



А. А. ЦВЕТКОВ

Теория и практика  
бизнес-анализа в ИТ

Том I

Учебное пособие

ИНСТИТУТ ПРОГРАММНЫХ СИСТЕМ РАН

**А. А. Цветков**

# Теория и практика бизнес-анализа в ИТ

Том I

*Учебное пособие*



Москва  
Берлин  
2019

УДК 339.1:658(075)  
ББК 65.29я7

*Решением Ученого Совета Института программных систем  
им. А. К. Айламазяна РАН от 12 декабря 2018 г.  
учебное пособие рекомендовано к печати*

Рецензенты:

*Александрова Ирина Алексеевна, кандидат технических наук;  
Нестейода Николай Николаевич, доктор физико-математических наук,*

В оформлении обложки использовалась гравюра А. Дюрера «Меланхолия».

**Цветков, А. А.**

Е 67 Теория и практика бизнес-анализа : учебное пособие. В 2 т.  
Т. I / А. А. Цветков. – Москва ; Берлин : Директ-Медиа, 2019. – 150 с.  
DOI: 10.23681/500835

ISBN 978-5-4475-8152-7

Разработка информационных и автоматизированных систем является сложным технологическим процессом, который требует участия в нем группы специалистов, выполняющих различные роли, и, имеющих различные специализации. В настоящее время сложилась парадоксальная ситуация, при которой, несмотря на множество нормативно-справочной информации, которая разбросана в разных документах (законодательных, регламентирующих, рекомендующих), в каждой организации по-своему трактуют, чем именно должен заниматься тот или иной специалист, входящий в группу разработки.

Настоящая книга «Теория и практика бизнес-анализа в ИТ» является первым из двух томов, в которой подробно описана роль бизнес-аналитика в группе разработки, включая описание того, ЧТО, КАК, ГДЕ, с КЕМ, КОГДА и ПОЧЕМУ он должен делать на протяжении жизненного цикла информационной или автоматизированной системы. Приводится подробное описание инструментов бизнес-аналитика: нотации описания модели бизнес-процессов (BPMN) и нотации моделей «Сущность-связь» (ERD) для описания данных. Нотации приводятся в соответствии с оригинальными документами Object Management Group (OMG), т. е. документами международного консорциума, на основе которых формируются международные стандарты ИСО/МЭК.

Каждый раздел книги снабжен заданиями для повторения и усвоения изученного материала.

В приложении приводится справочник по полной нотации BPMN.

В основу книги положен практический опыт разработки аналитических моделей автором и его коллегами, а также курс лекций, читавшийся студентам в рамках курса «Информационные системы». Учебное пособие публикуется в авторской редакции.

УДК 339.1:658(075)  
ББК 65.29я7

ISBN 978-5-4475-8152-7

© Цветков А. А., текст, 2019

© Издательство «Директ-Медиа», макет, оформление, 2019

# Оглавление

Предисловие автора .....	5
Как читать эту книгу.....	10
Глава 1. Введение в разработку ИС.....	12
а. Понимание термина «Информационная система» .....	12
б. Классификация ИС .....	17
с. Нормативно-справочная информация.....	18
д. Выводы.....	24
е. Вопросы и задачи для повторения материала .....	24
Глава 2 Методики разработки ИС .....	25
а. Диаграмма Захмана .....	25
б. Модели жизненного цикла.....	34
с. Выводы.....	49
д. Вопросы и задачи для повторения материала .....	51
Глава 3. Теоретические основы нотаций моделирования.....	53
Глава 4. Нотация моделирования бизнес-процессов (BPMN) .....	59
а. Введение в нотацию BPMN .....	59
б. «Правильные слова» в BPMN.....	61
с. Базовые элементы BPMN.....	63
д. Исполняемый частный (внутренний) бизнес-процесс. Пример.....	68
е. Неисполняемый частный (внутренний) бизнес-процесс. Пример.....	69
ф. Публичный бизнес-процесс. Пример .....	70
г. Хореография. Пример.....	71

h. Подмодель «Соглашения». Пример .....	72
i. Методика моделирования в нотации BPMN .....	75
j. Выводы .....	103
k. Вопросы и задачи для повторения материала.....	104
Глава 5. Нотация моделей сущность-связь (ERD) .....	106
a. Краткое введение в теорию БД.....	106
b. Нотация модели «Сущность-связь».....	107
c. Типы моделей ERD.....	118
d. Выводы или «Как разработать ERD».....	123
e. Вопросы и задачи для повторения материала.....	124
Заключение к Тому I .....	126
Приложение 1. Элементы нотации BPMN.....	127
Список терминов и сокращений .....	145
Список литературы .....	146
Об авторе.....	149

## ПРЕДИСЛОВИЕ АВТОРА

Революцией во многих областях человеческой деятельности или, как скажет профессиональный бизнес-аналитик, в различных «предметных областях» стало массовое появление персональных компьютеров.

Автор еще помнит времена, когда для того, чтобы рассчитать свою задачу приходилось делать множество вещей, о которых многие современные пользователи даже не догадываются: нужно было математизировать свою задачу (самому, если умеешь, или обратиться к специалисту, как тогда говорили – постановщику), нужно было написать программный код (самому, если умеешь, или обратиться к программисту), нужно было написать задание для электронной вычислительной машины (далее – ЭВМ) (самому, если умеешь, или обратиться к специалисту – обычно с этим справлялись сами программисты, но были и отдельные специалисты – системщики), а еще записаться в очередь на машинное время, дождаться результатов, которые с первого раза были с ошибками, и опять пойти по всем кругам компьютерного ада... Т. е., если специалист был, например, биологом или статистиком, то ему приходилось найти достаточно убедительные аргументы, чтобы убедить себя и руководство – мне действительно необходимо в своих работах использовать компьютерные вычисления.

Первые, действительно массовые персональные компьютеры (далее – ПК) по нынешним меркам содержали не так много инструментов (простые текстовый редактор и электронная таблица, система управления базой данных, редакторы кода и компиляторы для различных языков, которых тоже было не так много – C/C++, Pascal, ...). Но это уже было настоящим чудом: прямо перед тобой устройство, которое заменяет большую ЭВМ, и ... делай, что хочешь.

Но ПК сыграли и плохую шутку с пользователями. Если раньше нужно было сто раз подумать о том, ЧТО и КАК ты будешь делать, то теперь можно было сразу переходить к увлекательнейшему занятию: писать код. Это действительно увлекательно: изучить язык программирования сильно проще, чем, например, английский или китайский, написать несколько

строк кода и вот оно: на экране интерфейс пользователя, можно нажимать кнопки, что-то там происходит, ... Начинаешь чувствовать себя гением или даже Создателем: вот мир, который ты создал, и который живет по твоим законам. Но это было до поры, пока задачки были простые и однопользовательские.

А дальше стало веселее: стали доступны сети, появилась потребность делать распределенные приложения уровня отдела, предприятия, холдинга, ... Нужна единая архитектура, бизнес-модели, системные модели, ... Одним словом – ДОКУМЕНТАЦИЯ. И не просто документация, а реальный план того, что будет сделано или уже сделано. И документация, которая будет понятна всем специалистам, которые работают в области разработки информационных систем (далее – ИС) и программного обеспечения (далее – ПО), входящего в состав ИС. Но господа специалисты по кодированию, которые никогда и не думали документировать свой код<sup>1</sup>, и, решившие, что они главные в процессе создании ПО, уже создали в голове потенциальных заказчиков устойчивый стереотип: разработчик ИС и ПО – это некто, который сидит за ПК и пишет код. А еще какие-то сотрудники, которые после кодировщика задокументируют его труды и издадут, в лучшем случае, руководство системного администратора и руководство пользователя.

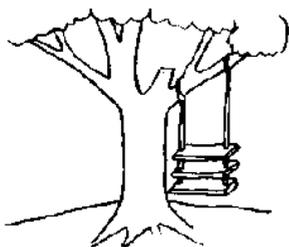
К чему это приводит отражено на известном многим поколениям программистов рисунке (см. ниже). Смеются, но ... делают по-своему.

Конечно, существуют документы, в которых написано, КАКИЕ документы нужно разрабатывать, чтобы разработка была успешной – это основные ГОСТы 34-й и 19-й серий. Но в настоящее время есть «маленькая» особенность. Если в СССР в ГОСТах писали «Несоблюдение преследуется по закону», то теперь ГОСТ «рекомендует», но разве, что за исключением ГОСТов, которые используют военные – там все строго: если в ГОСТ написано, то это обязательно к применению. Вот эту «маленькую» особенность и используют нынешние разработ-

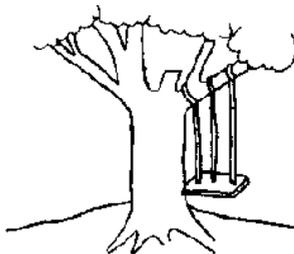
---

<sup>1</sup> Кстати, многие бывшие «наши» специалисты, которые решили укрепить собой компьютерную отрасль за рубежом, были страшно удивлены, что их увольняют за отсутствие документирования их деятельности.

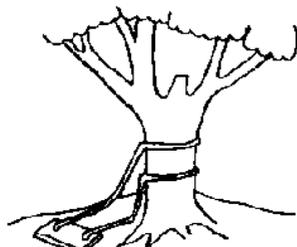
чки, как лазейку, чтобы делать не так, как надо, а как удобно тем, кто пишет код.



1. Как было предложено организатором разработки



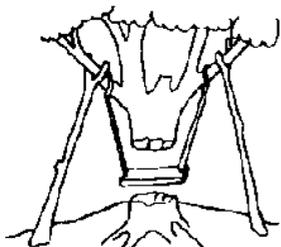
2. Как было описано в техническом задании



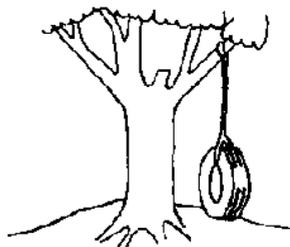
3. Как было спроектировано ведущим системным специалистом



4. Как было реализовано программистами



5. Как было внедрено



6. Чего хотел пользователь

Рисунок 1 – Процесс разработки [1]<sup>2</sup>

<sup>2</sup> Не хотел вставлять этот рисунок – он уже всем порядком надоел. Но... ничего не меняется.

Можно ли себе представить ситуацию в машиностроении или электронике, когда разработчик и конструктор разрабатывают чертежи не в соответствии со стандартами (российскими и международными), а так, как они сами понимают (сами придумывают обозначения и определяют набор чертежей и документов), а рабочий на станке делает детали или собирает компоненты электронного прибора так, как ему удобно, или так, как ему хочется (а эти умники потом сделают чертежи, посмотрев на мою деталь)? Правда смешно?

Тогда почему в информационных технологиях (далее – ИТ) мы, к сожалению, слишком часто наблюдаем такую картину: токарь (ой, простите, кодировщик) создает нечто так, как он умеет или понимает, а инженеры (опять прошу прощения – архитекторы, аналитики) вынуждены свои документы подгонять под это творение (и опять же: кто как понимает)?

Автор постарался в этой книге обобщить опыт лучших отечественных и зарубежных практик, а также свой многолетний опыт и навыки, полученные еще во времена СССР, в области ИТ. Нельзя объять необъятное, поэтому основной упор делается на теории и практике той деятельности, которой занимается бизнес-аналитик (в 70-е – 80-е годы XX века эта роль называлась «постановщик»): место бизнес-аналитика в команде разработки, современные нотации для моделей процессов и структур, документация, которая должна быть сформирована по результатам работы.

Как построена структура книги?

Во-первых, книга задумана, как учебное пособие, т. е. должна содержать изучаемый материал и контрольные вопросы для самопроверки. Но признаюсь сразу: вопросы, которые присутствуют в конце каждой главы, не простые. Это скорее можно назвать маленькими исследовательскими задачами. Для бизнес-аналитика крайне важно постоянно тренировать свои знания, умения и навыки на реальных или самим придуманных задачах, как музыканту – постоянно играть или потеряешь навыки. За основу взят курс лекций, который читался студентам университета.

Во-вторых, содержание книги разбито на несколько глав, каждая из которых содержит анонс того, что читатель узнает, изучив главу, непосредственно изучаемый материал, контрольные вопросы для самопроверки изученного материала.

В заключение хочу поблагодарить ООО «Директ-Медиа» за предоставленную возможность публикации настоящей книги.

Отдельная благодарность коллегам, которые приняли участие в обсуждении и рецензировании книги.

И большущая благодарность жене Елене, которая постоянно вдохновляла на написание этой работы.

## КАК ЧИТАТЬ ЭТУ КНИГУ

Сверхзадача, которую ставил перед собой автор, состоит в том, чтобы показать – в чем состоит роль бизнес-аналитика и какие задачи стоят перед ним в процессе разработки и создания ИС.

Для решения этой сверхзадачи главы книги, как кажется автору, построены в определенном логическом порядке, который при первом чтении не стоит нарушать – только в этом случае у читателя может сложиться верное представление о том, ЧТО должно интересовать бизнес-аналитика, КАК получить нужную информацию, ГДЕ получать информацию, КТО может дать информацию, КОГДА бизнес-аналитику нужно выполнять свои процессы и ПОЧЕМУ он должен это делать.

Понятное дело, что читать учебную и научную литературу не так весело и интересно, как детектив или интересный роман. Поэтому предлагаю некоторый компромисс: первый раз прочесть эту книгу, как детектив (только не заглядывайте сразу в конец, чтобы узнать, чем дело кончится), второй раз – прочесть, как роман – вдумчиво и со вкусом, изучая внимательно «главных героев» (нотации моделирования) и «ситуации» (процессы, которые должен выполнять бизнес-аналитик), связанные с «главными героями», третий раз – можно не читать все, но использовать как справочник для повседневной работы.

В конце книги любознательный читатель может обнаружить множество ссылок на первоисточники. Это не для демонстрации того, что автор много читает – это для читателя. В одной книге подробно описать какую-либо проблематику крайне проблематично – тем более, что коллеги, работающие в одной с тобой области, достаточно подробно исследовали некоторые аспекты проблемы. Убедительная просьба и совет – изучите литературу, которая приведена в ссылках! Тем более, что эти знания понадобятся вам при выполнении заданий, которые есть в конце каждой главы.

Кстати, по поводу заданий в конце каждой главы. Предполагается, что читатель уже ВНИМАТЕЛЬНО изучил содержание главы, изучил дополнительную литературу и готов выполнить маленькую научно-исследовательскую работу. Нужно иметь ввиду, что работа бизнес-аналитика состоит из двух главных

составляющих: графическое моделирование в одной из нотаций и формализованное описание этих моделей. Т.е. для выполнения заданий читателю понадобится какой-либо инструмент, который позволяет выполнить моделирование. Автор не будет обсуждать достоинства и недостатки существующих инструментальных средств – главное это то, чтобы инструмент моделирования поддерживал в полном объеме те нотации, которые разработаны в Object Management Group, т.е. организации-авторе спецификаций BPMN, UML, ERD, и, которые являются де-факто международными стандартами (для части из нотаций, например, BPMN, разработан и стандарт ИСО/МЭК, который полностью повторяет спецификацию Object Management Group). Могу только порекомендовать использовать для моделирования такой инструмент, как Visual Paradigm<sup>3</sup> (см. <https://www.visual-paradigm.com/>), на котором автор сделал большую часть иллюстративного материала книги.

Моделирование в бизнес-аналитике держится на «трех словах» или нотациях: нотация бизнес-процессов и модели (BPMN), универсальный язык моделирования (UML), диаграммы «сущность-связь» (ERD). Объем и сложность материала предопределили то, как построен материал настоящей книги: предполагается издание двух томов под общим названием «Теория и практика бизнес-анализа в ИТ». В настоящем томе читателю предстоит освоить следующие разделы:

- «Глава 1. Введение в разработку ИС»;
- «Глава 2. Методики разработки ИС»;
- «Глава 3. Теоретические основы нотаций моделирования»;
- «Глава 4. Нотация моделирования бизнес-процессов (BPMN)»;
- «Глава 5. Нотация моделей сущность-связь (ERD)».

Второй том будет посвящен моделированию в нотации UML, формированию пользовательских сценариев и прототипированию пользовательских интерфейсов.

---

<sup>3</sup> Компания Visual Paradigm дает пробную версию, которая обеспечивает полную функциональность продукта, на один месяц. Далее можно приобрести лицензию на нужную вам для работы версию или ограничиться

# ГЛАВА 1

## ВВЕДЕНИЕ В РАЗРАБОТКУ ИС

«Прежде чем спорить, давайте договоримся о терминах».

Франсуа Мари Аруэ (Вольтер)

- **Понимание термина «Информационная система»**
- **Классификация ИС**
- **Нормативно-справочная информация**
- **Выводы**

### **А. Понимание термина «Информационная система»**

Существует старая байка, когда некий философ начинает, как говорят в народе, «философствовать»: «Вы спрашиваете меня, что такое наука? А давайте разберемся в том, что такое «спрашиваете», что такое «Вы», что такое «Я», наконец, что такое «Что»...». Ну, так можно далеко уйти, подменяя пространными рассуждениями основную тему разговора. Основной темой книги является, если кто забыл, «Теория и практика бизнес-анализа в информационных технологиях», т. е. ключевыми понятиями для нас являются «бизнес-анализ» (далее – БА) и «информационные технологии» (далее – ИТ). И, конечно же, то, посредством чего ИТ могут быть реализованы – «информационная система».

На пояснении того, что есть БА и кто такие специалисты, которые выполняют этот процесс, т. е. «бизнес-аналитики», мы остановимся ниже. Начнем с однозначного понимания терминов «технология», «информация» и «система».

*Термин «Технология»* происходит от древнегреческих слов: τέχνη – «искусство», «мастерство», «умение» и λόγος – «слово», «мысль», «смысл», «понятие». Можно было бы интерпретиро-

вать, как «описание искусства»<sup>4</sup>, или «описание мастерства», или «описание умения».

Существует несколько формулировок термина «Технология». Например, в [19] используется следующая формулировка: «Технология – совокупность методов и инструментов для достижения желаемого результата». Более старая формулировка звучит так (см. [5]): «Технология – совокупность производственных методов и процессов в определенной отрасли производства, а также научное описание способов производства». Но это формулировки на все случаи жизни, которые подходят и для металлообработки, и для пищевой промышленности, и для ... Нас же интересует конкретная формулировка для термина «Информационные технологии». Если с формулировкой «технологии» более или менее ясно, то давайте теперь разберемся с термином «информация». Но, учитывая, что в книге рассматривается очень специфическое направление, которое, в значительной степени, должно соответствовать нормативным требованиям (как минимум – стандартам) – будем полагаться не на общефилософские рассуждения, а именно: на нормативно-справочную информацию (далее – НСИ<sup>5</sup>).

Согласно ГОСТ Р 43.0.4–2009 «Информационное обеспечение техники и операторской деятельности. Информация в технической деятельности. Общие положения» (см. п. 3.4 ГОСТ [24]), *«Информация – это совокупность каких-либо сигнальных*

---

<sup>4</sup> Шутки шутками, а в ИТ, стараниями некоторых коллег, которые работают не в соответствии с требованиями технологии, а так, как они понимают, получаем не процесс с предсказуемым результатом, а, в случае если коллеге повезет, нечто на грани шаманства или, как они считают, «на грани искусства». Но ведь, если заказчик обращается к токарю, то он вправе ожидать получить деталь в соответствии с чертежом, а не «произведение искусства».

<sup>5</sup> Надо отметить, что достаточно устоявшимся аналогичным термином является словосочетание «нормативно-справочная документация», которое сокращается, как НСД. Но, к сожалению, авторы многих официальных документов практически не отслеживают уже используемые аббревиатуры. Вот и получилось, что НСД чаще стало ассоциироваться с «несанкционированным доступом», т.е. это сокращение присвоили себе специалисты в области информационной безопасности. Ну с ними трудно спорить ... Но в остальных случаях... Трудно объяснить, каким образом к аббревиатуре МО, которое ранее однозначно ассоциировалось с Министерством Обороны, теперь «прицепили» еще и «Московская область», и «Медицинская организация», и еще Бог знает что. Коллеги, поосторожнее с выбором аббревиатур.

*воздействий (чувственного, чувственно-содержательного (перцептивно-семантического) восприятия) на субъект».*

В ГОСТ Р 43.2.1–2007 «Информационное обеспечение техники и операторской деятельности. Язык операторской деятельности. Общие положения» (см. п. 3.12 [25]) определение информации звучит несколько по-иному: «Информация – это фиксируемое каким-либо способом воспринимаемое отражение реального мира».

Еще одна формулировка из ГОСТ Р 51275–2006 «Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения» (см. п. 2.1(1) в [26]): «Информация – это сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления».

И еще. ГОСТ Р 50922–2006 «Защита информации. Основные термины и определения» (см. п. 1 Приложение А [27]): «Информация – это сведения (сообщения, данные) независимо от формы их представления».

Для объединения, перечисленных выше терминов «Технология» и «Информация», в один, а именно: «Информационные технологии», воспользуемся формулировкой, которая приведена в Федеральном законе от 27.07.2006 N 149-ФЗ (ред. от 18.12.2018) «Об информации, информационных технологиях и о защите информации» (см. [28]).

### **Определение термина «Информационные технологии»**

«Информационные технологии – процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов»

Нравится кому-то или нет такая формулировка, но это закон, а, как говорили еще в Древнем Риме: «*Dura lex, sed lex*!» («Закон суров, но это закон»). Кстати, читатель, если Вы только определяетесь с выбором: заниматься или нет ИТ, то следует задуматься, что эта сфера деятельность очень тесно связана с законодательством и специалисту в ИТ, кроме знаний по спе-

циальности, нужно хорошенько знать законы в этой области, а еще законы той предметной области, в которой Вы проводите автоматизацию. И знать лучше, чем знает юрист. В случае чего – все стрелки переведут на Вас – «это делал вот этот субъект...».

Теперь согласуем понимание термина «система».

Можно было бы ограничиться каким-нибудь стандартным определением «система», например: «Термин *Система* происходит от древнегреческого слова *σύστημα* “целое, составленное из частей; соединение” – т. е. множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство», сославшись на авторитетный источник [9].

Но и пытаться собрать и обобщить все известные определения понятия «система», было бы непростой задачей, которая потребовала бы отдельного исследования. Существуют, например, еще определения:

- «совокупность элементов, находящихся в определённых отношениях друг с другом и со средой» (см. [1]);
- «комбинация взаимодействующих элементов, организованных для достижения одной или нескольких поставленных целей» ... (см. [29]).

На взгляд автора, если рассмотреть множество определений термина «Система», то, в конечном итоге, они повторяют друг друга, но часто накладывается терминология предметной области, в которой работает автор определения.

Для целей, которые обозначены в данной работе, автор предлагает ограничиться определением, которое приводится в [9] (см. текст выше).

Теперь определим очередное слияние терминов: «Информация» и «Система», т. е. определим термин «Информационная система», воспользовавшись формулировкой из [28].

