать папку хранения модуля (соответствует структуре онтологии продукции);
- загрузить модуль;
- выбрать вид МД (будут подгружены атрибуты соответствующие выбранному типу);
- заполнить атрибуты модуля.

Регистрация новой функциональной возможности

Для регистрации новой функциональной возможности создается запрос на инженера по знаниям. В запросе обязательно указывается родительская возможность, наименование новой возможности и её перевод, а также ссылка на ФС или ТС, в рамках которых вводится функциональная возможность.

Порядок действий при регистрации возможности в банке терминов:

- Действия сайта -> Параметры сайта -> Управление банком терминов;
- выбрать дерево тегов «Возможности», найти родительский узел возможности;
- вызвать контекстное меню «Создать термин»;
- ввести наименование возможности на английском языке, а затем на русском (смена языка производится в поле «Язык»).

Регистрация нового вида МД и добавление/изменение атрибутов

Для регистрации нового вида МД или добавления/изменения атрибутов существующих видов МД создается запрос на инженера по знаниям. Запрос рассматривается в рабочем порядке. При положительном результате рассмотрения инженер по знаниям вносит изменения в библиотеку документации на ПС.

7. Внесение изменений в структуру СУЗ

Внесение изменений в структуру СУЗ инициируется в порядке непрерывного совершенствования в соответствии с СТО 00006-YYYY.

Для регистрации ошибок и предложений по модификации СУЗ используется запрос на инженера по знаниям.

При изменении требований настоящего стандарта все необходимые изменения в СУЗ осуществляются поддерживающим подразделением (группой документации), за исключением работ, требующих привлечения информационно-технического отдела.

8. Средства и инструменты

Microsoft SharePoint.

9. Распределение

Администрация;

Дирекция по разработке программного обеспечения;

Дирекция по работе с клиентами;

Информационно-технический отдел.

ПРИЛОЖЕНИЕ А

(справочное)

В настоящем приложении приведена модель СУЗ (рисунок А.1). Основными компонентами СУЗ являются:

- онтология информационной системы (используется для описания информационных ресурсов с учетом семантики обрабатываемой информации);
- подсистема поиска (выполняет поиск информации по запросам, навигацию, рубрицирование и визуализацию экземпляров онтологии информационной системы, обеспечивает обмен знаниями между пользователями СУЗПП при помощи механизмов комментирования и тегирования МД);
- подсистема компоновки документов (выполняет формирование требуемых цельных документов в определенных форматах);
- подсистема разграничения доступа (обеспечивает разграничение доступа к информационным ресурсам);
- подсистема целостности онтологии (обеспечивает целостность онтологии).

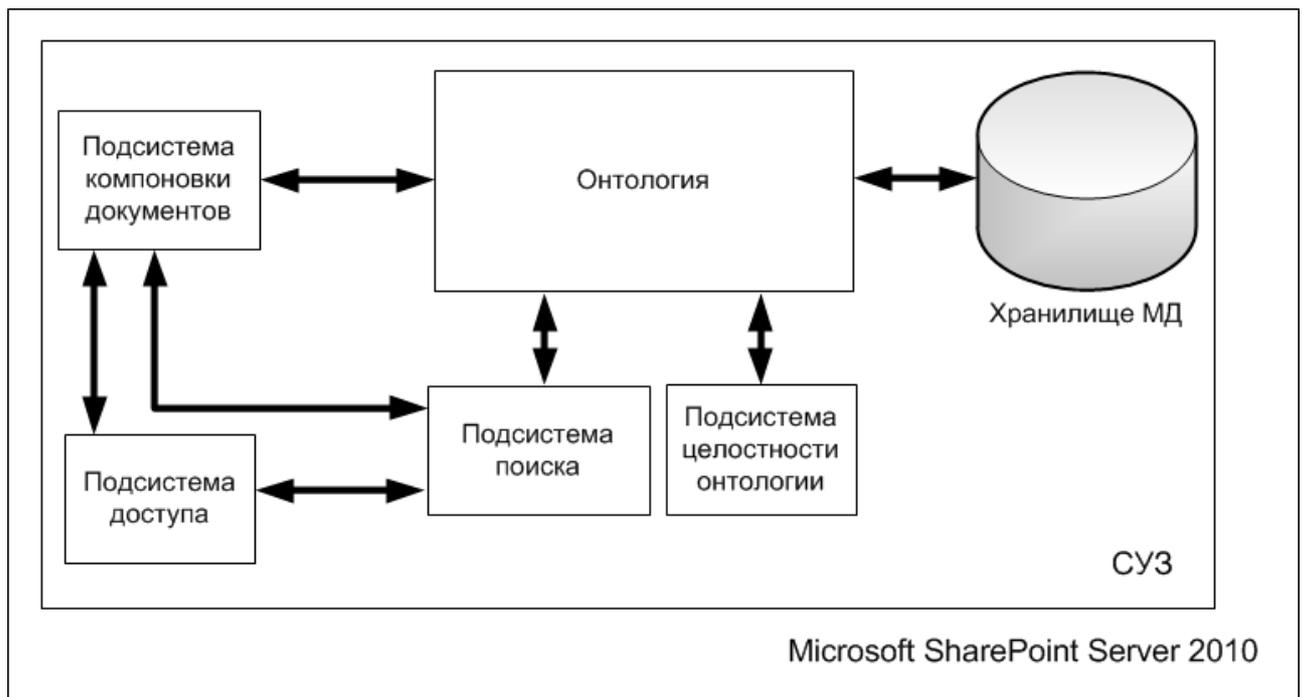


Рисунок А.1 – Модель СУЗ

ПРИЛОЖЕНИЕ Б

(справочное)

В приложении приведено описание типов контента (концептов) онтологии документации на ПС. Описание приведено с описанием атрибутов родительского типа контента «Модуль документации».

Таблица Б.1 – Настраиваемый параметр

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Name	Наименование файла МД	Файл (File)	-
2.	Title	Название МД	Строка (Single line of text)	-
3.	Язык	Язык локализации МД	Выбор (Choice)	Русский; английский; испанский
4.	Шифр	Номер параметра	Строка (Single line of text)	-
5.	Функциональная возможность	Наименование функциональной возможности	Управляемые метаданные (Managed Metadata)	-
6.	Связь с другим МД	Ссылка на связанный МД	Многострочный текст (Multiple lines of text)	-
7.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-

Приводится описание всех типов контента.

ПРИЛОЖЕНИЕ В

(справочное)

В приложении приведено описание концептов онтологии продукции.

Таблица В.1 – Подсистема

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Наименование	Название подсистемы на русском языке	Строка (Single line of text)	-
2.	Наименование английское	Название подсистемы на английском языке	Строка (Single line of text)	-
3.	Код	Техническое наименование (аббревиатура), присвоенная подсистеме	Строка (Single line of text)	-
4.	Регистрационный номер	Номер подсистемы, состоящий из пяти цифровых знаков	Строка (Single line of text)	00000÷99999
5.	Владелец подсистемы	Сотрудник ответственный за подсистему	Строка (Single line of text)	-
6.	Продукты	Продукты, в состав которых входит подсистема	Внешние данные (External Data)	-
7.	Опция	Опция, в состав которой входит подсистема	Внешние данные (External Data)	-
8.	Платформы	Коды операционных систем, на платформе которых проводилось тестирование подсистем	Многострочный текст (Multiple lines of text)	
9.	Тип подсистемы	Тип подсистемы в соответствии с правилами классификации	Выбор (Choice)	Универсальная; специализированная; регистрационный ключ
10.	Языки локализации	Языки локализации документации	Выбор (Choice)	Русский; английский; испанский
11.	Языки интерфейса	Существование интерфейса пользователя на указанных языках	Выбор (Choice)	Русский; английский; испанский

Приводится описание всех концептов онтологии продукции.

ПРИЛОЖЕНИЕ Г

(справочное)

В таблице Г.1 приведено описание атрибутов списка «Документы».

Таблица Г.1 – Список «Документы»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Код файла документа	Наименование файла документа	Строка (Single line of text)	-
2.	Вид документа	Наименование вида документа	Выбор (Choice)	«Руководство оператора», «Руководство системного программиста», «Описание комплекса» и т.д.
3.	Подсистема	Код подсистемы из списка «Подсистемы»	Внешние данные (External Data)	-
4.	Продукт	Код продукта из списка «Продукты»	Внешние данные (External Data)	-
5.	Группа продуктов	Код группы продуктов из списка «Группы продуктов»	Внешние данные (External Data)	-
6.	Решение	Код решения из списка «Решения»	Внешние данные (External Data)	-

ПРИЛОЖЕНИЕ Д

(справочное)

В приложении приведено описание структуры списков подсистемы разграничения доступа.

Таблица Д.1 – Список «Заказчики»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Заказчик	Наименование заказчика	Строка (Single line of text)	-
2.	Роль	Наименование роли	Пользователи и группы (Person or Group)	Группы (Group)

Таблица Д.2 – Список «Продукты заказчиков»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Заказчик	Наименование заказчика	Строка (Single line of text)	-
2.	Продукт	Код продукта из списка «Продукты»	Внешние данные (External Data)	-
3.	Статус ограничения	Указывается текущий статус ограничения	Выбор (Choice)	Используется; Не используется

Таблица Д.3 – Список «Исключения»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Вид МД	Тип контента	Строка (Single line of text)	-

Приложение Д – Примеры реализации обращений к Web-службам Microsoft SharePoint

1. Пример обращения к Web-службе управляемых метаданных (TaxonomyClientService): метод GetChildTermsInTermSet

```
// внешние библиотеки
<script type="text/javascript" src="../../jquery-1.6.1.min.js"></script>
<script src='../Tree/dynatree-0.5.4/jquery/ui.core.js'
type='text/javascript'></script>
<script src='../Tree/dynatree-0.5.4/jquery/jquery.cookie.js'
type='text/javascript'></script>
<link href='../Tree/dynatree-0.5.4/src/skin/ui.dynatree.css' rel='stylesheet'
type='text/css'>
<script src='../Tree/dynatree-0.5.4/src/jquery.dynatree.js'
type='text/javascript'></script>
<script type="text/javascript" src="../../jquery.SPServices-0.6.1.min.js"></script>
<script type="text/javascript" src="../../sessvars.js"></script>

<script type='text/javascript'>

// функция построения дерева первого уровня (GetChildTermsInTermSet)
function SP_fl(lang) {
    var terms = new Array();
    var terms_name = new Array();
    var terms_id = new Array();
    var tree_obj = new Array();
    var lcid;
    if (lang=='RUS') {lcid=1049} else {lcid=1033};
    $.SPServices({
        operation: "GetChildTermsInTermSet",
        sspId: "0f646e47-bf80-4cc3-a89a-9717f1bb5d57", //Справочники документации
        lcid: lcid, //1033 - ENU, 1049 - RUS
        termSetId: "212ebc29-c33e-407b-9c82-58b58a8b91fc", //Возможности
        completefunc: function (xData, Status) {
            var result = xData.responseXML.text;
            if (window.DOMParser) {
                parser=new DOMParser();
                xmlDoc=parser.parseFromString(result,"text/xml");
            }
            else { // Internet Explorer
                xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
                xmlDoc.async="false";
                xmlDoc.loadXML(result);
            }
            terms = xmlDoc.getElementsByTagName("T");
            for(var j=0; j<terms.length; j++)
            {
                terms_id[j] =terms[j].attributes[0].value;
                terms_name[j] = terms[j].childNodes(0).childNodes(0).getAttribute("a32");
            }
            // обработка терминов
            if (terms.length>0)
            {
                for (var j=0; j<terms.length; j++)
                {
                    tree_obj[tree_obj.length] = { title: terms_name[j] , key: terms_id[j] ,
                    tooltip: 'Возможность - '+ terms_name[j] , "isFolder": false, isLazy: true, checkbox:
                    true,
                    url: "view_func.html?var1="+terms_name[j]
                    +";var2="+lang+";var3="+terms_id[j]};
                }
                $("#tree").dynatree("getRoot").addChild(tree_obj);
                $("#divImage").html("").append('<br/>');
            }
        }
    });
}

```

```

    else
    {
        $("#divId").append('<font style="font-family:Arial; font-size:14pt; font-weight: bold;">Данные не найдены.</font>');
        $("#divImage").html("").append('<br/>');
    }
}
});
}
</script>

```

Пример SOAP-ответа Web-службы:

```

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetChildTermsInTermSetResponse
xmlns="http://schemas.microsoft.com/sharepoint/taxonomy/soap/"><GetChildTermsInTermSetResult>&lt;?xml version="1.0" encoding="utf-16"?&gt;&lt;TermStore&gt;&lt;T
a9="2d82f05a-1a82-4437-ad06-105650620d70" a21="false" a61="0"
a1000="10"&gt;&lt;LS&gt;&lt;TL a32="Взаиморасчеты" a31="true" /&gt;&lt;/LS&gt;&lt;DS
/&gt;&lt;TMS&gt;&lt;TM a24="212ebc29-c33e-407b-9c82-58b58a8b91fc" a12="Возможности"
a40="" a17="true" a67="" a45="2d82f05a-1a82-4437-ad06-105650620d70" a69="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/TermStore&gt;</GetChildTermsInTermSetResult></GetChild
TermsInTermSetResponse></soap:Body></soap:Envelope>

```

2. Пример обращение к Web-службе управляемых метаданных (TaxonomyClientService): метод GetChildTermsInTerm

```

// функция построения дерева возможностей 2-го уровня и ниже (GetChildTermsInTerm)
function SP_f2(dtnode,lang) {
    var terms = new Array();
    var terms_name = new Array();
    var terms_id = new Array();
    var tree_obj = new Array();
    var lcid;
    if (lang=='RUS') {lcid=1049} else {lcid=1033};
    var termId = dtnode.data.key;
    $().SPServices({
        operation:"GetChildTermsInTerm",
        sspId:"0f646e47-bf80-4cc3-a89a-9717f1bb5d57",
        lcid: lcid, //1033 - ENU, 1049 - RUS
        termId: termId,
        termSetId: "212ebc29-c33e-407b-9c82-58b58a8b91fc", //Возможности
        completefunc: function (xData, Status) {
            var result = xData.responseXML.text;
            if (window.DOMParser) {
                parser=new DOMParser();
                xmlDoc=parser.parseFromString(result,"text/xml");
            } else { // Internet Explorer
                xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
                xmlDoc.async="false";
                xmlDoc.loadXML(result);
            }
            terms = xmlDoc.getElementsByTagName("T");
            for(var j=0; j<terms.length; j++)
            {
                terms_id[j] =terms[j].attributes[0].value;
                terms_name[j] = terms[j].childNodes(0).childNodes(0).getAttribute("a32");
            }
            // обработка терминов
            if (terms.length>0)
            {
                for (var j=0; j<terms.length; j++)
                {

```



```

a61="0"&gt;&lt;/LS&gt;&lt;/TL a32="Отчеты" a31="true" /&gt;&lt;/LS&gt;&lt;/DS
/&gt;&lt;/TMS&gt;&lt;/TM a24="212ebc29-c33e-407b-9c82-58b58a8b91fc" a12="Возможности"
a25="2d82f05a-1a82-4437-ad06-105650620d70" a40="Взаиморасчеты" a17="true" a67=""
a45="2d82f05a-1a82-4437-ad06-105650620d70;8c95d2d0-bae4-4421-945e-cel25e32bb09"
a69="true" /&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/T a9="cb56fd78-3e42-4765-80b9-2efa7f280004"
a21="false" a61="0" a1000="7"&gt;&lt;/LS&gt;&lt;/TL a32="Сверка объемов и стоимости
переданного трафика" a31="true" /&gt;&lt;/LS&gt;&lt;/DS /&gt;&lt;/TMS&gt;&lt;/TM
a24="212ebc29-c33e-407b-9c82-58b58a8b91fc" a12="Возможности" a25="2d82f05a-1a82-4437-
ad06-105650620d70" a40="Взаиморасчеты" a17="true" a67="" a45="2d82f05a-1a82-4437-ad06-
105650620d70;cb56fd78-3e42-4765-80b9-2efa7f280004" a69="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/T a9="b3b7a5a7-f253-43dd-a51e-cb3d8c1f81f1" a21="false"
a61="0"&gt;&lt;/LS&gt;&lt;/TL a32="Формирование нормативно-справочной информации"
a31="true" /&gt;&lt;/LS&gt;&lt;/DS /&gt;&lt;/TMS&gt;&lt;/TM a24="212ebc29-c33e-407b-9c82-
58b58a8b91fc" a12="Возможности" a25="2d82f05a-1a82-4437-ad06-105650620d70"
a40="Взаиморасчеты" a17="true" a67="" a45="2d82f05a-1a82-4437-ad06-
105650620d70;b3b7a5a7-f253-43dd-a51e-cb3d8c1f81f1" a69="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/T a9="2bd8e8db-e404-4716-9ad7-3595c8f46732" a21="false"
a61="0" a1000="13"&gt;&lt;/LS&gt;&lt;/TL a32="Формирование счетов" a31="true"
/&gt;&lt;/LS&gt;&lt;/DS /&gt;&lt;/TMS&gt;&lt;/TM a24="212ebc29-c33e-407b-9c82-
58b58a8b91fc" a12="Возможности" a25="2d82f05a-1a82-4437-ad06-105650620d70"
a40="Взаиморасчеты" a17="true" a67="" a45="2d82f05a-1a82-4437-ad06-
105650620d70;2bd8e8db-e404-4716-9ad7-3595c8f46732" a69="true"
/&gt;&lt;/TMS&gt;&lt;/T&gt;&lt;/TermStore&gt;</GetChildTermsInTermResult></GetChildTer
msInTermResponse></soap:Body></soap:Envelope>