**Определение термина «Информационная система» (ГОСТ)**  
«Информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств»

В то же время, в Стандарте специальности «Прикладная информатика» дается более расширенное понимание того, что такое «Информационная система»:

**Определение термина «Информационная система» (Стандарт специальности)**

«Информационная система – это совокупность:

- функциональных и информационных процессов конкретной предметной области;
- средств и методов сбора, хранения, анализа, обработки и передачи информации, зависящих от специфики области применения;
- методов управления процессами решения функциональных задач, а также информационными, материальными и денежными потоками в предметной области»

В настоящей книге под ИС будем понимать систему в широком и узком смысле:

- ИС в широком смысле – это совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией;
- ИС в узком смысле – это подмножество компонентов (ИС в широком смысле), включающее базы данных (далее – БД), системы управления БД (далее – СУБД) и специализированные прикладные программы. ИС в узком смысле рассматривают как программно-аппаратную систему, предназначенную для автоматизации целенаправленной деятельности конечных пользователей, обеспечивающую, в соответствии с заложенной в нее логикой обработки, возможность получения, модификации и хранения информации.

## **В. Классификация ИС**

Будем классифицировать ИС по различным признакам: по архитектуре, по характеру обработки данных, по сфере применения, по масштабности.

**Классификация по архитектуре включает следующие ИС:**

- настольные, или локальные ИС, в которых все компоненты (БД, СУБД, клиентские приложения) находятся на одном компьютере;
- распределённые ИС, в которых компоненты распределены по нескольким компьютерам.

Распределённые ИС, в свою очередь, разделяют на:

- файл-серверные ИС;
- клиент-серверные ИС.

В файл-серверных ИС разделяемый файл с данными база данных находится на файловом сервере, а клиентские приложения находятся на рабочих станциях.

В клиент-серверных ИС БД и СУБД находятся на сервере, а на рабочих станциях находятся клиентские приложения.

В свою очередь, клиент-серверные ИС разделяют на двухзвенные и многозвенные.

В двухзвенных (англ. two-tier) ИС всего два типа «звеньев»: сервер БД, на котором находятся БД и СУБД, и рабочие станции, на которых находятся клиентские приложения. Клиентские приложения обращаются к СУБД напрямую.

**Классификация по характеру обработки данных включает следующие ИС:**

- Информационно-справочные, или информационно-поисковые ИС, в которых нет сложных алгоритмов обработки данных, а целью системы является поиск и выдача информации в удобном виде;
- ИС обработки данных, или решающие ИС, в которых данные подвергаются обработке по сложным алгоритмам. К таким системам в первую очередь относят автоматизированные системы управления и системы поддержки принятия решений.

*Поскольку ИС создаются для удовлетворения информационных потребностей в рамках конкретной предметной области, то каждой предметной области (сфере применения) соответствует свой тип ИС (см. примеры ниже):*

- Экономическая информационная система – информационная система, предназначенная для выполнения функций управления на предприятии;
- Медицинская информационная система – информационная система, предназначенная для использования в лечебно-профилактическом учреждении;
- Географическая информационная система – информационная система, обеспечивающая сбор, хранение, обработку, доступ, отображение и распространение пространственных данных.

**Классификация по масштабности включает следующие ИС:**

- Персональная ИС предназначена для решения некоторого круга задач одного пользователя;
- Групповая ИС ориентирована на коллективное использование информации группой пользователей;
- Корпоративная ИС в идеале охватывает все информационные процессы целого предприятия, достигая их полной согласованности, отсутствия избыточности и наличие прозрачности. Такие системы иногда называют системами комплексной автоматизации предприятия.

### **С. Нормативно-справочная информация**

Итак, мы определились с терминами и классификацией ИС. Но этого совсем недостаточно для того, чтобы проводить серьезные разработки. Как уже отмечалось выше, разработка ИС это не только знание ИТ, но и знание множества документов: законов, указов, кодексов, нормативов, рекомендаций, стандартов, ... Незнание всего этого комплекса документов может привести, как минимум, к ситуации схожей в известном произведении «Приключения Буратино»: «А, так сеньор Разработчик не знает

нормативно-справочной информации...». А дальше... Надеюсь, читатель знаком с упомянутым произведением и может продолжить – отдаю на откуп вашей фантазии. А, как максимум, у «сеньора Разработчика» будут серьезные проблемы с уголовным кодексом. Но тонкости юриспруденции выходят за рамки этой работы. Если автор не испугал будущих коллег, то давайте продолжим изучать ту НСИ, которая может понадобиться в дальнейшем.

Далее мы рассмотрим следующие виды документации:

- регламентация деятельности;
- общие требования к составу ИС и связям между ее компонентами, а также к ее качеству;
- виды, состав и содержание проектной и программной документации.

Классификация НСИ может быть следующей:

- По виду регламентации: закон, постановление, стандарт, руководящий документ, положение, инструкция и т. п.;
- По статусу регламентирующего документа: международный, национальный, отраслевой, организации;
- По области действия документа: заказчик, исполнитель, проект;
- По объекту регламентации.

Следует иметь в виду, что часть регламентирующих документов является обязательной к исполнению: Федеральные законы РФ, Постановления Правительства РФ, Приказы регуляторов, законодательство субъектов РФ (если оно не противоречит законодательству РФ) и т. д.; часть документов носит рекомендательный характер и не требует жесткого исполнения: рекомендации министерств, ведомств, регуляторов и т. д.; часть документов носят особый характер – это стандарты – могут быть обязательными к исполнению (например, военные стандарты, если их использует разработчик ИС – к исполнению обязательны!), могут быть рекомендованы к исполнению, но, если в требованиях заказчика ИС четко и однозначно прописано, что стандарт или группа стандартов обязательны – никуда не денешься и нужно исполнять.

Рассмотрим стандарты подробнее.

### **Определение термина «Стандарт»**

«Стандарт – это документ, устанавливающий требования, спецификации, руководящие принципы или характеристики, в соответствии с которыми могут использоваться материалы, продукты, процессы и услуги, которые подходят для этих целей» (см. официальный сайт Международной организации по стандартизации (International Organization for Standardization или ISO): <http://www.iso.org/iso/ru/home/standards.htm>)

Стандарты могут быть классифицированы по области действия:

- Международные стандарты – рассчитаны на применение во всем мире, главным образом, в международных проектах. Обычно это стандарты ISO и Международная Комиссия по Электротехнике (далее – ИЕС);
- Государственные стандарты, например, в Российской Федерации это документы, именуемые «Государственный стандарт» или ГОСТ;
- Отраслевые стандарты, например, в Российской Федерации это документы, именуемые «Отраслевой стандарт» или ОСТ;
- Стандарт уровня организации (предприятия, учреждения и т. п.), например, в Российской Федерации это документы, именуемые «Стандарт организации» или СТО.

В контексте настоящей книги, наибольший интерес будут представлять стандарты групп 34 и 19.

Но внимательный читатель может задать вопрос: стандарты группы ГОСТ 34\* рассматривают «автоматизированные системы», а не «информационные системы»; тогда почему, если рассматриваются ИС, мы должны использовать стандарт на автоматизированные системы?

В ГОСТ 34.003–90 (см. [4]) в п. 1.1 введено следующее определение: автоматизированная система – это «Система, состоящая из персонала и комплекса средств автоматизации его деятельности, реа-

*лизирующая информационную технологию выполнения установленных функций.*

*Примечания:*

*1. В зависимости от вида деятельности выделяют, например, следующие виды АС: автоматизированные системы управления (АСУ), системы автоматизированного проектирования (САПР), автоматизированные системы научных исследований (АСНИ) и др.*

*2. В зависимости от вида управляемого объекта (процесса) АСУ делят, например, на АСУ технологическими процессами (АСУТП), АСУ предприятиями (АСУП) и т. д.».*

Т. е. автоматизированная система (далее – АС) – это более общее понятие, которое включает в себя, как компонент, информационную технологию. Для определенности далее по тексту мы будем рассматривать именно ИТ и не будем рассматривать такие компоненты, как, например, система управления станком с числовым программным управлением. Информация, информация, ничего кроме информации. Как, например, в автоматизированной системе управления предприятием (далее – АСУП).

ИТ пока не настолько глубоко используют методы искусственного интеллекта (далее – ИИ), чтобы подменить директора человека на директора робота. Вот, если весь управленческий персонал будет заменен на роботов или еще что-нибудь этакое – вот тогда будем писать не о ИС, а о АС.

Ниже перечислим состав стандартов, входящих в группу ГОСТ 34\*. (см. Таблица 1).

Таблица 1 – Перечень ГОСТов, входящих в группу 34\*

Наименование ГОСТ 34*	Краткое описание ГОСТ
ГОСТ 34.003–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения»	Настоящий стандарт устанавливает термины и определения основных понятий в области автоматизированных систем (АС) и распространяется на АС, используемые в различных сферах деятельности (управление, исследование, проектирование и т. п., включая их сочетание), содержанием которых является переработка информации.
ГОСТ 34.201–89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем»	Настоящий стандарт распространяется на автоматизированные системы (АС), используемые в различных сферах деятельности (управление, исследование, проектирование и т. п.), включая их сочетание, и устанавливает виды, наименование, комплектность и обозначение документов, разрабатываемых на стадиях создания АС
ГОСТ 34.320–96 «Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы»	Настоящий стандарт устанавливает основные понятия и термины концептуальных схем и информационных баз, охватывающих разработку, описание и применение концептуальных схем и информационных баз, манипулирования информацией, а также описание и реализацию информационного процесса
ГОСТ 34.321–96 «Информационная технология. Система стандартов по базам данных. Эталонная модель управления данными»	Настоящий стандарт устанавливает эталонную модель управления данными
ГОСТ 34.601–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания»	Настоящий стандарт распространяется на автоматизированные системы (АС), используемые в различных видах деятельности (исследование, проектирование, управление и т. п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях (далее – организациях). Стандарт устанавливает стадии и этапы создания АС
ГОСТ 34.602–89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»	Настоящий стандарт распространяется на автоматизированные системы (АС) для автоматизации различных видов деятельности (управление, проектирование, исследование и т. п.), включая их сочетания, и устанавливает состав, содержание, правила оформления документа «Техническое задание на создание (развитие или модернизацию) системы»

Наименование ГОСТ 34*	Краткое описание ГОСТ
ГОСТ 34.603–92 «Информационная технология. Виды испытаний автоматизированных систем»	Настоящий стандарт распространяется на автоматизированные системы (АС), используемые в различных видах деятельности (исследование, проектирование, управление и т. п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях (далее – организациях). Стандарт устанавливает виды испытаний АС и общие требования к их проведению
РД 50–34.698–90 «Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные документы. Требования к содержанию документов»	Настоящие методические указания распространяются на автоматизированные системы (АС), используемые в различных сферах деятельности (управление, исследование, проектирование и т. п.), включая их сочетание, и устанавливают требования к содержанию документов, разрабатываемых при создании АС

Если читатель изучит, как минимум, серию ГОСТ 34\*, то станет совершенно ясно – разработка ИС это, главным образом, серьезная работа с НСИ и разработка новых документов, без которых разработать, создать и в дальнейшем поддерживать ИС очень неблагоприятное занятие и работа коллектива разработчиков превращается в хаос. Из названия книги – «Теория и практика бизнес-анализа в ИТ» – можно предположить, что главная цель книги описать процессы, которые выполняет бизнес-аналитик и его место в команде разработки. Забегая вперед скажу, что бизнес-аналитику не нужно писать всю эту громаду документов: далее мы очертим круг задач, которые должен решать бизнес-аналитик. Но сейчас, если кратко: бизнес-аналитик в коллективе разработчиков – это внутренний заказчик, который должен понять, ЧТО нужно заказчику ИС и донести это до всей группы, а потом осуществлять авторский надзор за тем, ЧТО пытаются делать остальные члены команды<sup>6</sup>.

---

<sup>6</sup> К сожалению, в большинстве случаев наблюдается ситуация, при которой руководители разработки (не буду вдаваться в причины) считают, что главный в группе разработки – это программист, а все остальные должны «конспектировать» и подгонять под ГОСТ его творчество. Вот и имеем то, что имеем....

## **Д. Выводы**

1. В настоящем разделе мы договорились о единстве терминологии, т. е., надеюсь, не будет разнотолков и подмены понятий таких важных терминов, как «информация», «система», «технология» и словосочетаний, которые из них можно образовать;

2. Выяснили классификацию ИС;

3. Определились с НСИ и, главным образом, определились с той, которая необходима в первую очередь в работе бизнес-аналитика;

4. Скользь упомянули обязанности бизнес-аналитика и декларируем важнейшее положение о том, что «бизнес-аналитик – это внутренний заказчик для остальных членов команды».

## **Е. Вопросы и задачи для повторения материала**

1. Перечислите группы НСИ, которые обязательны к исполнению в процессе разработки ИС. Приведите примеры, которые обязательны для разработчиков ИС.

2. Необходимо разработать ИС, которая содержит информацию по истории физики. Заказчик требует строгого соответствия группе ГОСТ 34\*. Какие этапы разработки необходимо выполнить и какие документы должны быть обязательно связаны с каждым этапом?

3. Необходимо разработать ИС, которая должна автоматизировать деятельность поликлиники, включая данные о сотрудниках и пациентах. Заказчик требует строгого соответствия группе ГОСТ 34\*. Какое обязательное требование отсутствует, но должно обязательно выполняться?

## ГЛАВА 2

# МЕТОДИКИ РАЗРАБОТКИ ИС

- **Диаграмма Захмана**
- **Модели жизненного цикла**
- **Выводы**

### **А. Диаграмма Захмана**

Очень ехидное замечание сделал Г. Буч (см. [2]): «Хотя это и кажется комичным, многие компании, разрабатывающие программное обеспечение, хотят создать небоскреб, в то же время как их подход к делу очень напоминает строительство собачьей будки». Что тут добавить? Печальный факт, с которым сталкиваешься постоянно, но еще печальнее то, что многие руководители и сотрудники компаний-разработчиков твердо уверены в своей правоте и переубедить их крайне трудно. Главный аргумент: «А так делают во всем мире...». Надеюсь, что читатель достаточно вдумчивый и не страдает манией величия – так делаю я, значит, так делают во всем мире...

В предыдущих главах мы договорились о терминах и выяснили какой НСИ будем пользоваться. Но пока непонятно, как эти знания можно использовать при разработке конкретной ИС. Каким образом можно эти знания связать с конкретной задачей и наполнить смыслом? Не секрет, что сложные системы, в т. ч. ИС, содержат огромное количество условий, правил и связей, которые невозможно охватить в одной модели или документе.

Блестящий выход из этого положения был предложен Джоном Захманом (см. [18]). Д. Захман предложил модель (или, как часто ее называют – диаграмма), которая позволяет описать архитектуру предприятия или, говоря словами Д. Захмана: «Набор элементарных описательных артефактов, которые составляют инфраструктуру знаний о предприятии».

Диаграмма Захмана имеет вид таблицы (или матрицы), имеющей шесть строк и шесть столбцов, как показано ниже (см. Таблица 2).

Таблица 2 – Диаграмма Захмана. Общее представление

	<b>ЧТО</b>	<b>КАК</b>	<b>ГДЕ</b>	<b>КТО</b>	<b>КОГДА</b>	<b>ПОЧЕМУ</b>	
<b>Границы контекста</b>	Инвентаризация ↓ Список объектов	Процессы идентификации ↓ Список процессов	Процессы определения ↓ Список размещения	Выявление организаций ↓ Список организаций	Выявление событий ↓ Список событий	Выявление мотиваций ↓ Список мотиваций	<b>Стратеги с ролью теоретика</b>
<b>Видение бизнеса</b>	Определение ресурсов ↓ Список бизнес-сущностей и связей	Определение процессов ↓ Изменения деятельности, входные данные для деятельности	Определение размещения ↓ Расположение бизнеса, связи между размещениями	Определение организации ↓ Бизнес-роли, бизнес-деятельность	Определение событий ↓ Цикл деятельности, периоды в деятельности	Определение мотивации ↓ Ограничения деятельности, средства ведения деятельности	<b>Управленцы с ролью владельца</b>
<b>Системная логика</b>	Учет представлений ↓ Системные сущности, системные связи	Системное представление ↓ Преобразованные системы, системный вход	Представление о размещениях ↓ Размещение системы, системные связи	Организационное представление ↓ Роли в системе, системные процессы	Представление событий ↓ Системные циклы, периоды в системе	Представление мотивации ↓ Системное ограничение, системная деятельность	<b>Архитекторы с ролью конструктора</b>
<b>Физическая реализация логики</b>	Спецификации учета ↓ Технологические сущности и связи	Спецификация системы ↓ Преобразованные технологии, технология входа	Спецификация размещения ↓ Технологическое размещение и связи	Организационная спецификация ↓ Технологическое представление ролей и процессов	Спецификация событий ↓ Технологическое представление циклов и периодов	Спецификация мотиваций ↓ Технологическое ограничение и деятельность	<b>Инженеры с ролью разработчика</b>

<b>Компоненты</b>	Спецификация конфигураций ↓ Компоненты сущностей и связей	Конфигурация процесса ↓ Компоненты преобразования и входа	Конфигурация размещения ↓ Размещение и связи между компонентами	Конфигурация организации ↓ Компоненты ролей и процессов	Конфигурация событий ↓ Компоненты циклов и событий	Конфигурация мотиваций ↓ Ограничения компонентов и деятельность компонентов	<b>Техники с ролью внедренца</b>
<b>Классы операций</b>	Спецификация экземпляров ↓ Сущности и связи процессов	Экземпляры процессов ↓ Преобразованные процессы и входа	Варианты размещения ↓ Размещение и связи процессов	Организационные варианты ↓ Роли и деятельность процессов	Варианты событий ↓ Циклы деятельности	Варианты мотиваций ↓ Ограничения деятельности и компонентов	<b>Работники с ролью пользователя</b>
	<b>Набор описаний</b>	<b>Процесс преобразования</b>	<b>Узловые размещения</b>	<b>Организационные роли</b>	<b>Временные периоды</b>	<b>Мотивационные причины</b>	

Строки представляют, если смещаться по ним сверху вниз, представляют собой эволюцию архитектуры организации: от существующей архитектуры к той архитектуре, которая будет создана в процессе разработки и создания (или модернизации) ИС. Последняя строка («Классы операций») представляет собой, по сути, описание созданной архитектуры, с которой взаимодействуют сущности «Работники с ролью пользователя». Каждая из строк предваряется кратким описанием того, что мы описываем, а завершается некоторым общим представлением того, кто описывает (или помогает описывать).

Итак, рассмотрим построчно диаграмму Захмана.

1. **Границы контекста** – строка, которая отражает (или должна отражать) видение тех, кто назван в диаграмме «Стратеги с ролью теоретика», т. е., если не столь высокопарно, исполнители процессов компании, которые хорошо знает свою деятельность или, как говорят специалисты в области ИС, «носитель знаний» или эксперт в какой-либо предметной области (далее – ПрО). Желательно, чтобы такой носитель знаний не только отлично знал свою ПрО, но и видел перспективы развития архитектуры организации в рамках своей компетенции (и не боялся фантазировать);

2. **Видение бизнеса** – строка, которая отражает (или должна отражать) видение управленцев, т. е. тех, кому от владельца всех процессов в организации (это, как правило, директор) делегированы права на какой-либо процесс или группу процессов. С точки зрения специалистов в области ИС, это тоже носитель знаний, но который видит архитектуру более глобально и не вдаётся в тонкости каждой деятельности, описание которой получено при описании контекста. Естественно, что к группе, описывающей видение бизнеса, относится владелец всех процессов в компании – руководитель компании;

3. **Системная логика** – строка, которая отражает видение «Архитектора с ролью конструктора» или, другими словами – системного аналитика (что вернее отражает, на взгляд автора, суть), т. е. это уже специалист компании-разработчика ИС, который на основе информации, полученной в строках «Границы контекста» и «Видение бизнеса», формирует представление

системы, понятное разработчикам: программистам, сетевым специалистам, разработчикам БД и т. п.;

4. **Физическая реализация логики** – строка, которая отражает видение группы «Инженеры с ролью разработчика», специалистов компании разработчика ИС, которые на основе документов, подготовленных системным аналитиком (или архитектором): функциональные требования, частные технические задания, спецификации, структура БД, архитектура аппаратных средств и др., проводят разработку программного обеспечения, включая функции и триггеры для БД, устанавливаемую в СУБД, формируют физическую структуру аппаратных средств и т. п. Результатом этой деятельности должен быть комплект эксплуатационных документов и руководств, передаваемых внедренцам ИС;

5. **Компоненты** – строка, которая отражает деятельность специалистов компании-разработчика ИС, а именно – «Техники с ролью внедренца». Задачей внедренцев является проведение работ по внедрению ИС в опытную эксплуатацию с последующей передачей ИС заказчику в промышленную эксплуатацию, включая комплект документов, который должен обеспечить эту эксплуатацию;

6. **Классы операция** – строка, которая отражает деятельность группы специалистов заказчика «Работники с ролью пользователя», и которая в качестве содержимого ячеек диаграммы Захмана предполагает соответствующий набор эксплуатационных документов и руководств.

Важно заметить следующее. Как правило (автору не пришлось встречать исключения), хотя строки «Границы контекста» и «Видение бизнеса» должен был бы заполнять заказчик, но, среди сотрудников заказчика не существует специалистов, которые владеют профессиональными знаниями в области ИТ. Поэтому и эта задача ложится на сотрудников компании-разработчика ИС, а именно: бизнес-аналитиков, которые должны в кратчайшие сроки освоить теоретические знания из ПроО заказчика настолько, чтобы специалисты заказчика начали воспринимать бизнес-аналитика, как своего коллегу. И не просто освоить знания ПроО, но и подготовить документы, которые требуются в соответствии с диаграммой Захмана, убедиться у

специалистов заказчика, что знания и требования верно отражены в моделях и описаниях. Т. е., по сути, бизнес-аналитик становится представителем интересов заказчика в области разработки ИС или осуществляет авторский надзор за выполняемыми работами. Но и это еще не все.

Бизнес-аналитик обязан представить своим коллегам по разработке ИС комплект документов, который позволит «Архитектору с ролью конструктора» начать разработку, не вникая в тонкости ПрО заказчика.

Эта особенность отмечена на диаграмме Захмана путем выделения первых двух строк – зона ответственности бизнес-аналитика, которую мы подробнее рассмотрим ниже.

Теперь рассмотрим столбцы диаграммы Захмана. По сути – это шесть вопросов, на которые нужно получить ответ в каждой из строк с различным уровнем абстракции и детализации:

1. «ЧТО» – имеются в виду данные, которые собираются, обрабатываются, хранятся в процессе деятельности организации, и которые будут обрабатываться в создаваемой ИС;

2. «КАК» – это процессы, которые используют, формируют и преобразовывают данные;

3. «ГДЕ» – место, где выполняются процессы и связи между ними;

4. «КТО» – участники, которые выполняют процесс, т. е. организации, структуры, люди;

5. «КОГДА» – это события, которые приводят к началу выполнения процессов, включая цикличность, периодичность, типы событий и т. п.;

6. «ПОЧЕМУ» – мотивация, т. е. ограничения и цели, которые определяют деятельность или функционирование ИС.

Каждая из ячеек в столбце содержит соответствующие данные, которые должны эволюционировать при переходе с одного уровня на другой. Понятное дело, что разместить все данные, которые имеют отношение к данной ячейке, в этой ячейке – невозможно. Т. е. ячейки диаграммы Захмана – это скорее ссылочный механизм, который содержит список соответствующих документов, которые полностью отвечают на вопрос для соответствующего уровня.

Как отмечено выше, в контексте настоящей работы нас интересует деятельность бизнес-аналитика, т. е. более пристально рассмотрим первые две строки: «Границы контекста» и «Видение бизнеса».

Идеальным случаем можно считать ситуацию, при которой участники, заполняющие ячейки диаграммы Захмана, являются не только сотрудниками организации, выполняющими свои процессы, но и владеют знаниями бизнес-аналитика, т. е. могут разработать модели данных, процессов и описать их в виде документов (согласно внутренним нормативным документам или рекомендациям/требованиям стандартов). Почти идеальным случаем можно считать вариант, при котором бизнес-аналитик владеет теоретическими (и, возможно, практическими) знаниями в ПрО, для которой создается ИС.

Рассмотрим более вероятный случай: компания-разработчик выделяет своего специалиста (бизнес-аналитика), который должен получить от носителей знаний в ПрО знания об объекте автоматизации, адекватно усвоить эти знания и формализовать их в виде моделей, описаний, документов, которые нужно верифицировать у заказчика и убедиться в том, что разработанные документы однозначно понятны в группе разработки.

Из-за тезиса «должен получить от носителей знаний в ПрО знания об объекте автоматизации» (или, как иногда говорят – проинтервьюировать) в некоторых компаниях-разработчиках полагают, что лучшим бизнес-аналитиком можно считать специалиста, который пришел из журналистики или имел такой опыт.

Если придерживаться стандартов из группы ГОСТ 34\*, а конкретнее – ГОСТ 34.601–90 «Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания», то строку диаграммы Захмана «Границы контекста» можно отнести к стадии «Формирование требований к АС», включающей этапы работ: «Обследование объекта и обоснование необходимости создания АС», «Формирование требований пользователя к АС», «Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-технического задания)». Вторая строка диаграммы Захмана будет соответствовать стадиям

«Разработка концепции АС» (включающей этапы: «Изучение объекта», «Проведение необходимых научно-исследовательских работ», «Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя», «Оформление отчёта о выполненной работе») и «Техническое задание», включающего этап «Разработка и утверждение технического задания на создание АС»).

Т. е. результатом работы бизнес-аналитика должен быть комплект документов, разработанный в соответствии с формализацией диаграммы Захмана (строки «Границы контекста» и «Видение бизнеса») и ГОСТ 34.601–90, который, как предполагается, должен включать всю необходимую информацию для заполнения ячеек на уровне «Системная логика» участником проектной группы «Архитектор с ролью конструктора» (или, иными словами – системным аналитиком).

Но функции бизнес-аналитика на этом не завершаются – на продолжении всего жизненного цикла ИС он должен осуществлять авторский надзор за соответствием получаемого результата требования документов, которые были сформированы при заполнении строк диаграммы Захмана «Границы контекста» и «Видение бизнеса».

Информационные потоки, которыми обмениваются между собой участники заполнения ячеек диаграммы Захмана для строк «Границы контекста» и «Видение бизнеса», а также потребитель результатов процесса «Архитектор» показаны ниже (см. Рисунок 2).

Модель представлена в нотации, которую мы рассмотрим ниже (см. ГЛАВА 3), но автор надеется, что, с небольшими пояснениями, данное иллюстрация достаточно прозрачно иллюстрирует выше сказанное.

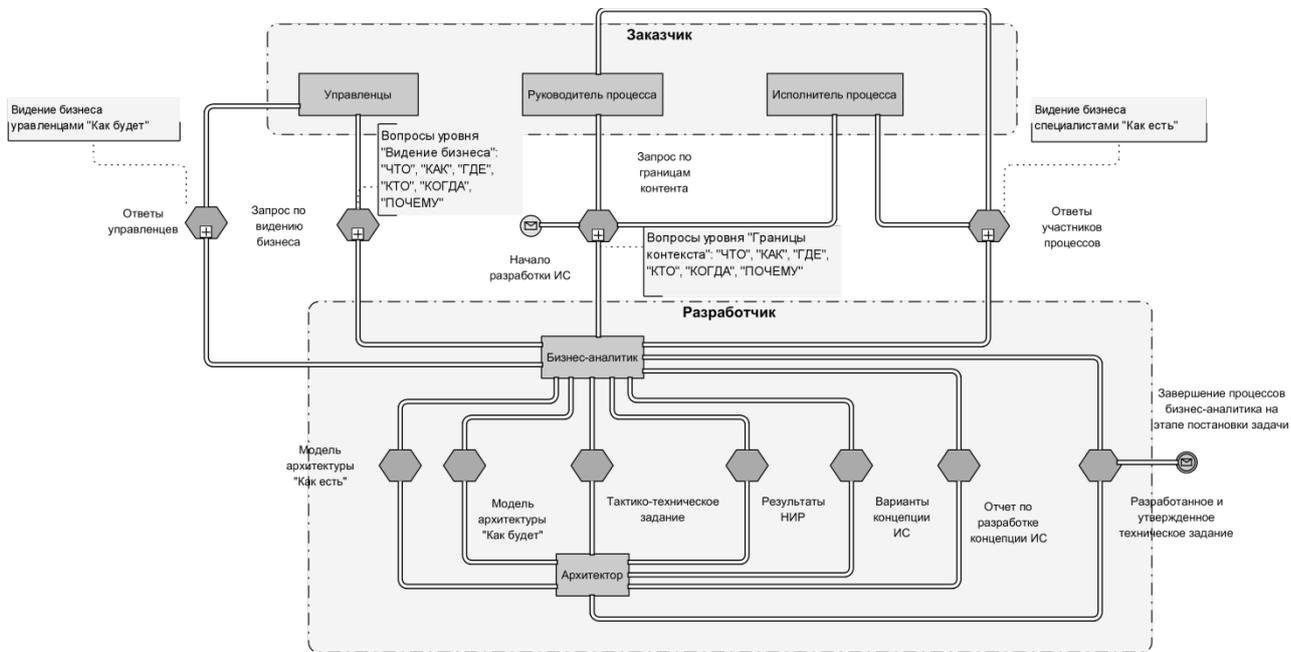


Рисунок 2 – Участники и информационные потоки при формировании технического задания. Модель типа «Соглашения» в нотации BPMN

Символом  обозначено начало процессов по сбору информации об объекте информатизации, т. е. вопросы к сотрудникам заказчика, которые являются носителями знаний по тем процессам, которые в текущий момент выполняются у заказчика. Это могут быть, как руководители процесса (т. е. сотрудники, которым делегированы права на управление процессом), так и рядовые исполнители процесса. Символом  обозначена информация, которой обмениваются участники. После получения информации от носителей знаний, бизнес-аналитик анализирует данные и формирует ряд моделей (см. ГЛАВА 4 и ГЛАВА 5), которые описывают состояние архитектуры «Как есть».

Далее бизнес-аналитик проводит сбор информации у управленцев и формирует модели «Как будет» (см. ГЛАВА 4 и ГЛАВА 5).

На основании, сформированных моделей и описаний к ним, бизнес-аналитик разрабатывает документы в соответствии с ГОСТ 34.601–90 (и связанными с ним ГОСТами), включая техническое задание (далее – ТЗ) на ИС. Разработанный комплект документов передается для выполнения процессов по заполнению диаграммы Захмана остальным членам группы разработки. Как отмечалось выше, бизнес-аналитик продолжает работу, выполняя функции авторского надзора. Тот факт, что процессы бизнес-аналитика не завершены, в модели отмечен символом  (промежуточное событие).

## **В. Модели жизненного цикла**

Последовательность процессов членов группы разработки зависит от выбранной модели жизненного цикла для разрабатываемой ИС. Далее рассмотрим эти модели. Но для начала вспомним тезис Вольтера: «Прежде чем спорить, давайте договоримся о терминах». На самом деле, если мы хотим разрабатывать ИС «по-взрослому», а не по принципу: «сейчас умные ребята чего-то там сделают, а потом разберемся», то следует учесть такой документ, как ГОСТ Р ИСО/МЭК 12207–2010 «Информационная технология. Системная и программная

инженерия. Процессы жизненного цикла программных средств» (см. [30]), в котором дается определение жизненного цикла (далее – ЖЦ), модели ЖЦ и рассматриваются процессы, обеспечивающие ЖЦ.

**Определение понятия «Жизненный цикл»** – развитие системы, продукта, услуги, проекта или других изготовленных человеком объектов, начиная со стадии разработки концепции и заканчивая прекращением применения.

**Определение понятия «Модель жизненного цикла»** – структура процессов и действий, связанных с жизненным циклом, организуемых в стадии, которые также служат в качестве общей ссылки для установления связей и взаимопонимания сторон.

Сами процессы, которые должны выполняться на протяжении всего ЖЦ, представлены в ГОСТ Р ИСО/МЭК 12207–2010 в следующем виде (см. Рисунок 3).

Среди множества процессов выделим те, с которыми непосредственно связана деятельность бизнес-аналитика.

Деятельность, которая связана с группой «Процессы соглашения», предполагает участие бизнес-аналитика как в «Процессе приобретения», так и в «Процессе поставки».

Среди «Процессов приобретения» существует подпроцесс «Подготовка требований и условий», который соответствует первым двум строкам в диаграмме Захмана, и, который, как отмечалось выше, находится в зоне ответственности бизнес-аналитика. Т. е., хотя согласно ГОСТ Р ИСО/МЭК 12207–2010 предполагается некоторый абсолют – «Заказчик сам выполняет процесс», в действительности эти задачи поручаются либо бизнес-аналитику разработчика, либо приглашенному бизнес-аналитику, который выполнит цикл по подготовке документации, включая техническое задание, для передачи разработчику.

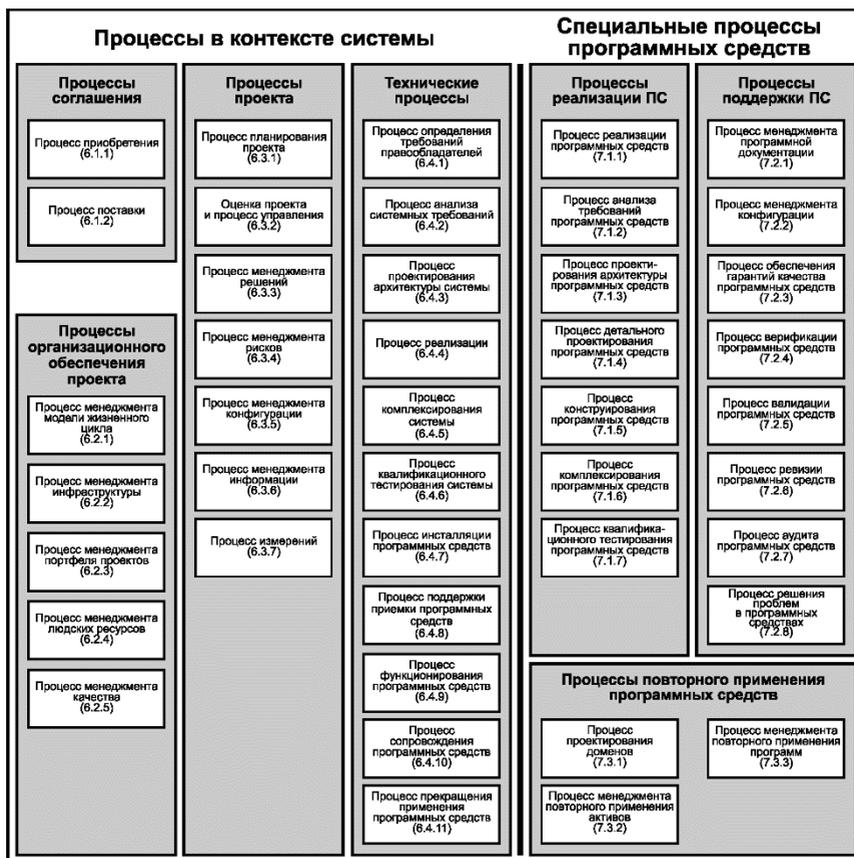


Рисунок 3 – Процессы жизненного цикла

«Процесс поставки» включает подпроцессы «Оценка требований и условий», «Планирование управления проектом», «Планирование работ» и «Выполнение и контроль работ», при выполнении которых также необходимо участие бизнес-аналитика.

По результатам подпроцесса «Оценка требований и условий» бизнес-аналитиком должны быть подготовлены отчеты или пояснительные записки, в которых должна быть представлена модель, интерпретирующая требования ТЗ, и проведена оценка реализуемости поставленной задачи. По результатам,

возможно, понадобятся дополнительные обсуждения с заказчиком по переформированию требований к ИС. Или разработчик может отказаться от разработки ИС.

Как отмечалось выше, бизнес-аналитик должен выступать в роли внутреннего заказчика для всей остальной группы, участвующей в разработке, а поэтому его роль в подпроцессе «Планирование управления проектом» состоит в том, чтобы определить приоритеты разработки, которые должны быть отмечены при планировании.

Подпроцесс «Планирование работ» также предполагает участие бизнес-аналитика, т. к. в процессе проведения работ по проекту требуется постоянный авторский надзор за соответствием выполненных работ тому, что разработано на этапе бизнес-анализа. Сюда входит планирование:

- аудита соответствия разработки ТЗ функциональным требованиям, решениям по структуре БД и концептам интерфейса;
- проведение внутреннего тестирования;
- согласование изменений с разработчиками и т. п.

Собственно, подпроцесс «Выполнение и контроль работ» является практическим воплощением работ, запланированных в подпроцессе «Планирование работ».

Конкретная реализация процессов и подпроцессов ЖЦ зависит от выбранной модели ЖЦ. Ниже проведем краткий обзор основных моделей ЖЦ, которые используются сейчас.

### ***Каскадная (или водопадная) модель ЖЦ***

Каскадная модель ЖЦ предполагает, что каждый из этапов разработки выполняется в строго фиксированном порядке и следующий этап может начаться тогда, когда завершен предыдущий. Пример каскадной модели представлен ниже (см. Рисунок 4).

К основным достоинствам каскадной модели можно отнести (при условии, что соблюдаются все требования стандартов, регламентов и др. НСИ):

1. Простота планирования проекта;
2. Понятность заказчику;
3. Полный комплект документов на каждом этапе;

4. Возможно поручить каждый из этапов различным командам разработчиков.

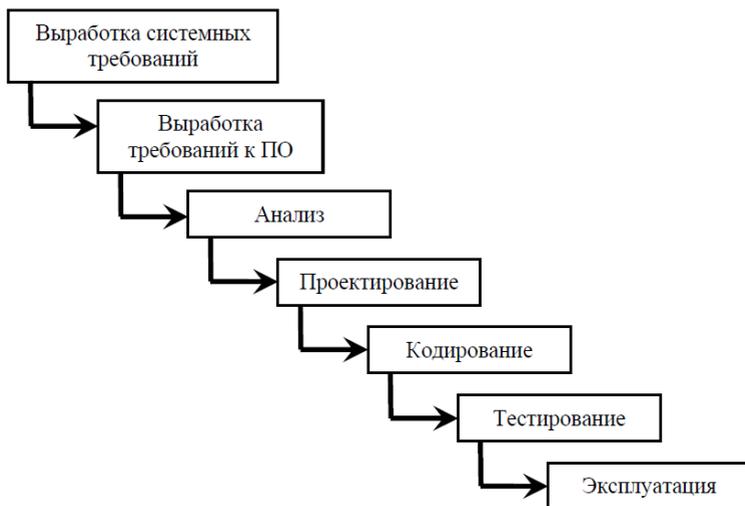


Рисунок 4 – Пример каскадной модели

*Обратите внимание на важный тезис «при условии, что соблюдаются все требования стандартов, регламентов и др. НСИ». К сожалению, во многих компаниях наблюдается очень плохая тенденция: даже при хорошем кодировании очень плохая документация или ее полное отсутствие. Бизнес-аналитик, как внутренний заказчик, обязан требовать, чтобы документация соответствовала требованиям по полноте и достоверности. Если его требования не выполняются, то это серьезный повод для размышлений: «А в том ли коллективе я работаю?». Это касается не только каскадной модели.*

Недостатки каскадной модели:

1. Существенные задержки в получении результатов;
2. Невозможность распараллеливания работ;
3. Рискованные инвестиции в проекты, которые выполняются в соответствии с каскадной моделью.

### ***Каскадная модель ЖЦ с промежуточным контролем***

Каскадная модель ЖЦ с промежуточным контролем предполагает, что после каждого этапа существует возможность возврата к предыдущим этапам, но с последующим прохождением всех этапов, которые расположены ниже этапа, в котором была проведена коррекция. Пример такой модели показан ниже (см. Рисунок 5). Что это значит?

Допустим, что на этапе «Кодирование» выясняется – были допущены ошибки на этапе «Выработка требований к ПО» и необходимо внести изменения после согласования с заказчиком. По завершению внесения изменений, необходимо опять пройти этапы «Анализ», «Проектирование» с учетом внесенных изменений на этапе «Выработка требований к ПО» и только после этого начать выполнять этап «Кодирование».

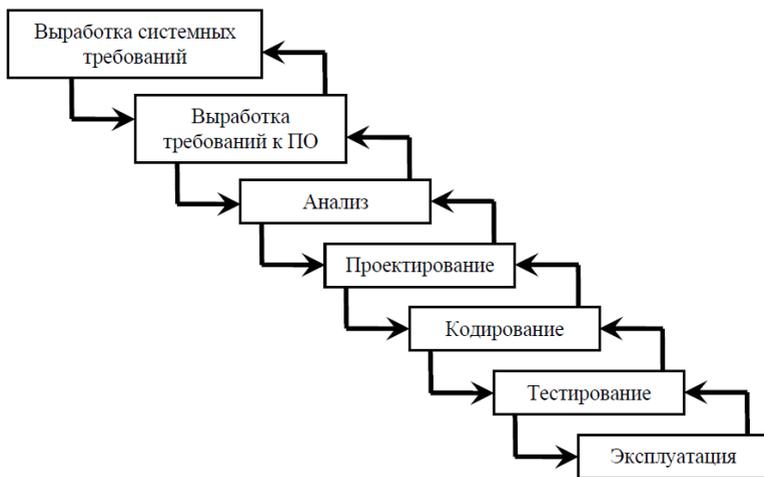


Рисунок 5 – Пример каскадной модели ЖЦ с промежуточным контролем

Данной модели присущи все достоинства и недостатки каскадной модели ЖЦ, но появляется возможность в случае ошибок – возврат к этапу, на котором ошибка совершена, но придется учитывать изменения на всех последующих этапах.

Симбиозом каскадной модели ЖЦ и каскадной модели ЖЦ с промежуточным контролем можно считать итеративную или инкрементальную модель (известно несколько таких моделей), которые предполагают разбиение создаваемой системы на набор частей, и, которые разрабатываются с помощью нескольких последовательных проходов всех этапов или их части. Пример такой модели показан ниже (см. Рисунок 6).

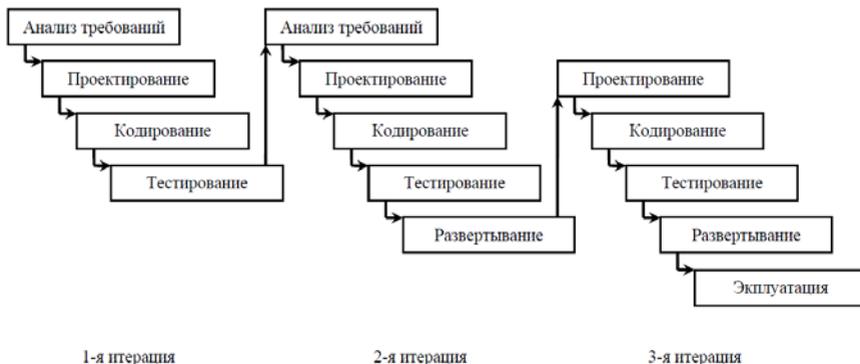


Рисунок 6 – Пример итеративной модели ЖЦ

На первой итерации разрабатывается часть системы, не зависящая от других. При этом большая часть или даже полный цикл работ проходит на эту часть, затем оцениваются результаты и на следующей итерации либо первая часть переделывается, либо разрабатывается следующая, которая может зависеть от первой, либо как-то совмещается доработка первой части с добавлением новых функций. В результате на каждой итерации можно анализировать промежуточные результаты работ и реакцию на них всех заинтересованных лиц, включая пользователей, и вносить корректирующие изменения на следующих итерациях. Каждая итерация может содержать полный набор видов деятельности: от анализа требований, до ввода в эксплуатацию очередной части ИС.

Развитием идеи итераций является **спиральная модель жизненного цикла ИС**.

Эта модель предлагает каждую итерацию начинать с выделения целей и планирования очередной итерации, определения основных альтернатив и ограничений при ее выполнении, их оценки, а также оценки возникающих рисков и определения способов избавления от них, а заканчивать итерацию оценкой результатов проведенных в ее рамках работ. Пример спиральной модели ЖЦ показан ниже (см. Рисунок 7).

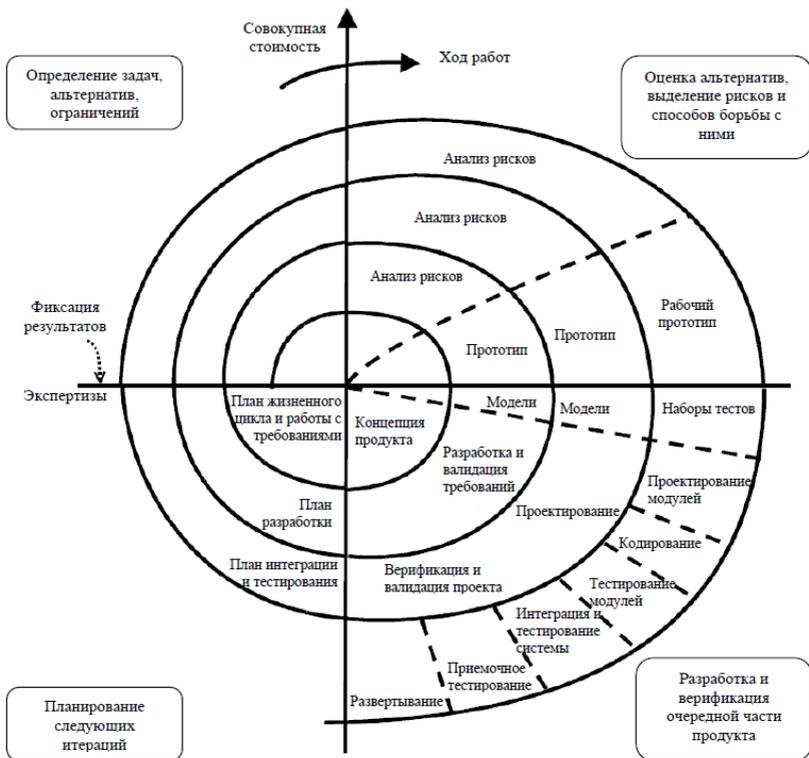


Рисунок 7 – Пример спиральной модели ЖЦ

Каждому циклу спирали соответствует такое же количество этапов, как в каскадной модели ЖЦ. К основным достоинствам

спиральной модели ЖЦ можно отнести (повторюсь: *при условии, что соблюдаются все требования стандартов, регламентов и др. НСИ*):

1. Простота учета изменившихся требований заказчика;
2. Элементы ИС интегрируются в пространство постепенно и непрерывно;
3. Возможность управления рисками, т. к. возможно оценивать риски на каждом обороте спирали. Риски снижаются с каждым новым циклом спирали;
4. Гибкость в управлении проектом;
5. Возможность повторного использования компонентов, полученных в предыдущих итерациях;
6. Вероятность успешного завершения проекта возрастает, т. к. выявляются и устраняются ошибки при каждом цикле спирали.