```

3. Пример обращения к Web-службе списков (Lists): метод GetListItems

```

// функция получения связанных МД для функциональной возможности
function SP_mod(prefix, lang) {
    var mod = new Array();
    var mod_min = new Array();
    var mod_con = new Array();
    var mod_t = new Array();
    var mod_shifr = new Array();
    var x="https://intra.billing.ru/";
    var val= '';
    //обращаемся к списку
    $(().SPServices({
        operation:"GetListItems",
        webURL: "https://intra.billing.ru/sites/docs/",
        listName:"Модули",
        CAMLQuery: "<Query><GroupBy><FieldRef
Name='ContentType' /></GroupBy><Where><And><Eq><FieldRef
Name='_x0424_x0443_x043d_x043a_x0446_x0438_x043e_x043d_x0430_x043b_x044c_x0
43d_x0430_x044f_x0020_x0432_x043e_x0437_x043c_x043e_x0436_x043d_x043e_x044
1_x0442_x044c_' /><Value Type='Text'>"+prefix+"</Value></Eq><Eq><FieldRef
Name='_x042f_x0437_x044b_x043a_x0020_x043b_x043e_x043a_x0430_x043b_x0438_x0
437_x0430_x0446_x0438_x0438_' /><Value
Type='Text'>"+lang+"</Value></Eq></And></Where></Query>",
        CAMLQueryOptions: "<QueryOptions><Folder></Folder></QueryOptions>", //чтобы папок
не было видно
        completefunc: function (xData, Status) {
            var i=0;
            $(xData.responseXML).find("z\\:row").each(function() {
                //отсекаем модуль - Возможность
                if ($(this).attr("ows_ContentType")!='Возможность')
                {
                    i=i+1;
                    val=$(this).attr("ows_FileRef");
                    var pos=val.lastIndexOf("#");
                    //получаем полный путь к очередному xml
                    mod[i]=x + val.substr(pos+1);
                    mod_min[i]=val.substr(pos+1);
                    mod_con[i]=$ (this).attr("ows_ContentType");
                    mod_shifr[i]=$ (this).attr("ows__x0428_x0438_x0444_x0440_");
                }
            });
            if (mod.length)

```

```

    {
        $("#divId2").append('<h3>Связанная информация:</h3>');
        var j = 0;
        var cont;
        var cont_t;
        for ( c= 1; c <mod.length; c++)
        {
            cont = mod_con[i];
            if(cont!=cont_t)
            {
                $("#divId2").append('<b>'+mod_con[c]+ '</b><br>');
            }
            $("#divId2").append('<UL>');
            if (mod_shifr[c]!=undefined)
            {
                $("#divId2").append('<LI><A
                HREF="https://intra.billing.ru/sites/docs/JS/html/view_mod.html?var1='+mod[c]+'
                TARGET="_blank">'+mod_shifr[c]+'</A></LI>');
            }
            else
            {
                val = mod_min[c];
                pos = val.lastIndexOf("/")+1;
                val = mod_min[c].substring(pos);
                pos = val.indexOf(".");
                val = val.substring(0,pos);
                $("#divId2").append('<LI><A
                HREF="https://intra.billing.ru/sites/docs/JS/html/view_mod.html?var1='+mod[c]+'
                TARGET="_blank">'+val+'</A><br></LI>');
            }
            $("#divId2").append('</UL>');
            cont_t = mod_con[i];
        }
    }
}
});
}

```

Пример SOAP-ответа Web-службы:

```

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetListItemResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/"><GetListItemResult><listitems
xmlns:s="uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882" xmlns:dt="uuid:C2F41010-65B3-11d1-
A29F-00AA00C14882" xmlns:rs="urn:schemas-microsoft-com:rowset"
xmlns:z="#RowsetSchema">
    <rs:data ItemCount="9">
        <z:row ows_DocIcon="xml"
ows_LinkFilename="Сверка_объемов_и_стоимости_переданного_трафика.FTR.RUS.xml"
ows_Modified="2011-08-11 16:02:31" ows_Editor="28;#Prokofieva, Olga"
ows_x0424_x0443_x043d_x043a_x0446_x0438_x043e_x043d_x0430_x043b_x044c_x043
d_x0430_x044f_x0020_x0432_x043e_x0437_x043c_x043e_x0436_x043d_x043e_x0441_
_x0442_x044c_"=7;#Сверка_объемов_и_стоимости_переданного_трафика"
ows_x042f_x0437_x044b_x043a_x0020_x043b_x043e_x043a_x0430_x043b_x0438_x043
7_x0430_x0446_x0438_x0438_"=RUS" ows_ContentType="Возможность"
ows_ModerationStatus="0" ows_Level="1" ows_UniqueId="24036;#{31B219F0-2097-44C2-
9516-87671DDD15EA}" ows_FSObjType="24036;#0" ows_Created_x0020_Date="24036;#2011-08-11
16:01:32" ows_ProgId="24036;#Word.Document"
ows_FileRef="24036;#sites/docs/DocLib/ITC/ITC_CMP_DATA_PROCESSING/Сверка_объемов_и_сто
имости_переданного_трафика.FTR.RUS.xml" ows_Last_x0020_Modified="24036;#2011-08-11
16:02:31" ows_ID="24036" ows_owshiddenversion="5"
ows_FileLeafRef="24036;#Сверка_объемов_и_стоимости_переданного_трафика.FTR.RUS.xml"
ows_PermMask="0x7fffffff" ows_HTML_x0020_File_x0020_Type="Word.Document" />
        <z:row ows_DocIcon="xml"
ows_LinkFilename="Создание_характеристики_финансовой_услуги.OPR.RUS.xml"
ows_Modified="2011-08-02 11:57:24" ows_Editor="12;#Rubinova, Olga"
ows_x041f_x043e_x0434_x0441_x0438_x0441_x0442_x0435_x043c_x0430_"=ITC_CMP_DA
TA_PROCESSING "