К недостаткам спиральной модели ЖЦ часто относят то, что эта модель имеет тенденцию к бесконечности – заказчик может постоянно вносить изменения, которые необходимо учитывать разработчику. Вот тут и нужно отметить ВАЖНЕЙШУЮ роль бизнес-аналитика: в большинстве случаев именно бизнес-аналитик готовит ТЗ (документ, который после согласования и подписания является законом и для заказчика, и для разработчика) и от того, как он его напишет, зависит «бесконечность» спиральной модели ЖЦ – правильно подготовленные требования, цели и задачи, которые должна решать ИС, состав функций и др. являются ограничениями модели, т. е. там должно быть четко описано ЧТО обрабатывает ИС, КАК обрабатывает, ГДЕ обрабатывает, КТО обрабатывает, КОГДА обрабатывает и ПОЧЕМУ обрабатывает. Требования заказчика сверх требований ТЗ – это уже требует обсуждений, т. к. новые требования, скорее всего, потребуют удорожания проекта: внесение дополнительного функционала в ИС потребует изменений в уже существующем решении, что повлечет необходимость увеличения временных и/или человеческих ресурсах. Искусство написания ТЗ сродни работе хорошего юриста – от его содержания зависит благополучие всей команды разработчиков.

На основе перечисленных выше подходов, ряд крупных компаний сформировали свои модели ЖЦ, в которых они

постарались учесть все достоинства прототипов и избавиться от их недостатков. Более подробно модели, которые были рассмотрены выше, можно найти, например, в [31].

Для примера рассмотрим модель ЖЦ, которую предлагает компания Microsoft.

Эта модель, построенная на двух взаимосвязанных наборах документов Microsoft Solutions Framework (или MSF) и Microsoft Operations Framework (или MOF), использует четкость целей и определяет моменты перехода от этапа к этапу из каскадной модели, а минимизацию рисков – из спиральной модели. При этом между собой идеологически увязаны две спирали: спираль разработки и спираль эксплуатации (см. Рисунок 8).

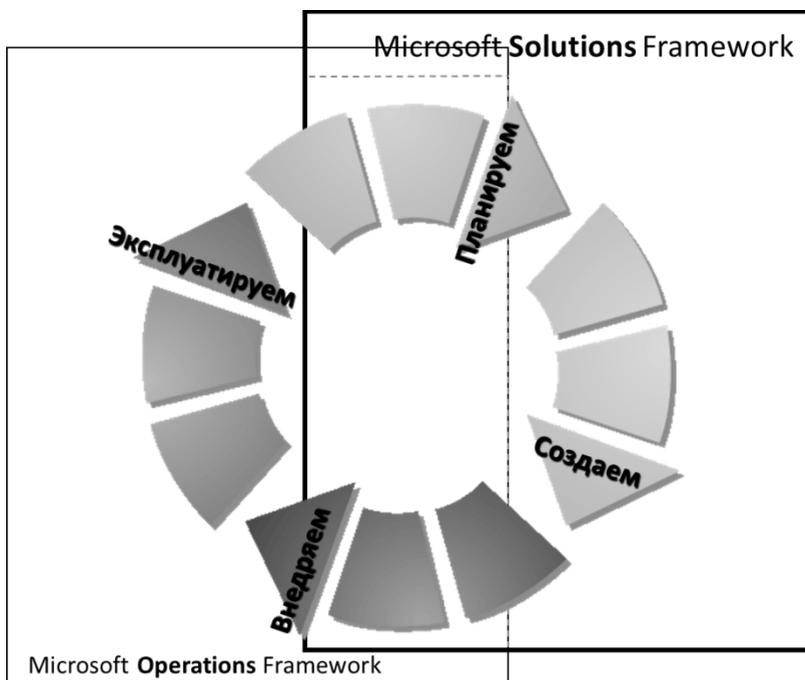


Рисунок 8 – Взаимосвязь MOF и MSF

Более обстоятельно познакомиться с MSF можно в [32]. Но, поскольку, ключевой темой данной работы является деятельность бизнес-аналитика, рассмотрим более подробно одну из моделей, которая входит в MSF, а именно: модель команды (см. Рисунок 9), которая, в отличие от многих других моделей, представляет команду без строгой иерархии. При этом, если рассматривать роль бизнес-аналитика, то его основной задачей является: понять потребности бизнес-заказчика и донести это понимание остальным членам команды. Кроме того, бизнес-аналитик «работает с пользователями и другими заинтересованными лицами, чтобы понимать их потребности и задачи, трансформировать их в конкретные определения, сценарии и требования к качеству, которые команда разработчиков будет использовать для построения приложения. Бизнес-аналитик определяет ожидания от функциональных возможностей системы и управляет ими. В проекте он представляет пользователей и участвует в управлении продуктом в том смысле, что постоянно отслеживает интересы пользователей и заказчиков проекта. И, наконец, бизнес-аналитики отвечают за обеспечение взаимодействия между разработчиками и пользователями» [32].



Рисунок 9 – Модель команды разработки в MSF (см. [32])

Согласно [32], основными действиями и операциями для бизнес-аналитика являются:

- Действия:
  - формулировка концепции проекта;
  - разработка требований к качеству;
  - создание сценария;
  - планирование итерации;
- **Операции при выполнении действия «Формулировка концепции проекта»:**
  - *написание концепции* – короткая и точная формулировка цели создания новой системы или улучшения существующей (см. Таблица 3);

Таблица 3 – Операция «Написание концепции»

Шаг операции	Описание шага
Обобщение исходных данных проекта	Написание одного-двух абзацев, которые содержат описание новой системы или улучшения существующей
Объяснение движущих факторов	Основные требования или выделенный интервал времени, которые управляют выпуском продукта
Определение пользователей	Определите тех пользователей, которые получают наибольшую пользу от системы
Определение основных достоинств	Предполагаемые достоинства новой или улучшенной системы

- *определение собирательных образов* – вымышленных объектов типа «Человек», представляющих группу пользователей. Собирательные образы используются для изучения потребностей целого сегмента пользователей путем рассмотрения характеристик одного вымышленного объекта (см. Таблица 4);

Таблица 4 – Операция «Определение собирательных образов»

Шаг операции	Описание шага
Определение ролей	Среди пользователей, определенных в концепции, необходимо выбрать группу пользователей, которые будут взаимодействовать с системой
Создание собирательного образа	Создание собирательного образа предполагает наличие реальных данных, полученных в процессе исследования объекта автоматизации

- *уточнение собирательных образов* – ревизия собирательных образов, которая возможна в процессе развития проекта (см. Таблица 5);

Таблица 5 – Операция «Уточнение собирательных образов»

Шаг операции	Описание шага
Определение отличий	Выявление сходства и различия в уровне знаний, способах использования, взаимодействии и задачах, имеющиеся между собирательным образом и реальным пользователем. Если нашлись такие отличия, необходимо разобраться, является ли пользователь тем, кого описывает собирательный образ
Обновление и передача изменений	Новые собирательные образы и изменения существующих

• **Операции при выполнении действия «Разработка требований к качеству»:**

- *определение требований к качеству путем «мозгового штурма»* – обсуждение аспектов системы, для которых в сценариях были пропущены необходимые требования к качеству. Список требований к качеству содержит нефункциональные требования к приложению, он же является списком ограничений функциональных возможностей системы. Бизнес-аналитик каждый раз переоценивает и корректирует список требований к качеству, когда в процессе тестирования выявляются новые требования, или, когда появляются изменения в проекте;

- *создание моментальных снимков деятельности* – это описание одного дня или короткого периода в жизни собирательного образа. При этом всегда описывается текущее состояние, без применения предлагаемой технологии, что помогает продемонстрировать, чем новая система или продукт будут полезны для пользователя;

- *определение приоритетов в списке требований к качеству* – это список требований к качеству, упорядоченных по приоритетам, включается в план итерации. Тот, в свою очередь, используется при планировании разработки для повышения эффективности использования ресурсов. При добавлении или удалении требований, а также при изменении потребностей пользователей

следует заново определить приоритеты списка требований к качеству (см. Таблица 6);

Таблица 6 – Операция «Определение приоритетов в списке требований к качеству»

Шаг операции	Описание шага
Обобщение общего приоритета	Каждому сценарию в списке назначается общий приоритет. В поле приоритета списка сценариев вписывается соответствующее числовое значение
Описание приоритетных сценариев	Используя шаблон описания сценария, необходимо контурно обрисовать наиболее приоритетные сценарии, снабдив их кратким описанием, но, в то же время, достаточно подробным, чтобы разработчики на его основании смогли сделать оценки
Передача запросов для выполнения оценок	Разработчикам отправляется запрос на получение общих оценок для самых приоритетных требований
Разбиение требований или изменение приоритетов	Если цена какого-либо требования к качеству превышает бюджет итерации, то необходимо попробовать разбить это требование

- *написание требований к качеству* – описания нефункциональных требований или ограничений на функциональные возможности системы. Требование к качеству должно быть таким, чтобы его можно было протестировать;

- *определение требований безопасности* – определяют уровень, до которого система будет защищать себя и ресурсы. Это могут быть конфиденциальные данные, требования закона или нематериальные активы, такие как репутация компании, торговые секреты или интеллектуальная собственность. Требования к системе безопасности должны быть конкретны, а для защищаемых ресурсов должна быть возможность проверки защиты;

- **Операции при выполнении действия «Создание сценария»:**

- *работа над сценариями методом «мозгового штурма»* – групповая деятельность над формированием списка сценариев для определения целей ИС. Список сценариев периодически пересматривается и корректируется;

- *определение приоритетов в списке сценариев* – в расчет принимаются как значимость сценария для пользователей, так и для

ИС в целом. Приоритеты определяют очередность реализации сценариев;

- *оценка задачи по разработке базы данных* – задача состоит в ограничении набора реализуемых функций, определении расписания и распределении приоритетов;

- *выполнение ретроспективного анализа* – позволяет бизнес-аналитику проанализировать свои правильные и ошибочные действия после завершения очередной итерации. Последняя итерация заканчивается ретроспективным анализом проекта;

- *оценка сценария* – выполняется для создания общей картины, которая помогает понять, сколько усилий потребуется для реализации сценария. Подобные оценки позволяют бизнес-аналитику расставлять приоритеты и регулировать внешние ожидания;

- *оценка требования к качеству* – служит для определения усилий, необходимых для их реализации. Подобные оценки позволяют бизнес-аналитику устанавливать приоритеты и регулировать внешние ожидания;

- *составление графика сценария* – разрабатывается и проходит первоначальное тестирование в рамках определенной итерации. В плане итерации отражено текущее понимание того, что в ней должно быть выполнено. План должен быть завершен до начала итерации. Первоначальный план создается на основе оценок, а затем уточняется по мере того, как сценарии и требования к качеству разбиваются на задачи;

- *составление графика реализации требований к качеству* – для реализации требований к качеству должен быть составлен график работ. Требования к качеству описывают нефункциональные требования, такие как, например, уровень производительности. С некоторыми требованиями не связаны задачи. Однако требования могут быть связаны со сценариями, в которых они реализуются. Таким образом, требования к качеству должны быть запланированы одновременно с соответствующими сценариями, чтобы обеспечить выполнение задач;

- *обзор целей* – контроль того, что после всех итераций, кроме нулевой, продукт находится в стабильном состоянии и готов к поставке;

- **Операции при выполнении действия «Планирование итерации»:**

- *определение продолжительности итерации* – включает в себя рассмотрение всех ключевых факторов, включая дату поставки продукта (если она определена), размер сценариев и их общее время. Обычно продолжительность итерации определяется в неделях, однако возможны и меньшие единицы времени. Продолжительность итераций определяется в начале проекта, а затем на основе этих данных определяется общая продолжительность проекта, в частности, эти данные должен использовать бизнес-аналитик при планировании своих операций;

- *планирование сценария* – формирование наброска плана итерации для сценариев;

- *планирования требования к качеству* – формирование наброска плана итерации для обеспечения требований к качеству;

- *планирование ресурсов на исправление дефектов* – резервирование ресурсов на исправление дефектов;

- *разбиение сценария на задачи* – формирование наброска разбиения сценария на задачи и оценка необходимых ресурсов;

- *разбиение требования к качеству на задачи* – позволяет выполнить оценку сложности и способствует лучшему пониманию разработчиками предстоящей работы.

## **С. Выводы**

В настоящем разделе мы обсуждали методики разработки ИС, т. е. выяснили КТО, ЧТО и КАК должен делать в процессе разработки ИС, но, главным образом, нас интересовал такой член команды разработки, как бизнес-аналитик. Итак, как следует из вышесказанного, бизнес-аналитик:

1. Может быть сотрудником компании-разработчика или быть сотрудником компании-заказчика (штатным или по договору). С точки зрения заказчика – бизнес-аналитик должен быть специалистом, осуществляющим авторский надзор, т. е. тем, кто от имени заказчика понимает пользовательскую архитектуру, создаваемой ИС, и ее место в архитектуре организации-заказчика;

2. Занимает промежуточное положение между функциональным заказчиком (т. е. группой специалистов, для которых создается ИС, и, которые будут выступать в роли пользователей ИС) и специалистами компании-разработчика, для которых бизнес-аналитик выступает в роли «внутреннего заказчика», а также экспертом в ПрО, в границах которой разрабатывается ИС;

3. Должен проводить предпроектное обследование объекта автоматизации (т. е. информационные бизнес-процессы в организации-заказчике), которое включает сбор информации о ПрО в соответствии с рекомендациями диаграммы Захмана: получение ответов на вопросы «ЧТО», «КАК», «ГДЕ», «КТО», «КОГДА», «ПОЧЕМУ»;

4. Обобщать полученные результаты и формировать отчеты (пояснительные записки);

5. Разрабатывать техническое задание, которое должно быть согласовано организацией-заказчиком и организацией-разработчиком ИС, что является подтверждением того, что заказчик согласен с тем, что требует ТЗ, а разработчик понимает то, что требует ТЗ;

6. Формировать концепцию проекта;

7. Формировать требования к качеству, включая требования к информационной безопасности;

8. Формировать сценарии использования ИС;

9. Участвовать в контроле качества разработки ИС;

10. Участвовать в приемо-сдаточных испытаниях.

Особенности разработки могут отличаться в зависимости от выбранной модели разработки: каскадная, спиральная или их комбинации, но взаимодействие между участниками проекта и бизнес-аналитиком в общем случае можно представить в общем случае, как на рисунке ниже (см. Рисунок 10).

Далее мы рассмотрим один из основных моментов в работе бизнес-аналитика: используемые нотации, которые позволяют ему моделировать бизнес-процессы «Как есть» и «Как будет».

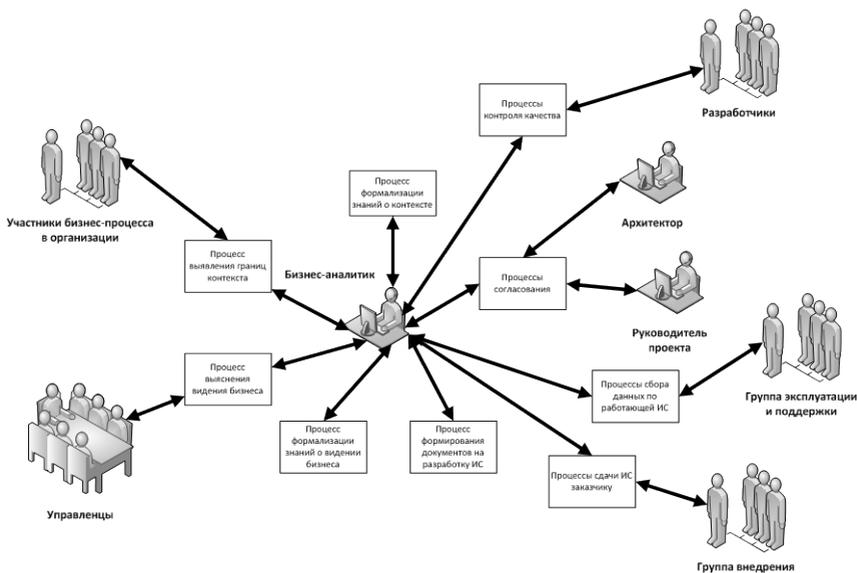


Рисунок 10 – Представление процессов бизнес-аналитика. Верхний уровень

## Д. Вопросы и задачи для повторения материала

1. Если в диаграмме Захмана поменять местами вопросы, что изменится?

Для ответа на вопрос представьте себе, что общаетесь с реальными людьми – с их поведенческими особенностями и характерами. Напишите протокол такого виртуального взаимодействия. По работе вам придется писать много документов, включая различные протоколы – тренируйтесь...

2. Рисунок 10 показывает процессы бизнес-аналитика. Свяжите представленные бизнес-процессы с ячейками диаграммы Захмана и каскадной моделью разработки (только для процессов бизнес-аналитика). Результаты представить в виде подробной модели, аналогичной той, что представлена выше (см. Рисунок 4). Перечислите документы, которые должны быть представлены в каждом из процессов. Если документы должны выполняться в соответствии с ГОСТ, то укажите, какой именно ГОСТ применим в данном случае.

3. Аналогично задаче 2, но используйте каскадную модель ЖЦ с промежуточным контролем (см. Рисунок 5).
4. Аналогично задаче 2, но используйте итеративную модель ЖЦ (см. Рисунок 6).
5. Аналогично задаче 2, но используйте спиральную модель ЖЦ (см. Рисунок 7).
6. По результатам задач 2 ... 5 сформируйте таблицу, в которой необходимо привести достоинства и недостатки каждого из подходов.

## ГЛАВА 3

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ НОТАЦИЙ МОДЕЛИРОВАНИЯ

Подходы к моделированию БП менялись на протяжении достаточно длительного времени и прошли несколько этапов или «волн» (см., например, работу [7]).

Первая и вторая волны, как правило, связаны с применением таких нотаций, как, например, «*Диаграммы потоков данных*» (известна, как DFD), «*Методология структурного анализа и проектирования*» (известна, как SADT), «*Объединение методологических понятий*» (известна, как семейство стандартов IDEF: IDEF0...IDEF14) и др. Часть этих нотаций используется и сейчас (автор не будет углубляться в причины того, почему унаследованные и явно устаревшие нотации продолжают использоваться – кому-то нравится, ну, и Бог с ним).

Третья «волна» методов моделирования, использующая «исполняемые» модели на основе языков моделирования на базе XML (eXtensible Markup Language – расширяемый язык разметки, см. [11]): BPMML (Business Process Modeling Language – язык моделирования бизнес-процессов, см. [12]), BPEL (Business Process Execution Language – язык выполнения бизнес-процессов, см. [13]), XPDL (Process Definition Language – язык описания процессов, см. [14]). Однако эти языки трудно понимаются бизнес-пользователями, что привело к появлению нотации BPMN (Business Process Model and Notation – нотация и модель бизнес-процесса, см. [15]), которая легче понимается и бизнес-пользователями, и специалистами в области ИТ, и однозначно преобразуется в исполняемую модель, которая в современных профессиональных инструментах моделирования может быть запущена в виде некоторой анимации. Во многих случаях это позволяет выявить синтаксические и логические ошибки, которые, возможно, допустил аналитик при разработке модели. Кроме того, разработчики следующих версий известных нотаций таких, как UML (Unified Modeling Language – унифицированный язык моделирования, см. [16]) и SysML (Systems

Modeling Language – язык моделирования систем, см. [17]), ввели в их спецификации возможность формирования исполняемых модулей для анимации моделей.

При рассмотрении подходов к моделированию в «третьей волне» можно заметить, что, перечисленные выше, языки моделирования и нотации тяготеют к стандартизации на уровне международных стандартов и спецификаций, т. е., если посмотреть кто входит в различные международные комитеты и группы по стандартизации нотаций моделирования, а среди их членов, как правило, присутствуют «киты» в области ИТ: Microsoft, Oracle, SAP SE, ...; то несложно сделать вывод – проблема выработки единых подходов к моделированию уже не только назрела, но и активно решается.

Но не многие рядовые специалисты задаются вопросом: а откуда берутся нотации для моделирования бизнес-процессы, что в основе? Помогают в работе – и отлично.

На самом деле, в основе методов моделирования лежат ориентированные нагруженные графы (см., например, [20]), различаясь приписываемой вершинам и дугам информацией и ее интерпретацией. Математические структуры в моделировании бизнес-процессов можно считать устоявшимися (см. Определение 1).

**Определение 1.** *Псевдографом (или просто графом)  $G$  называется пара множеств  $(S, U)$ , где элементы множества  $S$  называются вершинами или узлами графа, а элементы множества  $U$  – ребрами или дугами, и имеются два отображения **begin, end**:  $U \rightarrow S$ . Нагруженным графом называется граф  $G$ , для которого задано отображение **Inf**:  $S \cup U \rightarrow I$ , где  $I$  – некоторое множество объектов, приписываемых узлам и дугам графа.  $G$  называется носителем. См. [23].*

Узлу приписывается действие по преобразованию информации, а дуге – информация, выработанная началом дуги и передаваемая в ее конец, и условие, при котором эта передача информации может быть выполнена.

В любом бизнес-процессе существуют начальные (условие, при котором может начаться процесс, и некоторые начальные

условия) и конечные (целевые) узлы. Начальному узлу приписывается условие, при котором он «активируется». С этим начальным узлом связаны допустимые для него пути обхода по дугам графа, например, как показано на рисунке ниже (см. Рисунок 11).

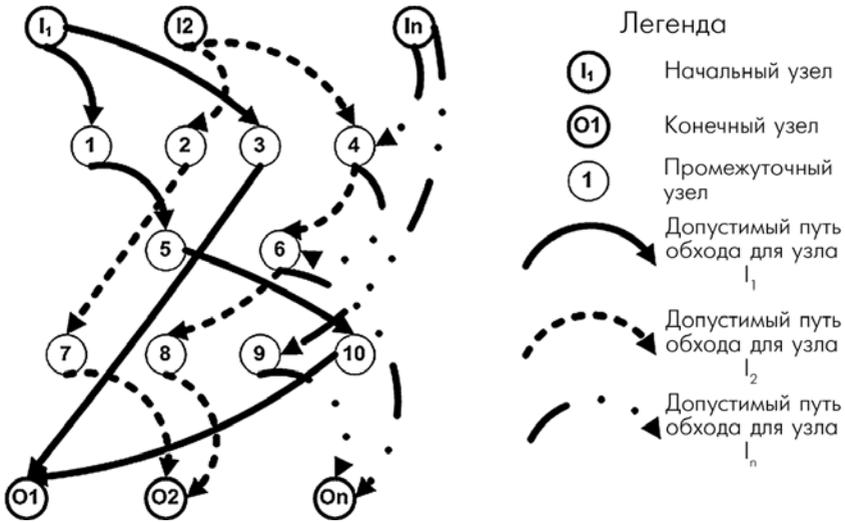


Рисунок 11 – Пример графа с несколькими начальными и конечными узлами (см.[23])

В соответствии с графом, представленным на рисунке, пути обхода можно представить следующим образом (см. ниже выражения 1 ... 3).

Для узла  $I_1$ :

$$\begin{aligned}
 I_1 &\rightarrow 1 \rightarrow 5 \rightarrow 10 \rightarrow O_1 \\
 I_1 &\rightarrow 3 \rightarrow O_1
 \end{aligned}
 \tag{1}$$

Для узла  $I_2$ :

$$\begin{aligned}
 I_2 &\rightarrow 2 \rightarrow 7 \rightarrow O_2 \\
 I_2 &\rightarrow 4 \rightarrow 6 \rightarrow 8 \rightarrow O_2
 \end{aligned}
 \tag{2}$$

Для узла  $I_3$ :

$$\begin{aligned} I_3 &\rightarrow 4 \rightarrow 6 \rightarrow O_3 \\ I_3 &\rightarrow 9 \rightarrow O_3 \end{aligned} \quad (3)$$

Однако на смену «третьей волне» подошла «четвертая волна», которая содержит в качестве узла графа рациональные агенты (см. Определение 2).