```

```

ows_x0424_x0443_x043d_x043a_x0446_x0438_x043e_x043d_x0430_x043b_x044c_x043
d_x0430_x044f_x0020_x0432_x043e_x0437_x043c_x043e_x0436_x043d_x043e_x0441_
_x0442_x044c_="7;#Сверка объемов и стоимости переданного трафика"
ows_x042f_x0437_x044b_x043a_x0020_x043b_x043e_x043a_x0430_x043b_x0438_x043
7_x0430_x0446_x0438_x0438_="RUS" ows_ContentType="Операция"
ows_x041f_x043e_x0434_x0441_x0438_x0441_x0442_x0435_x043c_x0430_x003a_x002
0_x041d_x0430_x0438_x043c_x0435_x043d_x043e_x0432_x0430_x043d_x0438_x0435_
="Обработка данных биллинга"
ows_x041f_x043e_x0434_x0441_x0438_x0441_x0442_x0435_x043c_x0430_x003a_x002
0_x041d_x0430_x0438_x043c_x0435_x043d_x043e_x0432_x0430_x043d_x0438_x0435_
_x0020_x0430_x043d_x0433_x043b_x0438_x0439_x0441_x043a_x043e_x0435_="Process
ing of billing results" ows_ModerationStatus="0" ows_Level="1"
ows_UniqueId="23801;#{B5BA85AA-C0A4-4B42-B36B-CB6A67657B9C}" ows_FSObjType="23801;#0"
ows_Created_x0020_Date="23801;#2011-07-28 11:06:31" ows_ProgId="23801;#Word.Document"
ows_FileRef="23801;#sites/docs/DocLib/ITC/ITC_CMP_DATA_PROCESSING/Создание_характерист
ики_финансовой_услуги.OPR.RUS.xml" ows_Last_x0020_Modified="23801;#2011-08-02
11:57:24" ows_ID="23801" ows_owshiddenversion="11"
ows_FileLeafRef="23801;#Создание_характеристики_финансовой_услуги.OPR.RUS.xml"
ows_PermMask="0x7fffffff" ows_HTML_x0020_File_x0020_Type="Word.Document"/>
</rs:data>
</listitems></GetListItemsResult></GetListItemsResponse></soap:Body></soap:Envelope>

```

4. Пример обращения к Web-службе поиска (Search): метод Query

```

//параметры поиска
var quickSearchConfig = {
    minCharacters: 2,           // минимальное количество символов для поиска
    scope: "All Sites",       // scope поиска
    numberOfResults: 1000     // количество показываемых результатов (по умолчанию -
    первые 10)
};

//функция поиска
function Query_S(query, lang) {

    var query = query; // запрос поиска
    var mod = new Array();
    var queryText = "<QueryPacket xmlns='urn:Microsoft.Search.Query' Revision='1000'>"
    queryText +=
    "<SupportedFormats><Format>urn:Microsoft.Search.Response.Document.Document</Format></S
upportedFormats>"
    queryText += "<Query>"
    queryText += "<Context>"
    queryText += "<QueryText language='en-US' type='STRING' >SCOPE:\" +
    quickSearchConfig.scope + \"\" + query + "
    SITE:\"https://intra.billing.ru/sites/docs/DocLib\""+</QueryText>"
    queryText += "</Context>"
    queryText += "<Range><StartAt>1</StartAt><Count>1000</Count></Range>"
    queryText += "<TrimDuplicates>>false</TrimDuplicates>"
    queryText += "</Query>"
    queryText += "</QueryPacket>";

    $.SPServices({
        operation: "Query",
        queryXml: queryText,
        completefunc: function(xData, Status)
        {
            $(xData.responseXML).find("QueryResult").each(function() {
                var result = $(this).text();
                if (window.DOMParser) {
                    parser=new DOMParser();
                    xmlDoc=parser.parseFromString(result,"text/xml");
                } else { // Internet Explorer
                    xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
                    xmlDoc.async="false";
                    xmlDoc.loadXML(result);
                }
            });
        }
    });
}

```

```

    }
    var LinkUrl = xmlDoc.getElementsByTagName("LinkUrl");
    var i = 0;
    for( var loopcount = 0; loopcount < LinkUrl.length; loopcount++)
    {
        var fileExt = LinkUrl[loopcount].getAttribute("fileExt");
        if (fileExt!=null) fileExt = fileExt.toUpperCase();
        if (fileExt=='XML')
        {
            mod[i] = LinkUrl[loopcount].text;
            i=i+1;
        }
    }
    for( var loopcount = 0; loopcount < mod.length; loopcount++)
    {
        SP_mod(mod[loopcount],lang);
    }
    });
}
});
$("#divImage").html("").append(' ');
};

```

Пример SOAP-ответа Web-службы:

```

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><QueryResponse
xmlns="urn:Microsoft.Search"><QueryResult>&lt;ResponsePacket
xmlns="urn:Microsoft.Search.Response"&gt;&lt;Response&gt;&lt;Range&gt;&lt;StartAt&gt;1
&lt;/StartAt&gt;&lt;Count&gt;4&lt;/Count&gt;&lt;TotalAvailable&gt;4&lt;/TotalAvailable
&gt;&lt;Results&gt;&lt;Document relevance="61807247"
xmlns="urn:Microsoft.Search.Response.Document"&gt;&lt;Title&gt;TAB.STP.1507.RUS&lt;/Ti
tle&gt;&lt;Action&gt;&lt;LinkUrl size="149858"
fileExt="XML"&gt;https://intra.billing.ru/sites/docs/DocLib/BIS/TAB/TAB.STP.1507.RUS.X
ML&lt;/LinkUrl&gt;&lt;Action&gt;&lt;Description /&gt;&lt;Date&gt;2009-11-
18T17:40:19&lt;/Date&gt;&lt;Document&gt;&lt;Document relevance="61804879"
xmlns="urn:Microsoft.Search.Response.Document"&gt;&lt;Title&gt;TAB.STP.1507.ENU&lt;/Ti
tle&gt;&lt;Action&gt;&lt;LinkUrl size="154961"
fileExt="XML"&gt;https://intra.billing.ru/sites/docs/DocLib/BIS/TAB/TAB.STP.1507.ENU.X
ML&lt;/LinkUrl&gt;&lt;Action&gt;&lt;Description /&gt;&lt;Date&gt;2009-11-
18T17:40:18&lt;/Date&gt;&lt;Document&gt;&lt;Document relevance="61486718"
xmlns="urn:Microsoft.Search.Response.Document"&gt;&lt;Title&gt;PSTN_TAB.STP.1507.RUS&lt
;/Title&gt;&lt;Action&gt;&lt;LinkUrl size="207228"
fileExt="xml"&gt;https://intra.billing.ru/sites/docs/DocLib/BFN/PSTN_TAB/PSTN_TAB.STP.
1507.RUS.xml&lt;/LinkUrl&gt;&lt;Action&gt;&lt;Description /&gt;&lt;Date&gt;2011-05-
04T11:30:16&lt;/Date&gt;&lt;Document&gt;&lt;Document relevance="55348591"
xmlns="urn:Microsoft.Search.Response.Document"&gt;&lt;Title&gt;PSTN_TAB&lt;/Title&gt;&
lt;Action&gt;&lt;LinkUrl
size="0"&gt;https://intra.billing.ru/sites/docs/DocLib/BFN/PSTN_TAB&lt;/LinkUrl&gt;&lt
;/Action&gt;&lt;Description /&gt;&lt;Date&gt;2011-05-
04T10:52:17&lt;/Date&gt;&lt;Document&gt;&lt;/Results&gt;&lt;/Range&gt;&lt;Status&gt;S
UCCESS&lt;/Status&gt;&lt;/Response&gt;&lt;/ResponsePacket&gt;</QueryResult></QueryResp
onse></soap:Body></soap:Envelope>

```

5. Пример обращения к Web-службе работы с пользователями и группами (Users and Groups): метод GetGroupCollectionFromUser

```

// получение пользователя
var User = $().SPServices.SPGetCurrentUser();
var role;

// получение группы пользователя
$(().SPServices({
    operation: "GetGroupCollectionFromUser",
    userLoginName: User,
    async: false,
    completefunc: function(xData, Status) {

```

```

$(xData.responseXML).find("[nodeName='Group']").each(function() {
    $("#divId2").append($(this).attr("Name"));
    role = $(this).attr("Name");
});
}
});

```

Пример SOAP-ответа Web-службы:

```

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetGroupCollectionFromUserResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/directory/"><GetGroupCollectionFromUserResult><GetGroupCollectionFromUser><Groups><Group ID="3" Name="Разработка документации Owners" Description="Use this group to give people full control permissions to the SharePoint site: Разработка документации" OwnerID="3" OwnerIsUser="False"/></Groups></GetGroupCollectionFromUser></GetGroupCollectionFromUserResult></GetGroupCollectionFromUserResponse></soap:Body></soap:Envelope>

```

Приложение Е – Таблица XSLT-стилей для преобразования XML-документа модулей документации

```

<?xml version="1.0" encoding="windows-1251"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"
xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint"
xmlns:wsp="http://schemas.microsoft.com/office/word/2003/wordml/sp2"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:aml="http://schemas.microsoft.com/aml/2001/core"
xmlns:v="urn:schemas-microsoft-com:vml">
<xsl:param name="selfDoc"/>
<xsl:output method="html" encoding="windows-1251" doctype-public="-//W3C//DTD XHTML
1.0 Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"/>
<xsl:template match="/">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251"/>
<title>
<xsl:apply-templates select="w:wordDocument/o:DocumentProperties"/>
</title>
</head>
<body>
<xsl:apply-templates select="w:wordDocument/w:body"/>
</body>
</html>
</xsl:template>

<xsl:template match="w:wordDocument/o:DocumentProperties">
<xsl:value-of select="o:Title"/>
</xsl:template>

<xsl:template match="w:wordDocument/w:body">
<xsl:apply-templates />
</xsl:template>

<!-- DEF Style -->
<xsl:template match="w:p">
<xsl:choose>

<!-- Format Contents -->
<xsl:when test="w:r/w:instrText">
<xsl:variable name="str" select="w:r/w:instrText" />
<xsl:variable name="hrname" select="substring-before(substring-after(w:r/w:instrText,
'PAGEREF '), '\h ')" />
<xsl:if test="contains(w:r/w:instrText, 'TOC \t')">
<div style="padding-left: 10px;"><a href="#"{$hrname}><xsl:apply-templates
select="w:r" /></a ></div>
</xsl:if>
<xsl:if test="contains(w:r/w:instrText, 'PAGEREF')">
<div style="padding-left: 10px;"><a href="#"{$hrname}><xsl:apply-templates
select="w:r" /></a ></div>
</xsl:if>
</xsl:when>

<!-- Sect11 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Sect11'">
<xsl:variable name="alink" select="aml:annotation/@w:name" />
<a name="{ $alink }">
<div style="margin: 2px; padding-bottom: 20px; padding-top: 15px;"> <font style="font-
family:Arial; font-size:24pt; font-weight: bold; text-transform: uppercase;">
<xsl:apply-templates select="w:r" /></font></div></a>
</xsl:when>

```

```

<!-- Sect22 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Sect22'">
<xsl:variable name="alink" select="aml:annotation/@w:name" />
<a name="{ $alink }">
<div style="margin: 2px; padding-bottom: 10px;padding-top: 25px;"> <font style="font-
family:Arial; font-size:14pt; font-weight: bold;">
<xsl:apply-templates select="w:r" /></font></div></a>
</xsl:when>

<!-- Sect33 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Sect33'">
<xsl:variable name="alink" select="aml:annotation/@w:name" />
<a name="{ $alink }">
<div style="margin: 2px; padding-bottom: 10px;padding-top: 25px;"> <font style="font-
family:Arial; font-size:14pt; font-weight: bold;">
<xsl:apply-templates select="w:r" /></font></div></a>
</xsl:when>

<!-- Sect44 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Sect44'">
<br/><div style="margin: 2px; padding-left: 10px; padding-bottom: 5px;"><i><font
style="font-family:Arial; font-size:12pt; font-weight: bold;">
<xsl:apply-templates select="w:r" /></font></i></div>
</xsl:when>

<!-- ParaPlate -->
<xsl:when test="w:pPr/w:pStyle/@w:val='ParaPlate'">
<div style="padding-bottom: 10px; padding-top: 15px;"><div style="font-family:Courier
New; font-size:12pt; background: E6E6E6; color: 000000; font-weight: bold;" >
<xsl:apply-templates select="w:r" />
</div></div>
</xsl:when>

<!-- ErrorMess -->
<xsl:when test="w:pPr/w:pStyle/@w:val='ErrorMess'">
<div style="padding-bottom: 10px; padding-top: 15px;"><div style=" font-family:Arial;
font-size:11pt; background: E6E6E6; color: 000000; " >
<xsl:apply-templates select="w:r" />
</div></div>
</xsl:when>

<!-- DBTabHeader -->
<xsl:when test="w:pPr/w:pStyle/@w:val='DBTabHeader'">
<div style="padding-bottom: 7px; padding-top: 12px; padding-left: 10px;"><div style="
font-family:Arial; font-size:11pt; background: E6E6E6; color: 000000; font-weight:
bold;" >
<xsl:apply-templates select="w:r" />
</div></div>
</xsl:when>

<!-- Line -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Line'">
<div style="padding-bottom: 5px; padding-top: 95px;"><div style="font-family:Arial;
font-size:12pt; background: 000000; color: FFFFFFFF; font-weight: bold; text-transform:
uppercase;" >
<xsl:apply-templates select="w:r" />
</div></div>
</xsl:when>

<!-- ListMarker11 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='ListMarker11'">
<div style="margin: 2px; padding-left: 10px;">
<li><font style="font-family:Arial; font-size:11pt">
<xsl:apply-templates select="w:r | w:hlink" />
</font></li></div>
</xsl:when>

<!-- ListMarker22 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='ListMarker22'">
<div style="margin: 2px; padding-left: 25px;">

```

```

<li><font style="font-family:Arial; font-size:11pt">
<xsl:apply-templates select="w:r | w:hlink" /></font></li></div>
</xsl:when>

<!-- ListMarker33 -->
<xsl:when test="w:pPr/w:pStyle/@w:val='ListMarker33'">
<div style="margin: 2px; padding-left: 40px;">
<li><font style="font-family:Arial; font-size:11pt">
<xsl:apply-templates select="w:r | w:hlink" /></font></li></div>
</xsl:when>

<!-- NotList -->
<xsl:when test="w:pPr/w:pStyle/@w:val='NotList'">
<div style="margin: 2px; padding-left: 25px;"> <font style="font-family:Arial; font-
size:11pt">
<xsl:apply-templates select="w:r | w:hlink" />
</font></div>
</xsl:when>

<!-- CodeNotList -->
<xsl:when test="w:pPr/w:pStyle/@w:val='CodeNotList'">
<div style="margin: 2px; padding-left: 25px;"> <i><font style="font-family:Courier
New; font-size:10pt">
<xsl:apply-templates select="w:r | w:hlink" />
</font></i></div>
</xsl:when>

<!-- AnnotateNotList -->
<xsl:when test="w:pPr/w:pStyle/@w:val='AnnotateNotList'">
<div style="margin: 2px; padding-left: 25px;"> <i><font style="font-family:Arial;
font-size:10pt; color: 444444;">
<xsl:apply-templates select="w:r | w:hlink" />
</font></i></div>
</xsl:when>

<!-- ListNumber11 poka zatyk-->
<xsl:when test="w:pPr/w:pStyle/@w:val='ListNumber11'">
<div style="margin: 2px; padding-left: 10px;">
<li><font style="font-family:Arial; font-size:11pt">
<xsl:apply-templates select="w:r | w:hlink" />
</font></li></div>
</xsl:when>

<!-- Annotate -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Annotate'">
<div style="margin: 2px; padding-left: 10px;"> <i><font style="font-family:Arial;
font-size:10pt; color: 444444;">
<xsl:apply-templates select="w:r | w:hlink" />
</font></i></div>
</xsl:when>

<!-- Code -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Code'">
<div style="margin: 2px; padding-left: 10px;"> <i><font style="font-family:Courier
New; font-size:10pt">
<xsl:apply-templates select="w:r | w:hlink" />
</font></i></div>
</xsl:when>

<!-- Picture -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Picture'">
<div style="margin: 20px; text-align: center; ">
<xsl:apply-templates select="w:r" /></div>
</xsl:when>

<!-- Outline -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Outline'">
<div style="margin: 10px; text-align: center; ">
<xsl:apply-templates select="w:r | w:fldSimple/w:r | w:hlink" /></div>
</xsl:when>