**Определение 2.** Рациональный агент – вероятностный функционал  $\vec{\varphi}(f)$ , принимающий в качестве входа вектор входных функций и выдающий вектор выходных воздействий. См. [8].

Функционалы высших уровней значительно выше по силе методов обработки, чем простые функции. Поэтому даже автоматически реализованный функционал позволяет моделировать интеллектуальное реагирование. В случае бизнес-анализа можно представить модель, в которой рациональные агенты (далее – РА) получают на вход последовательности документов и/или рекомендаций по их обработке (другими словами – бизнес-правила). На выходе имеем документ(ы) и/или рекомендации (бизнес-правила).

В соответствии с классификацией работы [8], рассматриваемые в настоящей работе функционалы одновременно являются простыми рефлексными агентами, если рассматривать одно назначение (документы или бизнес-правила на входе и документы или бизнес-правила на выходе), и рефлексивными агентами с памятью, если рассматривать другое назначение (на входе документы и бизнес-правила, и, документы и/или бизнес-правила на выходе).

В соответствии с изложенным выше, можно представить, например, обращение пациента в регистратуру следующим образом.

Поставим в соответствие узлам « $I_1$ », « $1$ », « $3$ », « $5$ », « $10$ », « $O_1$ » (см. Рисунок 11) следующие бизнес-процессы и рациональные агенты (см. Таблица 7).

Таблица 7 – Пример соответствия узлов графа бизнес-процессам

Обозначение узла	Бизнес-процесс	Рациональный агент, выполняющий бизнес-процесс
$I_1$	Обращение пациента в регистратуру	Регистратор
<b>1</b>	Запись пациента к врачу	Регистратор
<b>3</b>	Мотивированный отказ в записи к врачу	Регистратор
<b>5</b>	Прием пациента у врача	Врач
<b>10</b>	Запись в медицинской карте	Врач
$O_1$	Формирование документа о посещении	Регистратор

Тогда в терминах РА и с учетом выражения (1), а также данными, рисунка и таблицы выше (см. Рисунок 11 и Таблица 7), пример бизнес-процесса «Обращение пациента в регистратуру» будет выглядеть следующим образом:

1. РА «Регистратор», при обращении пациента в регистратуру, выполняет бизнес-правило «Принять документы пациента и принять решение о дальнейших действиях». В соответствии с результатом выполнения бизнес-правил, РА «Регистратор» принимает решение о том, как будет дальше (по какой дуге) развиваться процесс: записать пациента к врачу (см. п. 3) или мотивировано отказать в записи на прием (см. п. 2);

2. РА «Регистратор» выполняет бизнес-процесс «Мотивированный отказ в записи к врачу» и переходит к конечному узлу (см. 6);

3. РА «Регистратор» выполняет бизнес-процесс «Запись пациента к врачу», что соответствует переходу к бизнес-процессам РА «Врач»;

4. РА «Врач» выполняет бизнес-процесс «Прием пациента у врача» и переходит к бизнес-процессу «Запись в медицинской карте»;

5. РА «Врач» выполняет бизнес-процесс «Запись в медицинской карте» и передает управление процессу «Формирование документа о посещении»;

6. РА «Регистратор» выполняет бизнес-процесс «Формирование документа о посещении». Т. е. выполняется допустимый проход по графу между узлами **I<sub>1</sub>** и **O<sub>1</sub>** в соответствии с рисунком выше (см. Рисунок 11).

Далее в настоящей работе будет рассматриваться материал, который объединяет методы уходящей «Третьей волны» и методы, только идущей, «Четвертой волны».

## ГЛАВА 4

# НОТАЦИЯ МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ (BPMN)

- Введение в нотацию BPMN
- «Правильные слова» в BPMN
- Базовые элементы BPMN
- Исполняемый частный (внутренний) бизнес-процесс. Пример
  - Неисполняемый частный (внутренний) бизнес-процесс. Пример
    - Публичный бизнес-процесс. Пример
    - Хореография. Пример
    - Подмодель «Соглашения». Пример
  - Методика моделирования в нотации BPMN
  - Выводы

### **А. Введение в нотацию BPMN**

Как упоминалось выше (см. ГЛАВА 3), одним из «китов», на которых держится бизнес-анализ, можно считать нотацию для моделирования бизнес-процессов BPMN. Почему?

Во-первых, эта нотация соответствует требованиям «третьей волны», т. е. это исполняемая нотация на базе XML, которая опирается на международную спецификацию (см. [15]), закрепленную в международном стандарте (см. [21]).

Во-вторых, как будет показано ниже, BPMN позволяет проводить бизнес-анализ в соответствии с парадигмой «четвертой волны», т. е. модели, разработанные в соответствии с этой нотацией, фактически являются отражением реальных процессов в виде РА.

В-третьих, поскольку остальные ключевые нотации построены также на базе XML, легко автоматизируются процессы по конвертации представления в виде BPMN-моделей в представление в других нотациях (в допустимых нотациями пределах).

Например, BPMN-модель типа «Оркестровка» может быть преобразована в UML-модель типа «Вариант использования» (Use Case Diagram – в международных терминах спецификации UML).

Следует отметить один факт, связанный с распространением нотации BPMN в российской практике бизнес-анализа. Сама нотация BPMN и ее стандарт достаточно объемные и сложные документы, требующие достаточно хорошее владение английским языком. Очевидно трудности перевода привели к тому, что отечественный читатель получил сильно усеченную версию оригинальных документов, т. е. вместо триединых типов моделей и их мыслимых комбинаций, которые позволяют полностью формализовать видение бизнес-заказчика, в нашей переводной литературе можно увидеть только краткое (и достаточно неточное) описание только одного типа моделей («Оркестровка»). При этом утверждается, что это и есть BPMN. Но оригинальная нотация BPMN предполагает, что есть еще и другие типы моделей, которые только совместно могут дать исчерпывающее представление по моделируемым бизнес-процессам. Отсюда у многих возникло ложное представление – BPMN очередная игрушка, которая хороша для подготовки иллюстраций в документах, но для практики не имеет применения.

Последуем мудрым рекомендациям Рене Декарта: «Употребляйте правильно слова, и вы избавите мир от половины недоразумений». В нашем случае «правильными словами» будут слова из основополагающего документа для BPMN, а именно спецификации «Business Process Model and Notation (BPMN)», разработанной в международной организации Object Management Group (OMG).

Автор не будет интерпретировать нотацию, т. к. там есть много материалов, адресованных разработчикам CASE-средств, поддерживающих спецификацию BPMN, и, обеспечивающих совместимость между продуктами различных вендоров. Суть и только суть самой нотации<sup>1</sup>.

---

<sup>1</sup> Один мой хороший знакомый считал, что практическим применением BPMN (точнее – того, что у нас называют BPMN) является демонстрация «солидности» компа-

## В. «Правильные слова» в BPMN

BPMN по сути представляет собой некоторую модель, которая позволяет полностью описать некоторый бизнес-процесс. Но эта модель, на самом деле, содержит три основных подмодели:

1. Процессы (или «оркестровки»), которые включают:
  - частные (внутренние) неисполняемые бизнес-процессы;
  - частные (внутренние) исполняемые бизнес-процессы;
  - публичные бизнес-процессы;
2. Хореографии;
3. Взаимодействия, которые могут включать в себя процессы и/или хореографии, включая, как отдельную подмодель, представление типа «Соглашения».

**Определение 3.** *Частные (внутренние) бизнес-процессы* – это процессы, характерные для конкретной организации. Существует два типа частных (внутренних) бизнес-процессов: *исполняемые и неисполняемые*.

*Исполняемый частный (внутренний) бизнес-процесс* – это процесс, модель которого содержит достаточно деталей для интерпретации этого процесса в виде некоторого исполняемого модуля (анимация модели).

*Неисполняемый частный (внутренний) бизнес-процесс* – это процесс, модель которого не содержит достаточное количество деталей для формирования исполняемого модуля (анимации модели).

**Определение 4.** *Публичный бизнес-процесс* – это взаимодействие между частным бизнес-процессом и другим процессом или участником. Включены только те действия, которые используются для взаимодействия с другим процессом или участником в границах публичного процесса. Все остальные

---

нии. Конечно, если, незнающему человеку показать, например, только нос Джоконды, то реакция предсказуема – и это считают шедевром?

«внутренние» действия частного бизнес-процесса не отображаются в публичном процессе. Таким образом, публичный процесс демонстрирует потоки сообщений и порядок этих потоков, которые необходимы для взаимодействия между процессами или участниками.

**Определение 5.** *Хореография* – это отдельная подмодель BPMN, которая не содержит пулов (см. ниже) и оркестровок, и, которая определяет ожидаемое поведение на основе процедурного контракта между взаимодействующими участниками.

Хореография похожа на частный бизнес-процесс, т. к. состоит из сети действий, событий и шлюзов. Однако хореография отличается тем, что действия – это взаимодействия, которые представляют собой набор (1 или более) обменов сообщениями, в котором участвуют два или более участников. Кроме того, в отличие от обычного процесса, в нем нет центрального контролера, ответственного лица или наблюдателя за процессом.

**Определение 6.** *Взаимодействие* – это отображение взаимодействия между двумя или более бизнес-объектами. Взаимодействие обычно содержит два или более пула, представляющих участников взаимодействия. Обмен сообщениями между участниками показывают потоком сообщений, который соединяет два пула (или объекты внутри пулов). Сообщения, связанные с потоками сообщений, также могут быть отображены. Взаимодействие может быть показано, как два или более публичных процесса, взаимодействующих друг с другом. В публичном процессе задачи для участников сотрудничества можно рассматривать как «точки соприкосновения» между участниками. Соответствующие внутренние (исполняемые) процессы, вероятно, будут иметь гораздо больше активности и детализации, чем показано в открытых процессах. Или пул МОЖЕТ быть пустым, «черным ящиком». Хореографии МОГУТ отображаться «между» пулами, поскольку они делают потоки сообщений между пулами. В моделях типа «Взаимодействие» разрешены все комбинации пулов, процессов и хореографии.

**Определение 7.** *Соглашения* – это модель логических связей при обмене сообщениями. Обмен сообщениями связан друг с другом и отражает различные бизнес-сценарии. Модель типа «Соглашения» показывает соглашения (диалоги) в виде шестиугольников между участниками (пулами). Это дает представление «с высоты птичьего полета» о различных соглашениях, которые относятся к предметной области.

Ниже представлены примеры, поясняющие Определения 3...7, по ходу которых вводятся элементы, составляющие основу нотации BPMN.

### **С. Базовые элементы BPMN**

Существует пять базовых категорий элементов BPMN:

1. Поток объектные;
2. Данные;
3. Соединяющие объектные;
4. Полосы выделения (в оригинале «Swimlanes» или в дословном переводе «Плавательные дорожки»);
5. Артефакты.

*Потоковые объектные* являются основными графическими элементами, определяющими поведение бизнес-объектов, и включают три основных объектных:

- События;
- Деятельность;
- Шлюзы.

Категория «*Данные*» включает следующие объектные:

- Объекты данных;
- Входные данные;
- Выходные данные;
- Хранилища данных.

*Соединяющие объектные* предназначены для связи между потоковыми объектными и содержат четыре типа объектных:

- Последовательности;
- Потоки сообщений;

- Объединения;
- Объединения данных.

*Полосы выделения* предназначены для группировки элементов BPMN и включают два объекта:

- Пулы (или «бассейны»);
- Дорожки.

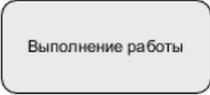
*Артефакты* предназначены для ввода дополнительной информации по моделируемым процессам:

- Группы;
- Текстовые аннотации.

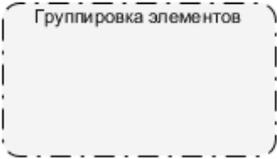
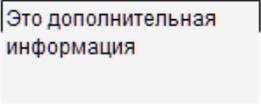
В таблице ниже (см. Таблица 8) показано графическое представление базовых категорий элементов BPMN.

Таблица 8 – Базовые элементы моделей BPMN

Элемент	Описание	Графическое представление элемента
<b>Потоковые объекты</b>		
Событие	Это то, что «происходит» в ходе процесса или хореографии. События влияют на выбор того или иного потока в модели и обычно имеют причину (триггер) или влияние (результат). Существует три типа событий, в зависимости от того, когда они влияют на поток: начало, промежуточное состояние и конец. Элемент «Событие» используются, как в стандартных процессах, так и в хореографии	 Начальное событие
		 Промежуточное состояние
		 Завершение

Элемент	Описание	Графическое представление элемента
Деятельность	Это та работа, которую компания выполняет в процессе. Деятельность может быть атомарной или неатомарной (составной). Элемент «Деятельность» используются, как в стандартных процессах, так и в хореографии	
Шлюз	Управляет расхождением и сходимостью последовательностей, как в стандартных процессах, так и в хореографии. Таким образом, он определяет разветвление, слияние и объединение путей	
Данные		
Объект данных	Объект данных предоставляет информацию о том, что должен выполнить объект «Деятельность» и/или, что он производят. Объект данных может представлять отдельный объект или набор объектов. Для данного объекта не важно, что производится: ввод или вывод данных	
Входные данные	Использование объекта «Входные данные» предполагает, что данные могут только поступать в объект	
Выходные данные	Использование объекта «Выходные данные» предполагает, что данные могут только поступать из объекта	

Элемент	Описание	Графическое представление элемента
Хранилище данных	Использование объекта «Хранилище данных» предполагает, что данные могут, как поступать, так и извлекаться из объекта, а сам объект является некоторым электронным хранилищем данных	
Соединяющие объекты		
Последовательность	Объект «Последовательность» используется, чтобы показать порядок выполнения процессов. Используется, как для оркестровок, так и в хореографии	
Поток сообщений	Объект «Поток сообщений» используется для отображения потока сообщений между двумя участниками, которые готовы отправлять и получать их	
Объединение	Текстовые аннотации и другие артефакты могут быть связаны с графическими элементами.	
Объединение данных	Ассоциация используется для связи информации и артефактов с графическими элементами BPMN. Стрелка на ассоциации указывает направление потока (например, данные), когда это уместно	

Элемент	Описание	Графическое представление элемента
<b>Полосы выделения</b>		
Пул (или «бассейн»)	Пул – это графическое представление взаимодействующего участника. Он может выступать либо в качестве одиночной «дорожки», либо в качестве графического контейнера для разделения набора операций от других пулов. Пул <b>МОЖЕТ</b> иметь внутренние детали в виде элементов «Оркестровки». Или пул <b>МОЖЕТ</b> не иметь внутренних деталей, то есть это может быть «черный ящик»	
Дорожка	Дорожка является подразделом в процессе, иногда в пуле, и будет расширять процесс, либо вертикально, либо горизонтально. Дорожки используются для организации и классификации деятельности.	
<b>Артефакты</b>		
Группа	Объединяет графические элементы, которые находятся в одной категории. Этот тип группировки не влияет на потоки последовательности внутри группы. Название категории отображается на диаграмме как метка группы	
Текстовая аннотация	Текстовые аннотации являются механизмом для бизнес-аналитика, которые позволяют ввести дополнительную текстовую информацию для читателя диаграммы BPMN	

После того, как введены основные «слова» нотации, попробуем «говорить» в формализме BPMN. Для начала проиллюстрируем определения, введенные выше (см. Определение 3 ... 7).

Полный список элементов BPMN приводится ниже (см. Приложение 1. Элементы нотации BPMN).

#### **Д. Исполняемый частный (внутренний) бизнес-процесс. Пример**

Рассмотрим в качестве примера исполняемого частного (внутреннего) бизнес-процесса некоторый гипотетический процесс, который состоит в следующем (см. Рисунок 12):

1. Началом процесса является поступление документа;
2. Выполняется проверка документа;
3. Если требуется правка, то выполняется переход к подпроцессу 4, если нет – к подпроцессу 5;
4. Выполняется правка документа;
5. Документ утверждается;
6. Документ публикуется;
7. Процесс завершается.



Рисунок 12 – Пример исполняемого частного (внутреннего) бизнес-процесса

Обратите внимание на то, как отображается п. 3 – используется базовый элемент «шлюз» для организации ветвления процесса, т. е., как элемент, известный в программировании, «ИЛИ».

Почему в данном случае можно говорить, что бизнес-процесс частный (внутренний) и исполняемый?

Во-первых, как следует из модели, процесс выполняется, скорее всего, одним субъектом внутри некоторой компании.

Во-вторых, видно, что присутствуют последовательность компонентов модели, которые вполне можно анимировать, но особенности исполнения, например, средствами языка VPEL, выходят за рамки настоящей книги.

### **Е. Неисполняемый частный (внутренний) бизнес-процесс. Пример**

Рассмотрим в качестве примера неисполняемого частного (внутреннего) бизнес-процесса то, что показано в разделе ГЛАВА 4.d, но с тем отличием, что покажем КТО и КАКИЕ процессы выполняет без последовательности выполнения (см. Рисунок 13).

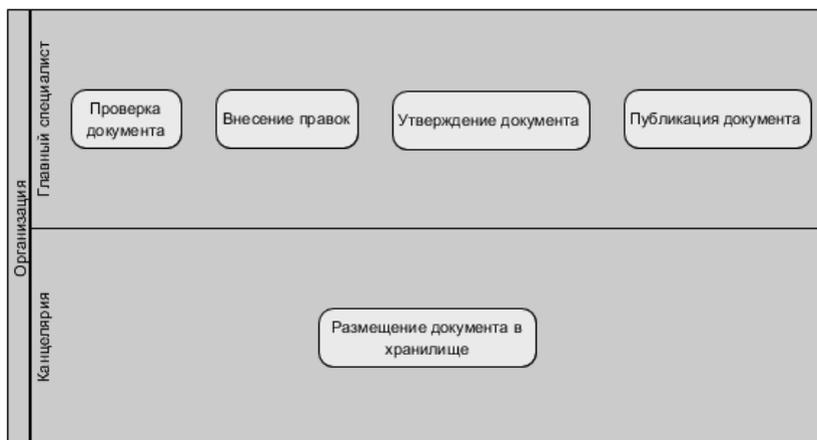


Рисунок 13 – Пример неисполняемого частного (внутреннего) бизнес-процесса

Такие модели часто делают в документах, в которых не важна последовательность выполнения, но важны моменты: ГДЕ делают, КТО делает, ЧТО делает (фактически, на начальных этапах проектирования при заполнении ячеек диаграммы Захмана (см. Таблица 2).

Отметим, что в данной модели использованы такие базовые элементы, как «пул» (см. «Организация» или ответ на вопрос «ГДЕ»), «дорожка» (см. «Главный специалист» и «Канцелярия» или ответ на вопрос «КТО»), «деятельность» (см. «Проверка документа», «Внесение правок», «Утверждение документа», «Публикация документа», «Размещение документа в хранилище» или ответ на вопрос «ЧТО»), но поскольку нет элементов типа «последовательность» интерпретатор языка BPЕL не сможет анимировать процессы. Собственно, поэтому данная модель и относится к неисполняемым.

## Ф. Публичный бизнес-процесс. Пример

Для демонстрации примера публичного бизнес-процесса, в сценарий, изложенный в ГЛАВА 4.d, внесем дополнение: с частным (внутренним) бизнес-процессом (см. Рисунок 12) взаимодействует внешняя сущность (т. е. не является сотрудником организации) «Автор документа». При этом, нас не интересуют детали того, КАКУЮ деятельность выполняет внешняя сущность, но интересуют потоки сообщений, через которые «Автор документа» взаимодействует с частными бизнес-процессами (см. Рисунок 14).

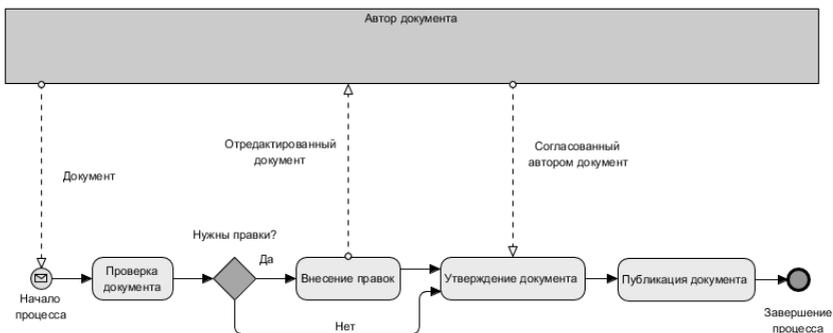


Рисунок 14 – Пример публичного бизнес-процесса

То, что изложено выше в данном разделе, отражено следующим образом: добавлен элемент «пул» в виде «черного ящи-

ка», с которым взаимодействуют элементы типа «деятельность» через потоки сообщений. Отметим, что анимация такой модели невозможна, т. к. существует неопределенность – когда и по какому потоку будут взаимодействовать элементы модели. Фактически здесь отражена концепция взаимодействия между процессом в некоторой организации и внешней сущностью, а для более четкого отражения последовательности выполнения требуется детализации модели.

## Г. Хореография. Пример

Для разработки подмоделей типа «Хореография» в нотации BPMN существует элемент «Задача хореографии», который показан ниже (см. Рисунок 15).



Рисунок 15 – Пример элемента «Задача Хореографии»

В качестве примера подмодели типа «Хореография» приведем процессы, описанные в примере для публичного бизнес-процесса (см. ГЛАВА 4.f). Пример подмодели типа «Хореография» приведен ниже (см. Рисунок 16).

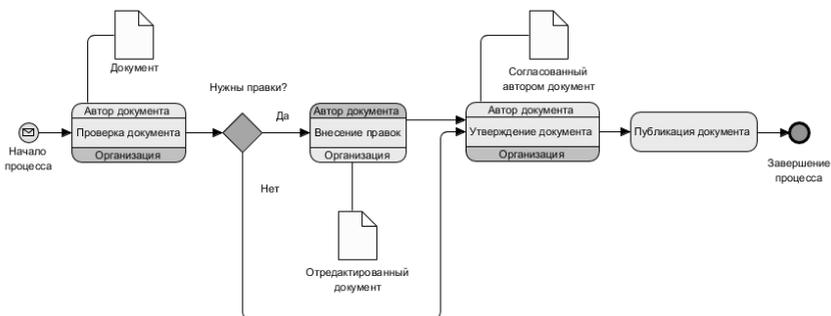


Рисунок 16 – Пример подмодели типа «Хореография»

В данном случае получено компактное представление модели, показанной выше (см. Рисунок 14), но лишенное ее недостатков:

1. Отсутствует неопределенность в последовательности выполнения процессов;
2. Модель может быть исполняемой.

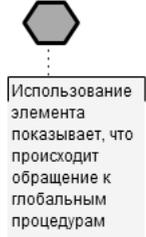
## Н. Подмодель «Соглашения». Пример

Для разработки подмоделей типа «Соглашения» в нотации BPMN вводят дополнительные элементы (см. Таблица 9).