```

```

<!-- Book -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Book'">
<div style="margin: 150px; text-align: center; "> <font style="font-family:Arial;
font-size:30pt; text-transform: uppercase; font-weight: bold; ">
<xsl:apply-templates select="w:fldSimple/w:r" />
</font></div>
</xsl:when>

<!-- Contents -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Contents'">
<div style="margin: 2px; padding-bottom: 10px;padding-top: 10px;"> <font style="font-
family:Arial; font-size:14pt; font-weight: bold;">
<xsl:apply-templates select="w:r" /></font></div>
</xsl:when>

<!-- Para -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Para'">
<div style="margin: 2px; padding-left: 10px;"> <font style="font-family:Arial; font-
size:11pt">
<xsl:apply-templates select="w:r | w:fldSimple/w:r | w:hlink" />
</font></div>
</xsl:when>

<!-- Delit -Introduction -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Introduction'">
<xsl:apply-templates select="w:p " />
</xsl:when>

<!-- Delit -Copyright -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Copyright'">
<xsl:apply-templates select="w:p | w:hlink" />
</xsl:when>

<!-- Delit -History -->
<xsl:when test="w:pPr/w:pStyle/@w:val='History'">
<xsl:apply-templates select="w:p | w:hlink" />
</xsl:when>

<!-- Delit -HistoryText -->
<xsl:when test="w:pPr/w:pStyle/@w:val='HistoryText'">
<xsl:apply-templates select="w:p | w:hlink" />
</xsl:when>

<!-- Delit -Date -->
<xsl:when test="w:pPr/w:pStyle/@w:val='Date'">
<xsl:apply-templates select="w:p | w:hlink" />
</xsl:when>

<!-- otherwise - Para -->
<xsl:otherwise>
<div style="font-family:Arial; font-size:11pt; padding-left: 11px;">
<xsl:apply-templates select="w:r" />
</div>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- Del -->
<xsl:template match="w:sectPr">
<xsl:choose>
<xsl:when test="w:ftr/w:type='even'">
<xsl:apply-templates select="w:p" />
</xsl:when>
</xsl:choose>
</xsl:template>

<!-- Formatting (Path, Code, Field, Menu, Button), images -->
<xsl:template match="w:r">
<xsl:choose>

```

```

    <xsl:when test="w:rPr/w:rStyle/@w:val='Path'"><font style="font-family:Courier New;
font-size:12pt">
<xsl:value-of select="w:t" /></font></xsl:when>

    <xsl:when test="w:rPr/w:rStyle/@w:val='Code'"><i><font style="font-family:Courier
New; font-size:10pt">
<xsl:value-of select="w:t" /></font></i></xsl:when>

    <xsl:when test="w:rPr/w:rStyle/@w:val='Field'"><font style="font-family:Arial;
font-size:11pt; font-weight: bold">
<xsl:value-of select="w:t" /></font></xsl:when>

    <xsl:when test="w:rPr/w:rStyle/@w:val='Menu'"><u><font style="font-family:Arial;
font-size:11pt; font-weight: bold">
<xsl:value-of select="w:t" /></font></u></xsl:when>

    <xsl:when test="w:rPr/w:rStyle/@w:val='Button'"><i><u><font style="font-
family:Arial; font-size:11pt; font-weight: bold">
<xsl:value-of select="w:t" /></font></u></i></xsl:when>

<xsl:when test="w:pict">
    <xsl:variable name="imgSrc" select="w:pict/v:shape/v:imagedata/@src" />
    <xsl:variable name="img" select="//w:binData[@w:name=$imgSrc]"/>
    
</xsl:when>

    <xsl:otherwise><xsl:value-of select="w:t" /></xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- Parse hyperlinks -->
<xsl:template match="w:hlink">
    <xsl:variable
name="PathToDocs">https://intra.billing.ru/sites/docs/DocLib/</xsl:variable>
    <xsl:variable name="LinkToDM" select="@w:dest" />
    <xsl:choose>
        <xsl:when test="contains($LinkToDM,$PathToDocs)">
            <a
href="https://intra.billing.ru/sites/docs/JS/html/view_mod.html?var1={@w:dest}">
                <font style="font-family:Courier New; font-size:12pt">
                    <xsl:value-of select="." />
                </font>
            </a>
        </xsl:when>
        <xsl:otherwise>
            <a href="{@w:dest}">
                <font style="font-family:Arial; font-size:11pt">
                    <xsl:value-of select="." />
                </font>
            </a>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

</xsl:stylesheet>

```

Приложение Ж – Шаблон для формирования руководства системного программиста в HTML-формате

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <title>Please wait...</title>
  <style media="print"> /* стили для печати */
  .noprint {display: none;} /* элементы этого класса не будут напечатаны */
  </style>
<script type="text/javascript" src="../jquery-1.6.1.min.js"></script>
<script type="text/javascript" src="../jquery.SPServices-0.6.1.min.js"></script>
<script type="text/javascript">
//-----
// Шаблон Руководства системного программиста на подсистему
//
// Пример запроса:
// https://intra.billing.ru/sites/docs/JS/html/ADMIN.html?var1=MTH;var2=ENU
// https://intra.billing.ru/sites/docs/JS/html/ADMIN.html?var1=BIL_PL;var2=RUS
//
// var1 - префикс подсистемы
// var2 - язык (RUS, ENU). По умолчанию - RUS
//-----
//-----

// инициализирующая функция
function start(){
  // первый параметр - префикс подсистемы
  var prefix = get_java("var1");
  // второй параметр - язык (RUS, ENU и т.д.)
  var lang = get_java("var2");
  // по умолчанию RUS
  if (lang == '')
  {
    lang = 'RUS';
  }
  SP_mod(prefix, lang);
};

// трансформация XML-модулей
function trans(mod, moj)
{
  var vers; //версия ворда
  //загружаем XML
  xml=new XMLHttpRequest("Microsoft.XMLDOM");
  xml.async=false;
  xml.load(mod);
  //определяем вариант стиля
  style = "https://intra.billing.ru/sites/docs/XSL/DSC.xsl";

  //загружаем выбранный XSL
  xsl = new XMLHttpRequest("Microsoft.XMLDOM");
  xsl.async = false;
  xsl.load(style);

  $("#divId").append('<a name="'+ moj +'">'+</a><br>'+ xml.transformNode(xsl) +
  '<br/>');
}

// применение стилей
function St(st,text, div)
{
  $(div).append('<a name="'+ text +'"></a><br>');
  if (st == 'Book')
  {

```

```

    $(div).append('<div style="margin: 10px; text-align: center; "> <font style="font-
family:Arial; font-size:30pt; text-transform: uppercase; font-weight: bold;
">'+text+'</font>');
    }
    if (st == 'NumberType')
    {
        $(div).append('<div style="margin: 2px; text-align: center; "> <font style="font-
family:Arial; font-size:14pt; font-weight: bold; text-transform: uppercase;
">'+text+'</font>');
    }

    if (st == 'Sect1')
    {
        $(div).append('<div style="margin: 2px; padding-bottom: 20px; padding-top:
15px;"><font style="font-family:Arial; font-size:24pt; font-weight: bold; text-
transform: uppercase;">'+text+'</font>');
    }
    if (st == 'Sect2')
    {
        $(div).append('<div style="margin: 2px; padding-bottom: 10px;padding-top:
25px;"><font style="font-family:Arial; font-size:14pt; font-weight:
bold;">'+text+'</font>');
    }
    if (st == 'Line')
    {
        $(div).append('<div style="padding-bottom: 5px; padding-top: 95px;"><div
style="font-family:Arial; font-size:12pt; background: 000000; color: FFFFFFFF; font-
weight: bold; text-transform: uppercase;" >'+text+'</div></div>');
    }
    if (st == 'Contents')
    {
        $(div).append('<div style="margin: 2px; padding-bottom: 10px;padding-top: 10px;">
<font style="font-family:Arial; font-size:14pt; font-weight: bold;">'+text+'</font>');
    }
}

// преобразовываем путь к файлу
function freplace(x) {
    var s = String(x);
    return s;
}

// реализация метода GET
function get_java(n)
{
    a=window.location.toString();
    a=" "+a.substring(a.indexOf("?")+1);
    p=a.indexOf(";"+n+"=");
    a=p==-1?'':a.substring(p+n.length+2);
    return a.indexOf(";")>=0?a.substring(0,a.indexOf(";")):a;
}

// получение наименования подсистемы
function SP_subsystem(prefix, lang) {
    var arr_name_ps = new Array();
    var kod = '';
    var kod_doc; ///643.11150642.XXXXX-05 32 01-M
    var subsystem;
    var subsystem_en;
    $("#divSubsystem").SPServices({
        operation:"GetListItems",
        webURL: "https://intra.billing.ru/dw/MasterData",
        listName:"Подсистема",
        CAMLQuery: "<Query><Where><Eq><FieldRef Name='PSCodeField' /><Value
Type='Text'>"+prefix+"</Value></Eq></Where></Query>",
        //CAMLRowLimit: "<RowLimit>10</RowLimit>",
        CAMLQueryOptions: "<QueryOptions><Folder></Folder></QueryOptions>",
        //чтобы папок не было видно
        completefunc: function (xData, Status)

```

```

    {
        $(xData.responseXML).find("z\\:row").each(function()
        {
            subsystem = $(this).attr("ows_Title");
            subsystem_en = $(this).attr("ows_PSTitleENField");
            kod = $(this).attr("ows_PSRegNumField");
        });
        if (lang == 'RUS')
        {
            St('Book', 'Подсистема «'+subsystem+'»', "#divSubsystem");
            St('NumberType', 'Руководство системного программиста',
"#divSubsystem");
        }
        // Заголовок окна
        document.title = 'Руководство системного программиста на
подсистему«'+subsystem+'» ('+prefix+')';
    }
    else
    {
        St('Book', subsystem_en, "#divSubsystem");
        St('NumberType', 'ADMINISTRATOR GUIDE', "#divSubsystem");
        document.title = 'Administrator Guide for the
application«'+subsystem_en+'» ('+prefix+')';
    }
    //формируем номер документа
    if (kod.length == 1)
    {
        kod_doc = '643.11150642.0000'+kod+'-05 32 01-M';
    }
    else if (kod.length == 2)
    {
        kod_doc = '643.11150642.000'+kod+'-05 32 01-M';
    }
    else if (kod.length == 3)
    {
        kod_doc = '643.11150642.00'+kod+'-05 32 01-M';
    }
    else if (kod.length == 4)
    {
        kod_doc = '643.11150642.0'+kod+'-05 32 01-M';
    }
    else if (kod.length == 5)
    {
        kod_doc = '643.11150642.'+kod+'-05 32 01-M';
    }
    St('NumberType', kod_doc, "#divSubsystem");
}
});
} //конец SP_subsystem

// выборка данных из SharePoint и вывод данных
function SP_mod(prefix, lang) {

    var mod = new Array();
    var mod_j = new Array();
    var mod_cont;
    var x="https://intra.billing.ru/";
    var style = '';
    var val= '';
    var i=0;
    var j=0;
    var prefix;
    var xml;
    var xsl;
    var raz;
    var mod_DSC = new Array();
    var moj_DSC = new Array();
    var DSC = 0;
    var mod_STP = new Array();
    var moj_STP = new Array();
    var STP = 0;

```

```

var mod_API = new Array();
var moj_API = new Array()
var API = 0;
var mod_ERR = new Array();
var moj_ERR = new Array();
var ERR = 0;
var shifr = 0;
if (prefix!='')
{
    waitMessage = "<table width='100%' align='center'><tr><td align='center'><img
src='/_layouts/images/GEARS_AN.GIF' /></tr><tr><td align='center'>Please
wait...</td></tr></table>";
    //обращаемся к списку
    $("#divImage").html(waitMessage).SPServices({
        operation:"GetListItems",
        webURL: "https://intra.billing.ru/sites/docs/",
        listName:"Модули",
        CAMLQuery: "<Query><OrderBy><FieldRef Name='ContentType'
/></OrderBy><Where><And><And><Or><Or><Or><Or><Eq><FieldRef Name='ContentType' /><Value
Type='Text'>Назначение подсистемы</Value></Eq><Eq><FieldRef Name='ContentType' /><"+
        "<Value Type='Text'>Настроечный
параметр</Value></Eq></Or><Eq><FieldRef Name='ContentType' /><Value
Type='Text'>Процедура</Value></Eq></Or><Eq><FieldRef Name='ContentType' /><Value
Type='Text'>Функция</Value></Eq></Or><Eq><FieldRef Name='ContentType' /><Value
Type='Text'>Сообщение об ошибке</Value></Eq></Or><Eq><FieldRef
Name='_x041f_x043e_x0434_x0441_x0438_x0441_x0442_x0435_x043c_x0430_' /><Value
Type='Text'>"+prefix+"</Value></Eq></And><Eq><FieldRef
Name='_x042f_x0437_x044b_x043a_x0020_x043b_x043e_x043a_x0430_x043b_x0438_x0
437_x0430_x0446_x0438_x0438_' /><Value
Type='Text'>"+lang+"</Value></Eq></And></Where></Query>",
        //CAMLRowLimit: "<RowLimit>5</RowLimit>",
        CAMLQueryOptions: "<QueryOptions><Folder></Folder></QueryOptions>",
//чтобы папок не было видно
        completefunc: function (xData, Status) {
            $(xData.responseXML).find("z\\:row").each(function() {
                i=i+1;
                //для каждого очередного элемента получаем путь
                val=$(this).attr("ows_FileRef");
                var pos=val.lastIndexOf("#");
                //получаем полный путь к очередному xml
                mod[i]=x + val.substr(pos+1);
                mod_j[i] = $(this).attr("ows_LinkFilename");
                mod_cont=$(this).attr("ows_ContentType");
                shifr = $(this).attr("ows_x0428_x0438_x0444_x0440");
                if (mod_cont == 'Назначение подсистемы')
                {
                    DSC = DSC + 1;
                    mod_DSC[DSC]= mod[i];
                    moj_DSC[DSC] = freplace(mod_j[i]);
                }
                if (mod_cont == 'Настроечный параметр')
                {
                    STP = STP + 1;
                    mod_STP[STP]= mod[i];
                    moj_STP[STP] = freplace(mod_j[i]);
                }
            }
            if ((mod_cont == 'Процедура') || (mod_cont == 'Функция')) //или
            {
                API = API + 1;
                mod_API[API]= mod[i];
                moj_API[API] = freplace(mod_j[i]);
            }
            if (mod_cont == 'Сообщение об ошибке')
            {
                ERR = ERR + 1;
                mod_ERR[ERR]= mod[i];
                moj_ERR[ERR] = freplace(mod_j[i]);
            }
        }
    });

```

```

if (i!=0)
{
$("#divMain").append('<form>');
if (lang == 'RUS')
{
$("#divMain").append('<button class=\\"noprnt\\"
onClick=\\"window.print()\\">Печать</button>');
}
else
{
$("#divMain").append('<button class=\\"noprnt\\"
onClick=\\"window.print()\\">Print</button>');
}
$("#divMain").append('</form>');

// ВЫВОДИМ НАИМЕНОВАНИЕ ПОДСИСТЕМЫ
SP_subsystem(prefix,lang);

// Номер главы
var NUM = 1;

//формируем оглавление на основе массива целей
if (lang == 'RUS')
{
St('Contents','Содержание', "#divId");
}
else
{
St('Contents','CONTENTS', "#divId");
}

if (DSC!=0)
{
if (lang == 'RUS')
{
raz = NUM + ' Общие сведения';
}
else
{
raz = NUM + ' GENERAL INFORMATION';
}
$("#divId").append('<a href="#" + raz.toUpperCase() + "' >' + raz.toUpperCase() +
'</a><br>');
NUM = NUM + 1;
}
if (STP!=0)
{
if (lang == 'RUS')
{
raz = NUM + ' Установка и настройка подсистемы';
}
else
{
raz = NUM + ' INSTALLING AND CONFIGURING THE APPLICATION';
}
$("#divId").append('<a href="#" + raz.toUpperCase() + "'>' + raz.toUpperCase() +
'</a><br>');
if (lang == 'RUS')
{
raz = 'Настроечные параметры';
}
else
{
raz = 'SETUP PARAMETERS';
}
$("#divId").append('<a href="#" + raz + "'>' + raz + ' </a><br>');
for (var c= 1; c <=STP; c++)
{

```