Таблица 9 – Базовые элементы нотации BPMN, используемые в подмоделях типа «Соглашения»

Элемент	Описание	Графическое представление элемента
Событие	Расширение элемента «Событие», рассмотренного в выше (см. Таблица 8), за счет явного указания типа события (триггера – для начальных и промежуточных событий; результата – для конечных событий), а именно –	Начальное событие



Элемент	Описание	Графическое представление элемента
	<ul style="list-style-type: none"> <li>• элемент инкапсулирует сложную структуру, которая может быть показана в моделях нижних уровней;</li> <li>• элемент, который обращается к глобальным структурам (в оригинале – Call Conversation Link)</li> </ul>	<div data-bbox="736 327 900 507" style="border: 1px solid gray; padding: 5px; margin-bottom: 20px;"> <p>Элемент инкапсулирует некоторую сложную структуру</p> </div>  <div data-bbox="754 735 900 1042" style="border: 1px solid gray; padding: 5px;"> <p>Обращение к глобальным процессам</p>  <p>Использование элемента показывает, что происходит обращение к глобальным процедурам</p> </div>

В качестве примера подмодели типа «Соглашения» приведем процессы, описанные в примере для публичного бизнес-процесса (см. ГЛАВА 4.f). Пример подмодели типа «Соглашения» приведен ниже (см. Рисунок 17).

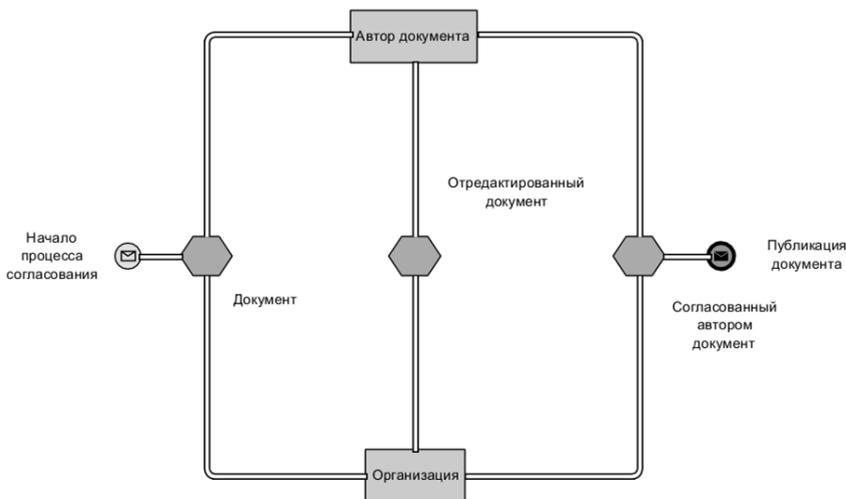


Рисунок 17 – Пример подмодели типа «Соглашения»

Обратите внимание: в модели используются элементы, которые используются и в других подмоделях, а именно: «Событие». В данном случае используются события с триггерами типа «Сообщение» («Начало процесса согласования» и «Публикация документа»), которые обозначают значком «конверт».

## I. Методика моделирования в нотации BPMN

Теперь, когда читатель познакомился с нотацией BPMN, изложим методику моделирования бизнес-процессов, согласно [23] и понимания того, что BPMN, по сути, представляет собой графы, отражающие моделируемые бизнес-процессы (см. ГЛАВА 3).

В качестве основы используем вопросы, которая задает диаграмма Захмана: «ЧТО», «КАК», «ГДЕ», «КТО», «КОГДА», «ПОЧЕМУ».

Методика моделирования бизнес-процессов содержит 3 этапа:

- **Этап 1.** «Предварительный анализ», при выполнении которого определяются участники процессов (в терминах теории графов – узлы). Выполнение Этапа 1 позволяет ответить на

вопросы диаграммы Захмана «КТО» и «ГДЕ». Отображение результатов осуществляется элементами подмодели (подмоделями) типа «Соглашение» с использованием только следующих элементов BPMN: «Участник» (т. е. ответ на вопрос «КТО»), «Группа» (т. е. ответ на вопрос «ГДЕ»), «Текстовая аннотация» (т. е. пояснения к остальным элементам);

- **Этап 2.** «Анализ бизнес-процессов, при выполнении которого определяется что должны делать участники бизнес-процессов (ответ на вопрос «ЧТО») и как они должны это делать (ответ на вопрос «КАК»), когда и почему нужно выполнять бизнес-процесс (ответ на вопросы «КОГДА» и «ПОЧЕМУ», соответственно). Собственно, если опять рассмотреть теорию графов, то:

- о отвечая на вопросы «КОГДА» и «ПОЧЕМУ», мы определяем событие, которое должно произойти, чтобы начать процесс или подпроцесс, т. е. формализуем условия перехода от одного узла графа к другому;

- о отвечая на вопрос «ЧТО», мы определяем, что должно произойти внутри узла для определения дальнейшего выбора обхода графа;

- о отвечая на вопрос «КАК», мы определяем процедуры, которые выполняет участник или участники процесса внутри узла (сами процедуры могут быть указаны, но не детализированы)<sup>2</sup>.

Отображение результатов осуществляется элементами подмодели (подмоделями) типа «Хореография», но допустимо использование элементов из других типов подмоделей: «Соглашения» и «Оркестровка». Все зависит от сложности отображаемых бизнес-процессов и мастерства бизнес-аналитика. *В качестве совета: используйте дополнительные элементы только в крайнем случае, чтобы не перегружать модель избыточной информацией.*

- **Этап 3.** «Синтез бизнес-процессов», при выполнении которого результаты Этапа 2 используются для завершения модели типа «Соглашения» из Этапа 1, т. е. участники объединяются соответствующими связями с указанием того, «ЧТО» они

---

<sup>2</sup> Автор надеется, что читатели, знакомые с объектно-ориентированным программированием или теорией рациональных агентов, уже представили себе соответствующие сущности и их атрибуты.

передают друг другу. При необходимости разрабатываются дополнительные модели, уточняющие и поясняющие предыдущие модели.

*Критерием успешного применения методики* можно считать результат, при котором:

1. Существующее множество моделей необходимо и достаточно для однозначного понимания, отображенных бизнес-процессов, как специалистом функционального заказчика (носителем знаний), так и бизнес-аналитиком;

2. Полученные модели позволяют перейти к моделированию концептуальной модели данных, обрабатываемых в границах бизнес-процессов;

3. Возможно формирование системных (или архитектурных) моделей<sup>3</sup> и соответствующих документов, включающих эти модели, для дальнейшего проектирования в соответствии с ГОСТ 34\* и ГОСТ 19\*.

Однако после успешного применения методики, описанной выше, точнее – ее первой итерации, получаем представление о бизнес-процессах в представлении «Как Есть» (или, как часто любят обозначать в английской терминологии – «AS-IS»). Конечно, имея на руках такой результат, можно уже сделать множество полезных вещей, например, увидеть с «высоты птичьего полета» недостатки и достоинства существующих в организации бизнес-процессов, предложить новую модель, которая учтет выявленные недостатки в структуре организации и ее документообороте. Тоже полезная вещь, которую часто называют «реорганизация бизнес-процессов» (или используют английский аналог «Business Process Reengineering, или просто – BPR»). В большинстве случаев ставится задача, все-таки, разработать и создать информационную систему, т. е. автоматизировать те информационные бизнес-процессы, которые существуют в организации.

Вот тут и появляется необходимость второй итерации методики моделирования в нотации BPMN: работаем в соответствии с методикой, которую уже применили, но добавляем еще одного

---

<sup>3</sup> Работа системного аналитика или, как сейчас стало модно говорить, архитектора выходит за рамки настоящей работы, но автор клятвенно обещает, что планируется издание книги, посвященных этой проблеме.

участника – «Информационную систему». Собственно, это и есть представление бизнес-процессов в представлении «Как Будет» (или, как часто любят обозначать в английской терминологии – «ГО-ВЕ»).

*Описанную методику рассмотрим на следующем примере.*

Существует небольшой магазин, в котором торгуют товарами, получаемыми от сущностей «Поставщик 1» и «Поставщик 2». Руководит магазином Директор, который делегировал финансовые вопросы Бухгалтеру, выявление потребностей в товаре – Товароведу, хранение поступающего товара – Заведующему складом. Рассматриваются некоторые из бизнес-процессов, которые Директор решил автоматизировать:

1. Товаровед выявляет потребность в товаре и направляет Директору заявку на пополнение товарных запасов на складе;

2. Если отсутствуют товары из некоторой условной Группы 1, то Директор обращается к Поставщику 1, если отсутствуют товары из Группы 2, то директор обращается к Поставщику 2;

3. Поставщик 1 и/или Поставщик 2 согласовывают с Директором цены и сроки поставки;

4. После согласования цен и сроков, в адрес магазина направляются документы на поставку, включая договор и счет;

5. Директор визирует документы и передает их Бухгалтеру;

6. После оплаты счета, Бухгалтер передает Директору подтверждение оплаты счета;

7. Директор передает документы на поставку Заведующему складом, а тот ожидает поставки;

8. Поставщик 1 и/или Поставщик 2 доставляют товар на склад и передают Заведующему складом сопроводительные документы;

9. Заведующий складом визирует документы, подтверждающие доставку, и направляет их в адрес соответствующего поставщика;

10. Заведующий складом приходит товар и уведомляет директора и бухгалтера о выполнении условий договора поставщиками;

11. Заведующий складом уведомляет Товароведа о пополнении товарных запасов.

Далее опускаются процессы поиска поставщика или разработчика программного обеспечения, которые могут автоматизировать перечисленные процессы. Перейдем сразу к выполнению методики, описанной выше.

### *Выполнение Этапа 1 (формирование модели «Как есть»)*

Бизнес-аналитик в процессе опроса сотрудников магазина выясняет, что следует рассмотреть следующих участников бизнес-процессов и их процессы:

1. Директор, который выполняет следующие бизнес-процессы:

1.1. Получает заявку на пополнение товарных запасов от товароведа;

1.2. Редактирует при необходимости полученную заявку;

1.3. Направляет заявку поставщикам в соответствии с группами товаров, которые они поставляют;

1.4. Согласовывает с поставщиками цены и сроки поставки;

1.5. Получает от поставщиков коммерческое предложение на поставку;

1.6. При необходимости, согласовывает с поставщиками договор и счет на поставку;

1.7. Получает от поставщиков договорные документы на поставку и визирует их;

1.8. Направляет документы на поставку Бухгалтеру;

1.9. Ожидает подтверждение оплаты от Бухгалтера;

1.10. После получения подтверждения об оплате направляет документы на поставку Заведующему складом;

1.11. Ожидает уведомления от Заведующего складом о поступлении товара;

1.12. Ожидает уведомления от Бухгалтера о закрытии договора;

1.13. Процесс повторяется циклически в случайные моменты времени.

2. Товаровед, который выполняет следующие бизнес-процессы:

2.1. Направляет Директору заявку на пополнение товарных запасов на складе;

- 2.2. Ожидает уведомления о пополнении товарных запасов;
- 2.3. Процесс повторяется циклически в случайные моменты времени;
3. Бухгалтер, который выполняет следующие бизнес-процессы:
  - 3.1. Получает от Директора документы на поставку товара;
  - 3.2. Оплачивает счет и передает Директору подтверждение оплаты поставки товара;
  - 3.3. Ожидает подтверждения о выполнении условий договора на поставку товара от Заведующего складом;
  - 3.4. Закрывает договор на поставку и уведомляет Директора;
  - 3.5. Процесс повторяется циклически в случайные моменты времени.
4. Заведующий складом, который выполняет следующие бизнес-процессы:
  - 4.1. Получает документы на поставку товара от Директора;
  - 4.2. Ожидает поставки товара;
  - 4.3. Получает товар и сопроводительные документы от Поставщика;
  - 4.4. Визирует сопроводительные документы и направляет их Поставщику;
  - 4.5. Приходит товар и уведомляет директора и бухгалтера о выполнении условий договора поставщиками;
  - 4.6. Уведомляет Товароведа о пополнении товарных запасов;
  - 4.7. Процесс повторяется циклически в случайные моменты времени.
5. Поставщик 1 (внешняя сущность по отношению к магазину), который обеспечивает товары из товарной группы 1 и выполняет следующие бизнес-процессы:
  - 5.1. Получает заявку от Директора магазина на поставку товара из Группы 1;
  - 5.2. Согласовывает с Директором магазина цены и сроки поставки;
  - 5.3. Направляет Директору магазина коммерческое предложение на поставку товара;

- 5.4. Согласовывает с Директором магазина коммерческое предложение на поставку товара;
- 5.5. Направляет Директору магазина договор и счет на поставку товара;
- 5.6. Ожидает подтверждения оплаты счета на поставку;
- 5.7. Направляет товар и сопроводительные документы За-ведующему складом в магазин;
- 5.8. Ожидает подтверждения о выполнении условий дого-вора на поставку от Директора магазина;
- 5.9. Процесс повторяется циклически в случайные момен-ты времени.
6. Поставщик 2 (внешняя сущность по отношению к мага-зину), который обеспечивает товары из товарной группы 2 и выполняет следующие бизнес-процессы:
  - 6.1. Получает заявку от Директора магазина на поставку товара из Группы 2;
  - 6.2. Согласовывает с Директором магазина цены и сроки поставки;
  - 6.3. Направляет Директору магазина коммерческое пред-ложение на поставку товара;
  - 6.4. Согласовывает с Директором магазина коммерческое предложение на поставку товара;
  - 6.5. Направляет Директору магазина договор и счет на поставку товара;
  - 6.6. Ожидает подтверждения оплаты счета на поставку;
  - 6.7. Направляет товар и сопроводительные документы За-ведующему складом в магазин;
  - 6.8. Ожидает подтверждения о выполнении условий дого-вора на поставку от Директора магазина;
  - 6.9. Процесс повторяется циклически в случайные момен-ты времени.

В результате, бизнес-аналитик может построить подмодель нотации BPMN типа «Соглашения», как, например, показано ниже (см. Рисунок 18).

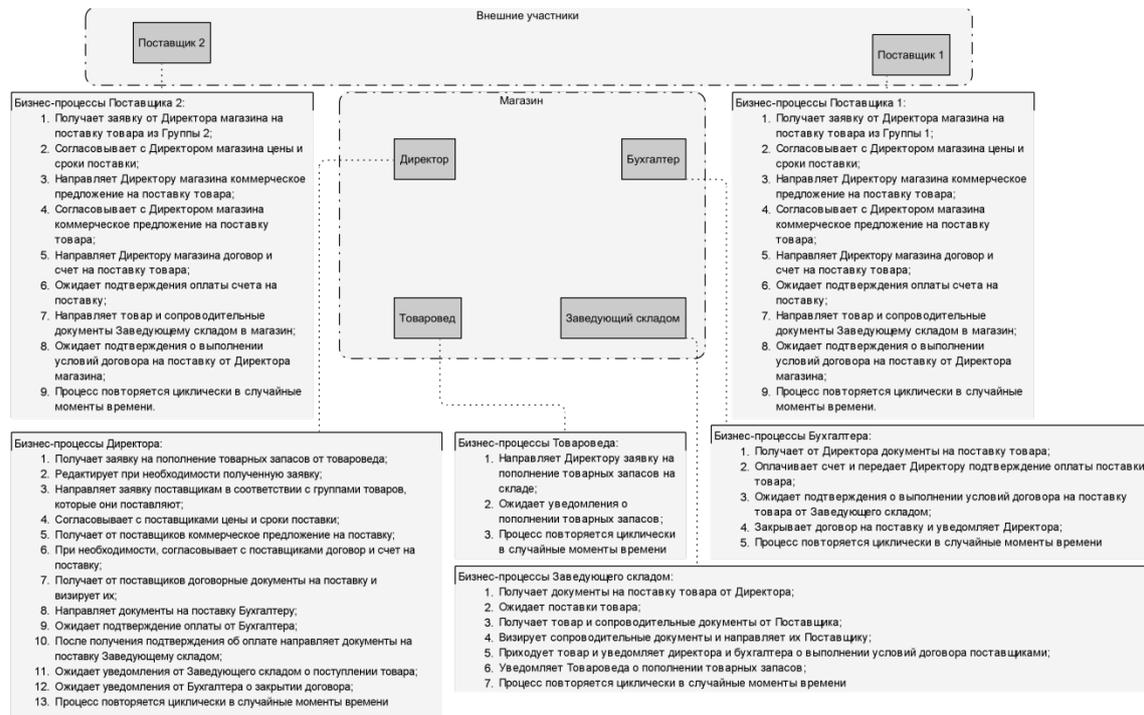


Рисунок 18 – Пример подготовленной модели для этапа «Предварительный анализ»  
Выполнение Этапа 2 (формирование модели «Как есть»)

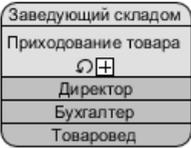
После получения результатов, полученных при выполнении Этапа 1, рассматриваем на какие основные бизнес-задачи разбивается бизнес-процесс, последовательность выполнения бизнес-задач, кто из участников участвует в решении каждой задачи, какая мотивация существует для начала и завершения, как всего бизнес-процесса, так и отдельных задач, что получаем на входе и на выходе, как всего бизнес-процесса, так и отдельных задач, какие документы или иную информацию используют или преобразовывают при выполнении процессов.

Пример возможного отображения результатов моделирования на Этапе 2 показан ниже (см. Рисунок 19 и Рисунок 20).

На рисунках можно заметить, что появился ряд новых элементов, которые мы не вводили ранее (см. Таблица 10).

Таблица 10 – Дополнительные элементы модели

Элемент	Описание	Графическое представление элемента
Сообщение	Элемент «Сообщение» используется для изображения содержимого общения между двумя участниками, как правило, в том случае, если сообщение не отправлено информационной системой, т. е. обычный бумажный документ	
Шлюз типа «ИЛИ»	Управляющий шлюз, который может перенаправить поток по одному направлению при выполнении определенного условия	
Шлюз типа «Распараллеливание»	Расщепляет или объединяет потоки	

Элемент	Описание	Графическое представление элемента
Промежуточное событие «Таймер»	Показывает, что событие может произойти только в определенное время или используется, если нужно показать задержку по времени	 <p>Ожидание поставки товара</p>
Элемент «Задача» в моделях «Хореография» с несколькими участниками	Показывает, что в выполнении некоторой задачи задействованы несколько участников. Кроме того, в данном случае, символ «+» показывает, что данный элемент является инкапсулирующим, т. е. может скрывать внутри себя несколько процессов (задач). Т. е., по сути, символ «+» показывает, что элемент может быть, как бы, «раскрыть» внутреннее содержание, показывающее, что выполняют отдельные участники	

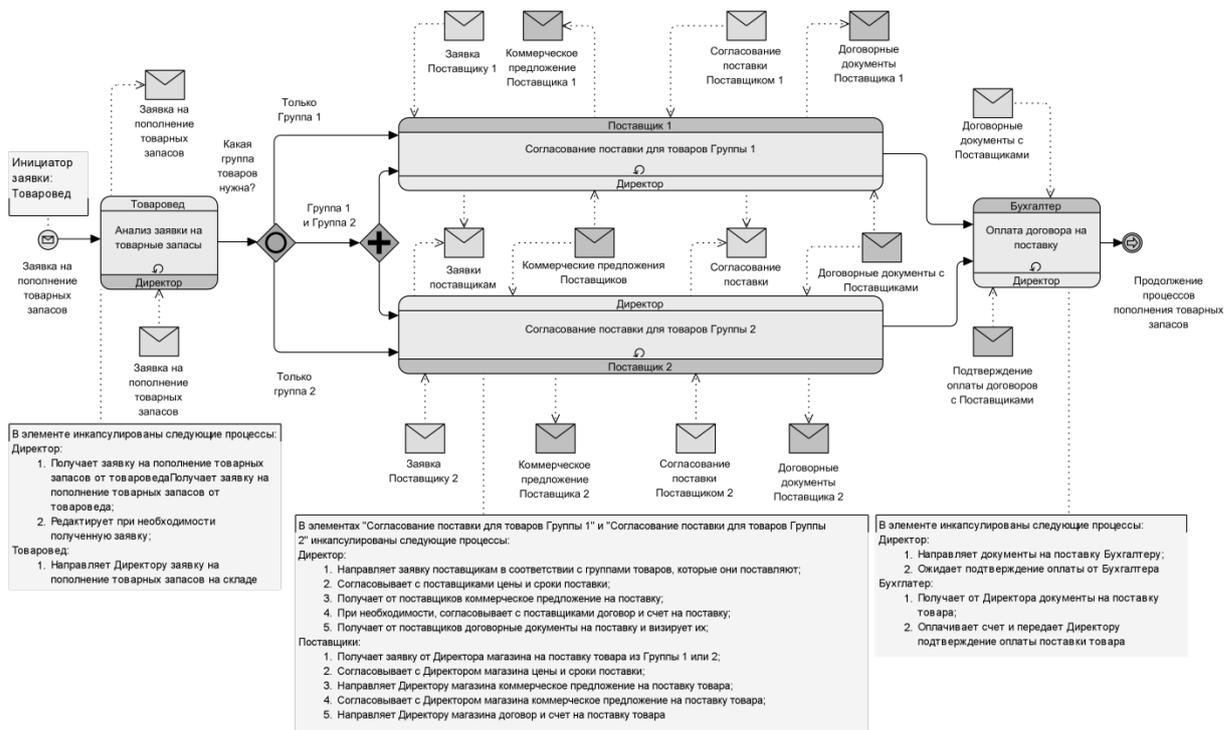


Рисунок 19 – Пример отображения анализа бизнес-процессов на Этапе 2 (Часть 1)



### *Выполнение Этапа 3 (формирование модели «Как есть»)*

Теперь, после того, как было получено понимание того, ЧТО, КАК, КОГДА и ПОЧЕМУ, возвращаемся к модели, которая создавалась на Этапе 1, и можем ее завершить, т. е. расставить информационные потоки и дать комментарии (при необходимости) ПОЧЕМУ возник конкретный поток. Возможный вариант завершенной модели типа «Соглашения» показан ниже (см. Рисунок 21).

Рассмотренный пример организации и возможный вариант описания ее информационных бизнес-процессов во многом показателен по многим причинам:

1. Мы фактически описали один из слоев архитектуры предприятия, а именно – то, что в современной терминологии называют фронт-офис<sup>1</sup> в архитектуре предприятия, т. е.: информационные процессы, участники процессов (включая обмен информацией с внешними участниками), какими именно документами они обмениваются и т. п.;

2. В реальной жизни крайне мало организаций, которые провели четкий анализ своей архитектуры и могут представить в том или ином виде ее модель – в лучшем случае, во время предпроектного обследования, аналитик может получить в отделе кадров документ под названием «Штатная структура», «Структура организации» или что-либо подобное. А посему, коллеги аналитики, готовьтесь: эту работу придется делать вам (см. методику и пример ее применения);

3. Фактически мы получили модель, которую в проектировании информационных систем называют «Как есть». В нашем случае, модель «Как есть» показывает организацию, которую крайне тяжело найти в наше ИТ-время – согласно модели, в организации вообще нет ни одного автоматизированного процесса. Но, все бывает...;

---

<sup>1</sup> Автор искренне не понимает ПОЧЕМУ у нас прижилась эта иностранщина: менеджер – вместо управляющий, дорожная карта – вместо плана действий... Фронт-офис по-русски – основные функции. Вынужден писать, т. к. не могу прямо сейчас повлиять на процессы поглощения одного языка другим. Но – «Коллеги, берегите родной язык!».

4. Модель «Как есть» и ее описание, фактически, это набор данных, которые аналитик должен проанализировать на предмет полноты и эффективности бизнес-процессов в организации, а также, при необходимости, найти неэффективные места. Если квалификация и знание предметной области позволяют, то предложить изменения в архитектуре предприятия или, в принятой ныне терминологии, провести Business Process Reengineering (BPR). Т. е. от полноты модели «Как есть» зависит качество дальнейшей разработки, в частности, разработка модели «Как будет». Или, как говорила моя бабушка: «Как потопаешь – так и полопаешь»...

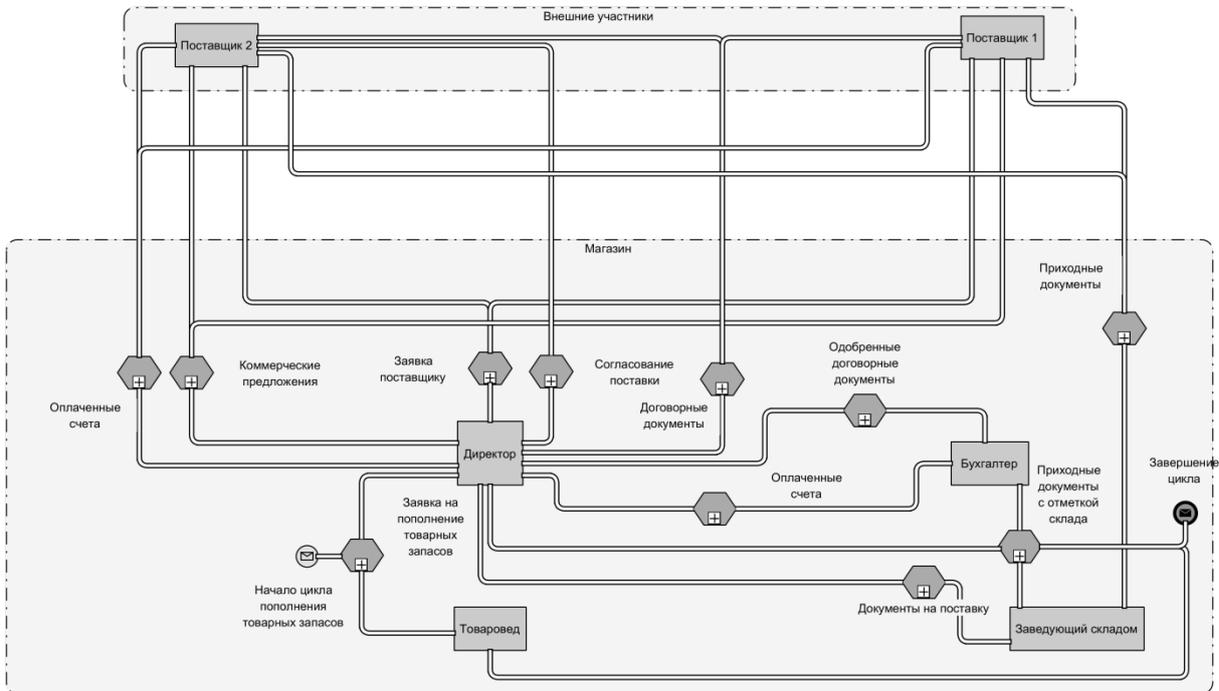


Рисунок 21 – Вариант модели для этапа синтеза. Модель типа «Соглашения» в нотации BPMN

Итак, перейдем к разработке модели или, точнее, группы моделей «Как будет».

*Выполнение Этапа 1 (формирование модели «Как будет»)*

Предположим, что основные функции, выполняемые в организации, остаются неизменными. При этом, рассмотрим из внимательнее и сделаем предположения о том, что из, присутствующего в модели «Как есть», можно автоматизировать (см. Таблица 11).

Таблица 11 – Автоматизируемые функции

Функция в модели «Как есть»	Необходимость автоматизации	Описание автоматизируемой функции
Заявка на пополнение товарных запасов	Да	<b>Участники:</b> Товаровед, Директор <b>Исходные данные:</b> Заявка, сформированная Товароведом <b>Процесс:</b> 1. Товаровед формирует заявку; 2. Товаровед отправляет заявку Директору; 3. Директор согласует заявку с Товароведом (при необходимости редактирует); 4. Директор направляет Товароведу отредактированную заявку с решением по поставке (исключение – отказ без отправки отредактированной заявки) <b>Результат:</b> Решение по заявке, Отредактированная заявка
Отправка заявки на поставку	Да	<b>Участники:</b> Директор, Поставщики <b>Исходные данные:</b> Отредактированная заявка, Список поставщиков <b>Процесс:</b> 1. Директор выбирает из имеющегося списка Поставщиков и, при необходимости, разделяет отредактированную заявку в соответствии с группами товаров, которыми могут обеспечить поставщики; 2. Директор направляет заявки поставщикам <b>Результат:</b> Поставщики получают отредактированную заявку от Директора
Согласование с Поставщиками цен и сроков поставки	Нет	Не применимо

Функция в модели «Как есть»	Необходимость автоматизации	Описание автоматизируемой функции
Получение коммерческого предложения	Да	<p><b>Участники:</b> Директор, Поставщики</p> <p><b>Исходные данные:</b> Коммерческое предложение</p> <p><b>Процесс:</b></p> <ol style="list-style-type: none"> <li>1. Директор получает коммерческое предложение от Поставщиков;</li> <li>2. Директор принимает решение по полученному предложению (если требуется согласование, то процесс может быть циклическим)</li> </ol> <p><b>Результат:</b> Согласованный договор на поставку товара</p>
Оплата договора	Да	<p><b>Участники:</b> Директор, Бухгалтер</p> <p><b>Исходные данные:</b> Согласованный договор на поставку товара</p> <p><b>Процесс:</b></p> <ol style="list-style-type: none"> <li>1. Директор отправляет договор и счет на поставку к Бухгалтеру;</li> <li>2. Бухгалтер отправляет Директору документ об оплате;</li> <li>3. Директор направляет Поставщикам подтверждение оплаты договора на поставку</li> </ol> <p><b>Результат:</b> Оплаченный договор на поставку</p>
Получение товара	+	<p><b>Участники:</b> Директор, Бухгалтер, Заведующий складом, Поставщики, Товаровед</p> <p><b>Исходные данные:</b> Комплект документов на поставку</p> <p><b>Процесс:</b></p> <ol style="list-style-type: none"> <li>1. Поставщики доставляют товар и передают Заведующему складом комплект документов на поставку;</li> <li>2. Заведующий складом принимает товар и визирует документы, подтверждающие выполнение условий поставки, и передает соответствующие документы Поставщикам;</li> <li>3. Заведующий складом передает комплект документов, подтверждающих выполнение условий договора на поставку, Директору и Бухгалтеру;</li> <li>4. Заведующий складом уведомляет Товароведа о пополнении товарных запасов</li> </ol> <p><b>Результат:</b> Комплект документов, подтверждающий выполнение условий договора</p>

Функция в модели «Как есть»	Необходимость автоматизации	Описание автоматизируемой функции
Закрытие договора	+	<p><b>Участники:</b> Директор, Бухгалтер</p> <p><b>Исходные данные:</b> Комплект документов, подтверждающий выполнение условий договора</p> <p><b>Процесс:</b></p> <ol style="list-style-type: none"> <li>1. Директор визирует документы, подтверждающие факт закрытия договора на поставку;</li> <li>2. Бухгалтер оформляет закрытие договора</li> </ol> <p><b>Результат:</b> Закрытый договор</p>

Казалось бы, что ничего не изменилось и при разработке моделей Этапа 1 можно оставить без изменений то, что отображено на рисунке выше для модели «Как есть» (см. Рисунок 18). Но, как показано выше (см. Таблица 11), мы решили автоматизировать часть процессов. Т. е. в модели для Этапа 1 должен появиться некий артефакт – «Автоматизированная система». Возможный вариант такой модели – «Как будет» – показан ниже (см. Рисунок 22).

#### *Выполнение Этапа 2 (формирование модели «Как будет»)*

На этом этапе, фактически, повторим действия, которые выполнялись при формировании моделей «Как есть», но с учетом особенностей, главная из которых – «Информационная система», отмеченных выше (см. Таблица 11 и Рисунок 22). Главный технический прием (или уловку), который мы используем – формируем последовательность из элементов «Задача хореографии», инкапсулирующие задачи нижнего уровня, а наименования задачам присваиваем в соответствии с наименованиями функций, приведенными выше (см. Таблица 11). Отметим, что для элементов «Задача хореографии», включающих подзадачи, количество участников, выполняющих задачу может быть несколько (теоретически – неограниченно). Т. е. при разработке модели на Этапе 2 мы просто вставим участников для каждой из функций (см. Таблица 11).

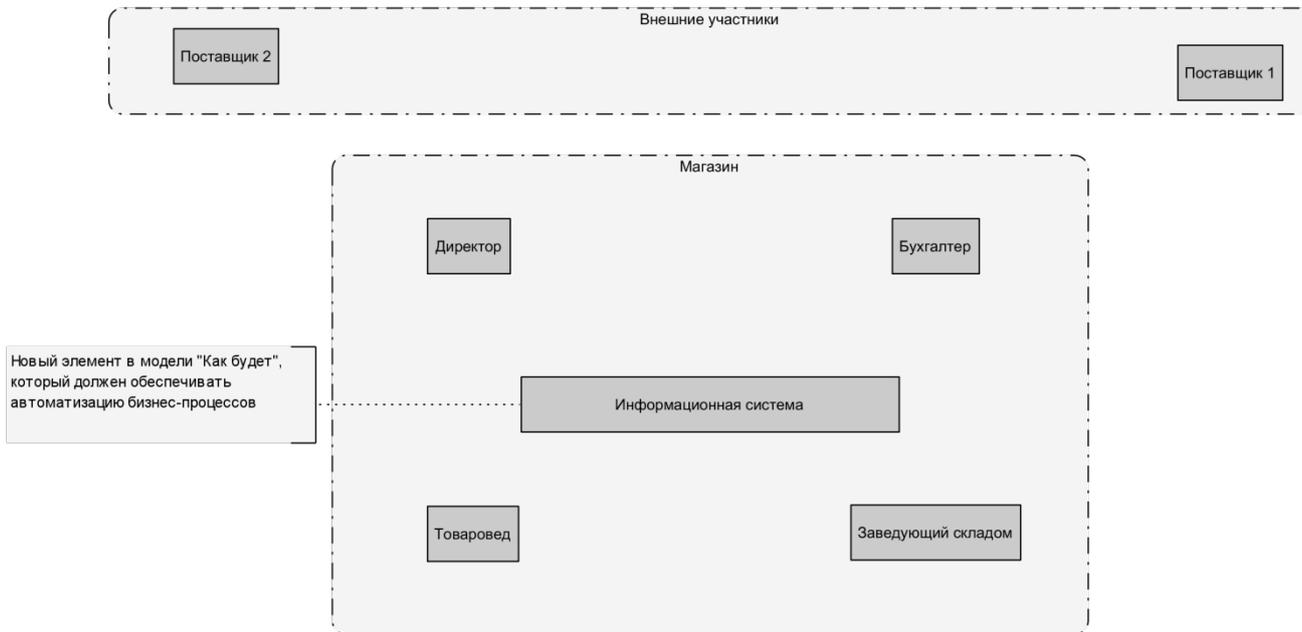


Рисунок 22 – Пример подготовленной модели «Как будет» для этапа «Предварительный анализ»

Нотация BPMN, как правило, не предполагает при описании бизнес-процессов указывать технические подробности реализации. Поэтому, в разрабатываемых моделях, обычно данные представляют путем указания того, что это источник или хранилище данных. При этом, если источник или хранилище данных будут являться (или являются) элементом информационной системы, то в модели его обозначают через элемент «Хранилище данных» (см. Рисунок 23), если источник или хранилище данных являются бумажными, то используют элемент «Объект данных» и его возможные варианты (см. Рисунок 24 – Рисунок 27).



Рисунок 23 – Элемент «Хранилище данных». Поток информации может быть в обе стороны



Рисунок 24 – Элемент «Объект данных». Поток информации может быть в обе стороны



Рисунок 25 – Элемент «Набор данных». Поток информации может быть в обе стороны



Рисунок 26 – Элемент «Ввод данных». Поток информации может быть только к элементу «Задача»



Рисунок 27 – Элемент «Вывод данных». Поток информации может быть только от элемента «Задача»

Возможный вариант модели верхнего уровня для представления «Как будет» показан ниже (см. Рисунок 28).

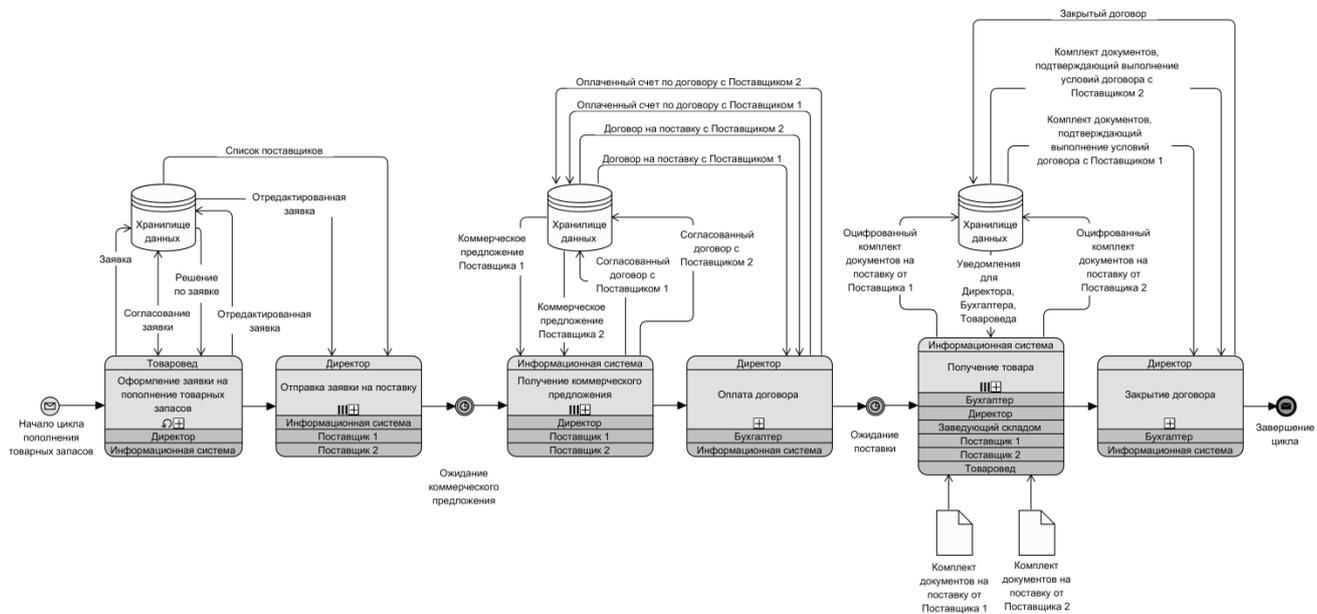


Рисунок 28 – Пример отображения анализа бизнес-процессов на Этапе 2 (для модели «Как будет»)

Однако представление, которое показано выше (см. Рисунок 28) дает только самое общее представление о тех функциях, которые должна выполнять будущая информационная система. Поэтому необходима дальнейшая детализация (не просто так были выбраны элементы типа «Задача хореографии», включающая подпроцессы).

Возможный вариант для детализации процесса «Оформление заявки на пополнение товарных запасов» показан ниже (см. Рисунок 29).

Обратите внимание на следующие особенности модели:

1. Внутри пула «Магазин» расположены две дорожки: «Директор» и «Товаровед», т. е. модель прямо показывает, что есть некоторая сущность «Магазин», в границах которой выполняются процессы двух участников (сущности «Директор» и «Товаровед»). Эти процессы описывают некоторую деятельность, общую для обобщающего участника – «Магазин», который взаимодействует через информационный обмен с еще одним участником – сущность «Информационная система». Обратите внимание на тот факт, что требования нотации BPMN предполагают именование пулов и дорожек, как существительные (например, как в нашей модели или «Компания «Рога и копыта»», «Водитель», «Департамент Здравоохранения» и т. п.), которые, если на основе модели строить текстовое описание, будут выступать в качестве подлежащих. Например, в нашей модели, если взять элементы «Товаровед», «Начало процесса Товароведа», «Обращение к ИС о товарных запасах», то можно построить вполне связное предложение: **«Товаровед идет в Начало процесса Товароведа и выполняет Обращение к ИС о товарных запасах»**. Надеюсь, что теперь понятно, ПОЧЕМУ многие новички в области моделирования бизнес-процессов допускают ошибку, когда пул или дорожку именуют, как процесс: с точки зрения грамматики любого естественного языка нельзя, чтобы ПОДЛЕЖАЩЕЕ в предложении было ГЛАГОЛОМ. А ведь BPMN моделирует не только бизнес-процессы, но и язык, на котором можно строить описания<sup>1</sup>;

---

<sup>1</sup> Некоторые разработчики средств моделирования на основе BPMN делают серьезную ошибку, когда пул или дорожку по умолчанию называют, например, «Process». Это сбивает новичков с толку, и они пишут в наименовании пула что-то типа «Процессы компании». Или разработчик продукта просто не удосужился изучить нотацию.

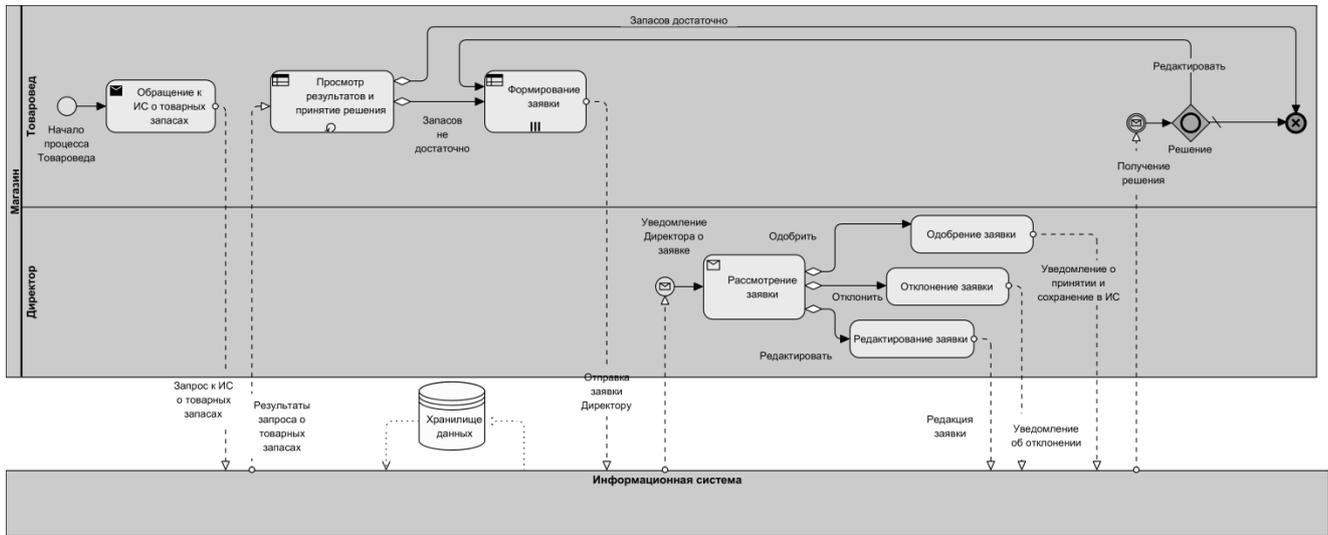


Рисунок 29 – Детализация процесса «Оформление заявки на пополнение товарных запасов». Подмодель типа «Оркестровка»

2. Пул «Информационная система» показан как «черный ящик», т. е. задачей модели, которая показана выше (см. Рисунок 29) является демонстрация тех процессов, которые выполняют участники при взаимодействии с информационной системой и между собой без отвлечения на особенности самой системы: важны только информационные потоки, через которые участники взаимодействуют с системой. Показан неисполняемый элемент «Хранилище данных». Именно по причине того, что он неисполняемый, а использован только для сведения, что информационная система где-то хранит свои данные, этот элемент допустимо расположить в любом месте и показать, что он взаимодействует с системой. Можно было бы вообще не показывать в данной модели, а показать в другой модели, для которой существенной будет демонстрация факта «Информационная система взаимодействует с хранилищем данных»;

3. Элементы модели расположены, как бы по некоторой условной оси времени, что показывает порядок протекания процессов. Вообще-то в нотации BPMN нет такого жесткого требования – «располагать по оси времени», но, как отмечалось выше, BPMN является исполняемым языком, т. е., если ваше средство моделирования позволяет запустить модель на исполнение, то при нарушении порядка следования мы получим ошибку исполнения или результатом исполнения будет нечто странное. Т. е. всегда стремитесь располагать элементы модели в порядке выполнения процессов. Автору известно мало случаев, когда оправдано расположение элементов не так, как надо, а как удобно разработчику модели;

4. Элементу «Начальное событие» (в модели именован, как «Начало процесса Товароведа») не сопоставлен конкретный триггер. Это связано с тем, что участник «Товаровед» может начать выполнять свои процессы в случайные моменты времени и по неформализованным причинам. Следует отметить, что элемент «Начальное событие» является обязательным в подмоделях типа «Оркестровка» и этих элементов должно быть не менее одного;

Логически напрашивается необходимость раскрытия «черного ящика», т. е. необходимость демонстрации тех процессов,

которые выполняются информационной системой при взаимодействии с ней участников, выполняющих общую задачу «Оформление заявки на пополнение товарных запасов». Возможный вариант модели, который показывает, ЧТО и КАК происходит в информационной системе, если к ней обращается Товаровед, показан ниже (см. Рисунок 30).

Модель требует небольших комментариев:

1. ВРМН не предназначена для формирования технических подробностей реализации системы, но позволяет предусмотреть основные модули, из которых будет строиться система. Рисунок 30 показывает, что соответствующая цепочка процессов для запроса Товароведа будет запускаться посредством некоторого стартового события «Обращение к ИС от Товароведа» с триггером типа «Сигнал» (напоминаю, что в пуле элемент «Стартовое событие» – это обязательный элемент). При этом происходит обработка выбора на стороне рабочего места пользователя. Для упрощения отображения, показаны две альтернативы: «Запрос товарных запасов» и «Завершить работу». В реальной информационной системе таких альтернатив у пользователя, конечно же, больше;

2. Если пользователь выбрал альтернативу «Завершить работу», то будет осуществлен переход к конечному событию «Завершить работу с ИС» (напоминаю, что в пуле элемент «Конечное событие» или «Промежуточное событие» – это обязательные элементы);

3. Выбор альтернативы «Запрос товарных запасов» предполагает запрос к элементу «Хранилище данных». Модель показывает «тонкую структуру» до уровня сущностей «Товарные запасы», «Шаблоны экранных форм», «Заявки на товарные запасы». Далее видны два элемента типа «Активность», т. е. «Формирование экранной формы отчета о товарных запасах» и «Отправка Web-формы». Опытный аналитик сразу поймет в чем дело: учитывая, что присутствует сущность «Шаблоны экранных форм» и активность «Отправка Web-формы», можно предположить, что информационная система предполагается к реализации на основе Web-технологий.