```

        $("#divId").append('<a href="#" + moj_STP[c] +'>' + moj_STP[c].toUpperCase() +
'</a><br>');
    }
    NUM = NUM + 1;
}
if (API!=0)
{
    if (lang == 'RUS')
    {
        raz = NUM + ' Работа с подсистемой';
    }
    else
    {
        raz = NUM + ' USING THE APPLICATION';
    }
    $("#divId").append('<a href="#" + raz.toUpperCase() +'>' + raz.toUpperCase() +
'</a><br>');
    for (var c = 1; c <=API; c++)
    {
        $("#divId").append('<a href="#" + moj_API[c] +'>' + moj_API[c].toUpperCase() +
'</a><br>');
    }
    NUM = NUM + 1;
}
if (ERR!=0)
{
    if (lang == 'RUS')
    {
        raz = NUM + ' Сообщения об ошибках';
    }
    else
    {
        raz = NUM + ' ERROR MESSAGES';
    }
    $("#divId").append('<a href="#" + raz.toUpperCase() +'>' + raz.toUpperCase() +
'</a><br>');
    for (var c = 1; c <=ERR; c++)
    {
        $("#divId").append('<a href="#" + moj_ERR[c] +'>' + moj_ERR[c].toUpperCase() +
'</a><br>');
    }
    NUM = NUM + 1;
}

NUM = 1;
//
// Основная часть документа
if (DSC!=0)
{
    if (lang == 'RUS')
    {
        St('Line','ГЛАВА', "#divId");
        raz = NUM + ' Общие сведения';
        St('Sect1', raz.toUpperCase(),"#divId");
    }
    else
    {
        St('Line','CHAPTER', "#divId");
        raz = NUM + ' GENERAL INFORMATION';
        St('Sect1', raz.toUpperCase(),"#divId");
    }
    for (var c = 1; c <=DSC; c++)
    {
        trans(mod_DSC[c], moj_DSC[c]);
        if (lang == 'RUS')
        {
            $("#divId").append('<a href="#">К содержанию</a><br>');
        }
        else
        {

```

```

        $("#divId").append('<a href="#CONTENTS">To contents</a><br>');
    }
}
NUM = NUM + 1;
}
if (STP!=0)
{
    if (lang == 'RUS')
    {
        St('Line','ГЛАВА', "#divId");
        St('Sect1', NUM + ' Установка и настройка подсистемы', "#divId");
        St('Sect2','Настроечные параметры', "#divId");
    }
    else
    {
        St('Line','CHAPTER', "#divId");
        St('Sect1', NUM + ' INSTALLING AND CONFIGURING THE APPLICATION', "#divId");
        St('Sect2','SETUP PARAMETERS', "#divId");
    }
    for (var c= 1; c <=STP; c++)
    {
        trans(mod_STP[c], moj_STP[c]);
        if (lang == 'RUS')
        {
            $("#divId").append('<a href="#Содержание">К содержанию</a><br>');
        }
        else
        {
            $("#divId").append('<a href="#CONTENTS">To contents</a><br>');
        }
    }
    NUM = NUM + 1;
}
if (API!=0)
{
    if (lang == 'RUS')
    {
        St('Line','ГЛАВА', "#divId");
        St('Sect1', NUM + ' Работа с подсистемой', "#divId");
    }
    else
    {
        St('Line','CHAPTER', "#divId");
        St('Sect1', NUM + ' USING THE APPLICATION', "#divId");
    }
    for (var c= 1; c <=API; c++)
    {
        trans(mod_API[c], moj_API[c]);
        if (lang == 'RUS')
        {
            $("#divId").append('<a href="#Содержание">К содержанию</a><br>');
        }
        else
        {
            $("#divId").append('<a href="#CONTENTS">To contents</a><br>');
        }
    }
    NUM = NUM + 1;
}
if (ERR!=0)
{
    if (lang == 'RUS')
    {
        St('Line','ГЛАВА', "#divId");
        St('Sect1', NUM + ' Сообщения об ошибках', "#divId");
    }
    else
    {
        St('Line','CHAPTER', "#divId");
        St('Sect1', NUM + ' ERROR MESSAGES', "#divId");
    }
}

```

```

    }
    for (var c= 1; c <=ERR; c++)
    {
        trans(mod_ERR[c], moj_ERR[c]);
        if (lang == 'RUS')
        {
            $("#divId").append('<a href="#Содержание">К содержанию</a><br>');
        }
        else
        {
            $("#divId").append('<a href="#CONTENTS">To contents</a><br>');
        }
    }
    NUM = NUM + 1;
}
// */
$("#divImage").html("").append('<br/>');
} //if (i!=0)
else
{
    St('Sect2', 'Для подсистемы "' +prefix+'" не найдены данные!', "#divId");
    $("#divImage").html("").append('<br/>');
}
} //конец гл. проц.
});
} // конец if (prefix!='')
else
{
    St('Sect2', 'Не задан префикс подсистемы!', "#divId");
}
} // конец SP_mod

</script>

</head>
<body bgcolor="#F2FFF3"; onload="start()">

<div id="divMain"></DIV>
<div id="divImage"></DIV>
<div id="divSubsystem"></DIV>
<div id="divId"></DIV>

</body>
</html>

```

Приложение 3 – Структура списков подсистемы разграничения доступа

1. Список «Заказчики»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Заказчик	Наименование заказчика	Строка (Single line of text)	-
2.	Роль	Наименование роли	Пользователи и группы (Person or Group)	Группы (Group)

2. Список «Продукты заказчиков»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Заказчик	Наименование заказчика	Строка (Single line of text)	-
2.	Продукт	Код продукта из списка «Продукты»	Внешние данные (External Data)	-
3.	Статус ограничения	Указывается текущий статус ограничения	Выбор (Choice)	Используется; Не используется

3. Список «Исключения»

Номер	Атрибут	Пояснение	Тип данных	Возможные значения
1.	Вид МД	Тип контента	Строка (Single line of text)	-

Приложение И – Определение относительных весов показателей качества

Таблица П8.1 – Определение относительных весов показателей качества процесса проектирования и разработки в части документации на ПС

Наименование показателя	Эксперт					Сумма рангов	Отклонения от средней суммы	Квадрат отклонения	Вес показателя
	1	2	3	4	5				
Среднее количество ошибок, найденных на этапе тестирования	1	1	2	1	1	6	-6,5	35,75	0,12
Среднее количество ошибок, найденных на этапе контрольной инсталляции	3	3	1	3	2	12	-0,5	-0,25	0,24
Среднее количество сообщений об ошибках в документации, полученных от заказчиков	4	4	4	4	4	20	7,5	63,75	0,40
Среднее количество сообщений с вопросами по эксплуатации ПС, полученных от заказчиков	2	2	3	2	3	12	-0,5	-0,25	0,24
Сумма								99	1,00
Коэффициент конкордации (W)	0,792								

Таблица П8.2 – Определение относительных весов показателей качества процесса сопровождения

Наименование показателя	Эксперт					Сумма рангов	Отклонения от средней суммы	Квадрат отклонения	Вес показателя
	1	2	3	4	5				
Среднее время на анализ и выработку решения по сообщению с замечанием по постановке задачи (категория Р)	4	3	4	3	4	18	3	12	0,24
Среднее время на анализ и выработку решения по сообщению с ошибкой в компонентах ПС (категория S)	5	5	5	5	5	25	10	110	0,33
Среднее время на анализ и выработку решения по сообщению с ошибкой в документации (категория D)	3	4	3	4	3	17	2	6	0,23

Продолжение таблицы П8.2

Наименование показателя	Эксперт					Сумма рангов	Отклонения от средней суммы	Квадрат отклонения	Вес показателя
	1	2	3	4	5				
Среднее время на анализ и ответ по сообщению с вопросами по эксплуатации ПС (категория Q)	1	1	1	1	2	6	-9	72	0,08
Среднее время на анализ и выработку решения по сообщению с замечанием, связанным с процессом инсталляции подсистем продукта (категория I)	2	2	2	2	1	9	9	90	0,12
Сумма								200	1,00
Коэффициент конкордации (W)	0,8								

Приложение К – Свидетельство о государственной регистрации

программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014613465

Система управления знаниями о программной продукции

Правообладатель: *Дмитриев Павел Игоревич (RU)*

Автор: *Дмитриев Павел Игоревич (RU)*

Заявка № **2014611043**

Дата поступления **12 февраля 2014 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **26 марта 2014 г.**



*Руководитель Федеральной службы
по интеллектуальной собственности*

Б.П. Симонов

Рисунок К.1 – Свидетельство о государственной регистрации программы для ЭВМ