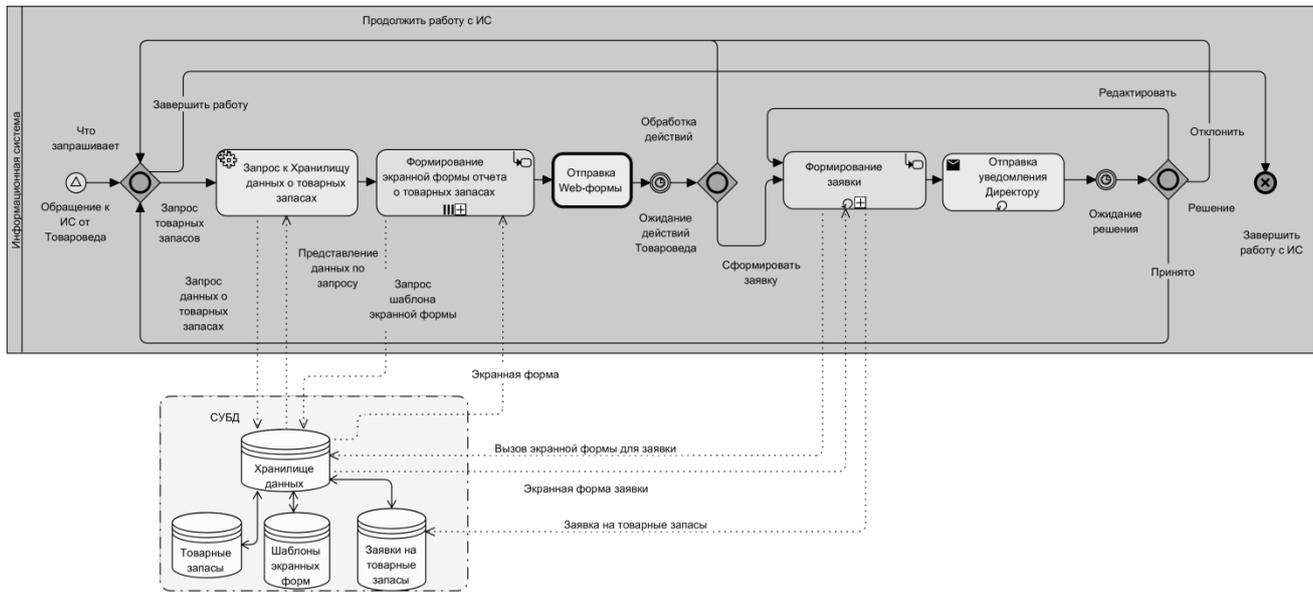


Рисунок 30 – Детализация процессов, происходящих в Информационной системе, при обращении к ней Товароведа. Подмодель типа «Оркестровка»

Идеология формирования остальных наборов моделей, а именно:

1. Детализация процессов, происходящих в Информационной системе, при обращении к ней Директора (для процессов Директора при выполнении процессов «Оформление заявки на пополнение товарных запасов»);

2. Детализация процесса «Отправка заявки на поставку» с учетом процессов всех участников и представлением информационной системы в виде «черного ящика»;

3. Детализация процессов, происходящих в Информационной системе, при обращении к ней Директора (для процессов Директора при выполнении процессов «Отправка заявки на поставку»);

4. Детализация процесса «Получение коммерческого предложения» с учетом процессов всех участников и представлением информационной системы в виде «черного ящика»;

5. Детализация процессов, происходящих в Информационной системе, при обращении к ней Директора (для процессов Директора при выполнении процессов «Получение коммерческого предложения»);

6. Детализация процесса «Оплата договора» с учетом процессов всех участников и представлением информационной системы в виде «черного ящика»;

7. Детализация процессов, происходящих в Информационной системе, при обращении к ней Директора (для процессов Директора при выполнении процессов «Оплата договора»);

8. Детализация процессов, происходящих в Информационной системе, при обращении к ней Бухгалтера (для процессов Директора при выполнении процессов «Оплата договора»);

9. Детализация процесса «Получение товара» с учетом процессов всех участников и представлением информационной системы в виде «черного ящика»;

10. Детализация процессов, происходящих в Информационной системе, при обращении к ней Директора (для процессов Директора при выполнении процессов «Получение товара»);

11. Детализация процессов, происходящих в Информационной системе, при обращении к ней Бухгалтера (для процессов Директора при выполнении процессов «Получение товара»);

12. Детализация процессов, происходящих в Информационной системе, при обращении к ней Заведующего складом (для процессов Директора при выполнении процессов «Получение товара»);

13. Детализация процессов, происходящих в Информационной системе, при обращении к ней Товароведа (для процессов Директора при выполнении процессов «Получение товара»);

14. Детализация процесса «Закрытие договора» с учетом процессов всех участников и представлением информационной системы в виде «черного ящика»;

15. Детализация процессов, происходящих в Информационной системе, при обращении к ней Директора (для процессов Директора при выполнении процессов «Закрытие договора»);

16. Детализация процессов, происходящих в Информационной системе, при обращении к ней Бухгалтера (для процессов Директора при выполнении процессов «Закрытие договора») – будет аналогичной для моделей, показанных выше (см. Рисунок 29 и Рисунок 30).

Возникает вопрос: а где же участники «Поставщик 1» и «Поставщик 2»?

Как правило, взаимодействие с такими участниками осуществляется через обмен сообщениями электронной почты, т. е. их процессы будут заключаться в приеме и отправке электронных сообщений, которые будут формироваться автоматически в границах информационной системы магазина. Обычно такое взаимодействие отображается через элемент BPMN «Вызов» (или в оригинале «Call Activity»), который предполагает, что взаимодействие осуществляется через некоторую внешнюю систему, особенности реализации которой нас не интересуют. Забегая вперед, скажу, что, обычно, мы представляем такую

внешнюю систему, как набор интерфейсов, свойств, перечня событий, на которые внешняя система реагирует, и понимание того, как эти события обрабатываются. Обычно для этого используется почтовый клиент.

Но никто не мешает вам предусмотреть вариант, при котором внешние участники являются пользователями информационной системы.

Еще одно замечание. Некоторые продвинутые читатели могут спросить о процессах авторизации и о защите персональных данных.

К сожалению, объем издания и объемность темы «Информационная безопасность» не позволили автору рассмотреть эту тему и особенности проектирования бизнес-моделей для процессов информационной безопасности. Надеюсь, что удастся в следующих редакциях книги (или будет издана отдельная книга, посвященная этому вопросу) осветить этот очень непростой вопрос. Тем не менее, в упражнениях читателю будет предложено самостоятельно разработать BPMN-модель, которая будет учитывать вопросы информационной безопасности.

## **Ж. Выводы**

1. BPMN – это нотация, которая позволяет построить модель бизнес-процессов, отражающих существующую архитектуру организации, т. е. заполнить в диаграмме Захмана первую строку или, в контексте настоящей книги, построить BPMN-модель «Как есть»;

2. BPMN позволит вам в процессе анализа модели «Как есть» и общения с сотрудниками организации (носителями знаний в предметной области) выявить объект автоматизации (т. е. какие процессы будем автоматизировать). Это соответствует второй строке в диаграмме Захмана или модели «Как будет»;

3. Построение моделей «Как есть» и «Как будет» следует выполнять по методике, которая не позволит вам упустить важные моменты и самоорганизоваться. В книге описана методика, которая позволяет обеспечить некоторую технологическую дисциплину (см. раздел «ГЛАВА 4.1. МЕТОДИКА МОДЕЛИРОВАНИЯ В НОТАЦИИ BPMN»);

4. Следует твердо придерживаться основных постулатов нотации BPMN:

а. Подмодели типа «Оркестровка» и «Хореография» ВСЕГДА должны содержать элементы, которые четко и однозначно показывают начало и завершение процессов, т. е. элементы «Начальное событие» или «Промежуточное событие» (для начала процессов) и «Промежуточное событие» или «Конечное событие (для завершения процессов);

б. Наименование пулов и дорожек в пулах (имена участников процессов) должно быть существительным или именем собственным;

с. Наименование активностей или задач должно быть глагольной формой, которое четко и однозначно отражает основную функцию элемента;

д. Стремитесь изображать подмодели типа «Оркестровка» и «Хореография» так, чтобы элементы, входящие в них, были выставлены в соответствии с последовательностью выполнения и разнесены по времени выполнения (ось времени);

5. Рекомендация, выработанная годами практического опыта: количество значимых элементов на одной модели не должно превышать шести элементов – в противном случае восприятие процессов размывается. Используйте элементы, которые позволяют инкапсулировать процессы нижнего уровня или используйте промежуточные события типа «Связывающее событие», т. е. переход от одной страницы к другой.

## **К. Вопросы и задачи для повторения материала**

1. Используя подмодель типа «Оркестровка» и концепцию «Черного ящика», отобразить информационные процессы при взаимодействии покупателя и продавца в процессе покупки, например, телевизора.

2. Процессы из задачи 1 отобразить с помощью подмодели типа «Хореография».

3. Смоделировать процессы бизнес-аналитика на основе материала, который описан выше (см. ГЛАВА 2), в виде группы моделей «Как есть», используя все типы подмоделей BPMN («Хореография», «Оркестровка», «Соглашения»), а также методику, описанную в этом разделе (см. ГЛАВА 3).

4. Аналогично задаче 3, но предположить, что разрабатывается некоторая ИС, которая будет поддерживать работу бизнес-аналитика, т. е. необходимо построить группу моделей «Как будет».

## ГЛАВА 5

# НОТАЦИЯ МОДЕЛЕЙ СУЩНОСТЬ-СВЯЗЬ (ERD)

- Краткое введение в теорию БД
- Нотация модели «Сущность-связь»
- Типы моделей ERD
- Выводы или «Как разработать ERD»

### А. Краткое введение в теорию БД

Одна из задач, которую приходится решать бизнес-аналитику, состоит в проектировании структуры хранилища данных или базы данных (далее – БД), которая, как правило, строится на основе теории реляционных БД (см., например, [10]).

**Определение реляционной базы данных** – это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об определенном объекте (сущности), и между таблицами существуют связи, организованные посредством ключей (ключевых атрибутов), каждый из которых является уникальным для таблицы. Строка таблицы содержит информацию об одном объекте, а колонка представляет собой атрибут объекта и должна иметь строго определенный тип данных. Таблицы базы данных должны отвечать требованиям нормализации, т. е. ограничениям, которые позволяют устранить дублирование данных и обеспечить их непротиворечивость. Используются, главным образом, три формы нормализации:

- *Первая нормальная форма.* Реляционная таблица приведена к первой нормальной форме тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто;
- *Вторая нормальная форма.* Реляционная таблица задана во второй нормальной форме, если она удовлетворяет требованиям первой нормальной формы и все ее поля, не входящие в

первичный ключ, связаны полной функциональной зависимостью с первичным ключом. Чтобы привести таблицу ко второй нормальной форме, необходимо определить функциональную зависимость полей. Функциональная зависимость полей – это зависимость, при которой в экземпляре информационного объекта определенному значению ключевого реквизита соответствует только одно значение описательного реквизита;

- *Третья нормальная форма.* Таблица находится в третьей нормальной форме, если она удовлетворяет требованиям второй нормальной формы, ни одно из ее неключевых полей не зависит функционально от любого другого неключевого поля.

Созданию БД предшествует сбор информации о тех объектах и связях между ними, которые используются при выполнении бизнес-процессов в организации, и, которые будут использоваться в разрабатываемой ИС (см. ГЛАВА 2). Как правило, модель данных формируется в соответствии с нотацией «Сущность-связь», которая рассматривается ниже.

## **В. Нотация модели «Сущность-связь»**

БД является неотделимой частью любой современной ИС, иными словами – интегрирована в ИС. Для гарантии того, что разрабатываемая ИС, будет иметь высокую производительность, хорошую управляемость и возможность дальнейшего сопровождения, используется диаграмма «Сущность-связь», которая также является средством взаимодействия между различными членами группы разработки.

Что такое диаграмма «Сущность-связь», которую еще называют ERD (от английского Entity Relationship Diagram), ER Diagram или ER-модель?

ERD – это структурированная диаграмма, которую используют в процессе разработки БД, и которая содержит различные символы и соединители для визуализации важной информации: что является основными сущностями в границах системы и какие связи существуют между этими сущностями.

Под сущностями в ERD понимаются одушевленные бизнес-объекты, такие, как например, человек/роль (например, студент), материальные бизнес-объекты (например, продукт), нематериальные бизнес-объекты (например, электронный журнал) и т. д. «Отношения» – это то, как эти объекты связаны друг с другом в системе.

Каковы основные причины для разработки ERD? Дело в том, что, как показала практика, и, как следует из теории, ERD наиболее приспособлена для разработки БД в терминах визуализации концептуальной модели и терминах разработки физической реализации БД, а также в других многочисленных случаях, когда ERD может быть полезна. Ниже приводятся типичные сценарии использования ERD:

- **Разработка БД** – при необходимости внесения изменений в структуру БД достаточно рискованно работать непосредственно в СУБД, т. е. возможно повредить или уничтожить данные в работающей БД, а поэтому так важно тщательно спланировать изменения. ERD – это инструмент, который помогает. Разработка ERD при проектировании БД, дает возможность выявления ошибок и недостатков проектирования новой БД, а также возможность внесения исправлений/изменений в существующую БД (с последующим анализом рисков) перед выполнением изменений в работающей БД;

- **Наладка БД** – отладка БД, как правило, сложная задача, особенно когда БД содержит много таблиц, которые требуют написания сложного SQL-запроса для получения необходимой информации. Визуализируя схему БД с помощью ERD, возможно получить полную картину всей схемы БД. Можно легко находить сущности, просматривать их атрибуты и определять отношения, которые одни сущности имеют с другими. Все это позволяет анализировать существующую БД и легче выявлять проблемы с БД;

- **Создание БД и внесение изменений** – существуют инструменты, поддерживающие ERD, такие, как, например, Visual Paradigm, которые включают инструмент генерации БД, и, который может автоматизировать процесс создания и исправления БД с помощью, созданной ERD. Таким образом,

при наличии такой возможности ERD уже не просто статическая диаграмма, а зеркало, которое действительно отражает физическую структуру БД;

- **Сбор требований** – определение требований к ИС, через разработку концептуальной ERD позволяет изображать бизнес-объекты высокого уровня в ИС. Такая концептуальная модель также может быть преобразована в физическую модель БД, которая поможет, как в создании реляционной БД, так и в создании карты процессов или модели потока данных (известной, как DFD – от английского Data Flow Diagram).

Как уже упоминалось выше, нотация ERD включает ряд элементов, из которых строятся модели (три типа, которые подробнее будут рассмотрены ниже): концептуальная, логическая, физическая. Итак, элементы ERD:

- **Сущность ERD** – это определяемые в системе объект или понятие, такие, как, например, человек или роль (например, специалиста), объект (например, счет), концепция (например, профиль) или событие (например, транзакция) (заметим, что в ERD часто используются равнозначные термины «сущность» и «таблица»). При определении сущностей, следует их определять, как существительные. В ERD сущность отображается в виде прямоугольника со скругленными углами и ее именем сверху, а также атрибутами сущности, перечисленными в теле сущности. Ниже показан пример сущности в ERD (см. Рисунок 31).

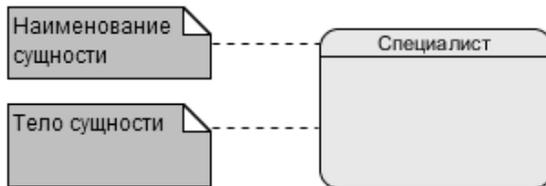


Рисунок 31 – Элемент «сущность» в нотации ERD

- **Атрибуты сущности** – это свойства или характеристики сущности, которая их содержит. Вместо термина «атрибут»

часто используют термин «столбец»<sup>1</sup>. Атрибут имеет имя, которое описывает свойство, и тип, который описывает тип атрибута, например, *varchar* для строки и *int* для целого числа (типы атрибутов могут зависеть от СУБД, на которой будет реализована модель). Когда для разработки физической БД используется ERD, важно обеспечить использование типов, которые поддерживаются целевой СУБД. Ниже показан пример сущности с некоторыми атрибутами (см. Рисунок 32).

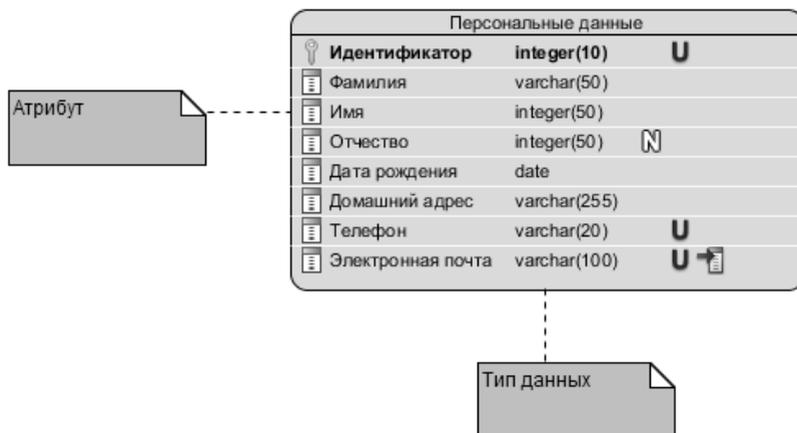


Рисунок 32 – Сущность, включающая атрибуты и типы данных для каждого атрибута

- **Первичный ключ** – это особый вид атрибута сущности (часто используют сокращение РК от английских слов Primary Key), который однозначно определяет запись в физической таблице БД. Другими словами, не должно быть двух (или более) записей, которые имеют одинаковое значение для атрибута РК. В приведенном выше примере ERD (см. Рисунок 32) показана сущность «Персональные данные» с атрибутом для первичного ключа «Идентификатор» и предварительный просмотр записей

<sup>1</sup> Автор предлагает для того, чтобы обеспечить единство терминологии и не вызывать путаницы, с термином «сущность» употреблять в связке термин «атрибут», а если читателю больше нравится не «сущность», а «таблица», то в связке с ней употреблять термин «столбец».

таблицы в БД (см. Таблица 12). Четвертая запись недействительна, потому что значение идентификатора 'PD00002' уже используется другой записью (механизмы обеспечения целостности реальной СУБД просто не допустили бы такую запись; такая вольность возможна только в примере).

Таблица 12 – Пример записей в физической таблице БД для сущности «Персональные данные» (см. Рисунок 32)

Идентификатор	Фамилия
PD00001	Кузнецов
PD00002	Иванов
PD00003	Петров
PD00002	Сидоров

- Внешний ключ** – является ссылкой на первичный ключ в другой таблице. Он используется для идентификации отношений между сущностями. Обратите внимание, что внешние ключи не обязательно должны быть уникальными. Несколько записей могут иметь одинаковые значения. В приведенном ниже примере ERD (см. Рисунок 33), показана сущность с некоторыми столбцами, среди которых внешний ключ используется для ссылки на другую сущность. Предварительный просмотр записей таблицы в БД (см. Рисунок 34).

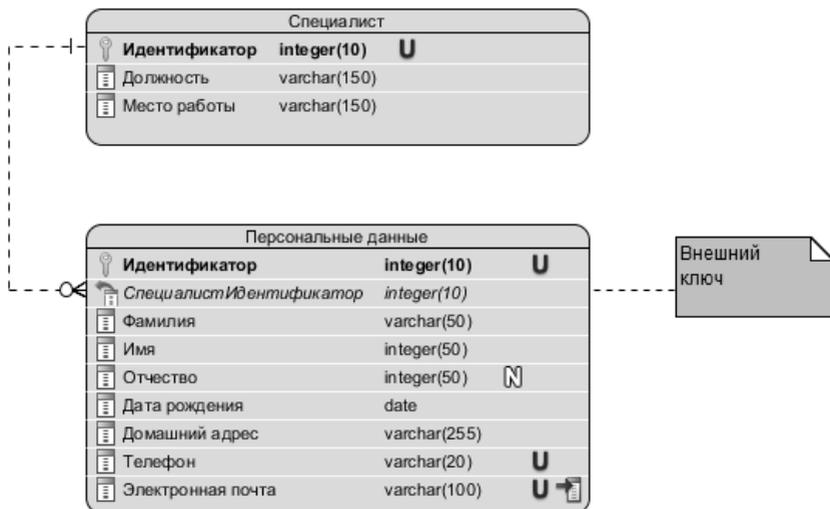


Рисунок 33 – Пример связи сущностей через внешний ключ

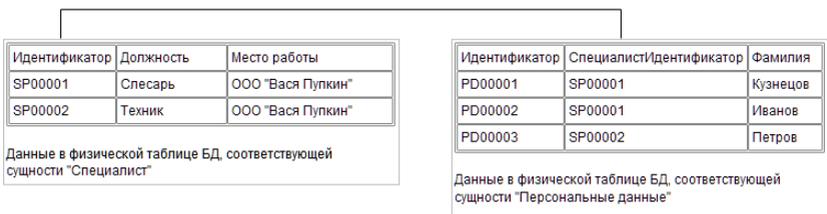


Рисунок 34 – Образец записей в БД для примера, показанного выше (см. Рисунок 33)

В колонке «СпециалистИдентификатор» данные внешнего ключа. Обратите внимание, что присутствуют одинаковые данные, что допустимо и соответствует следующей информации, сохраненной в БД:

- «В ООО «Вася Пупкин» в должности «Слесарь» работают Кузнецов и Иванов»;
- «В ООО «Вася Пупкин» в должности «Техник» работает Петров».

- **Связь** – это отношение между двумя сущностями, которое означает, что две сущности связаны друг с другом определенным образом. Например, если на предприятии может быть несколько слесарей, то с каждой из должностей должен быть связан определенный субъект (см. Рисунок 33 и Рисунок 34). Таким образом, сущность «Специалист» связана с сущностью «Персональные данные», а отношения представлены в виде связующего звена между ними.

- **Кардинальность** – определяет возможное количество вхождений в одну сущность, которая связана с некоторым числом вхождений в другой сущности. Например, в ОДНОЙ должности может работать МНОГО субъектов. Если это представить в ERD, то сущности «Специалист» и «Персональные данные» связаны между собой связью «один-ко-многим» (см. Рисунок 33). На ERD кардинальность представлена в виде «вороньей лапки» на конце связи. Существует три общих кардинальных отношения для связей: «Один-к-Одному», «Один-ко-Многим» и «Многие-ко-Многим».

○ **Связь типа «Один-к-Одному»** – эта связь в основном используется для разделения сущности на две части с тем, чтобы представить информацию кратко и сделать ее более понятной. На рисунке ниже показан пример связи «Один-к-Одному». Пример разделения одной сущности на две показан ниже (см. Рисунок 35).

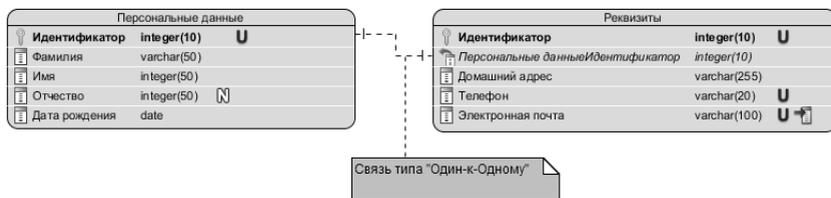


Рисунок 35 – Пример разделения одной сущности на две с помощью связи «Один-к-Одному»

Действительно возможно создать одну сущность «Персональные данные» и разместить в ней все атрибуты, имеющие отношение к персональным данным. Но при необходимости изменений в списке атрибутов или редактирования типа атрибута (например, необходимо изменить ошибочный тип integer(50) атрибута «Имя» на правильный тип varchar(50), см. Рисунок 35) придется проводить обновление не только таблицы, хранящейся в БД, но и всех записей, которые в ней хранятся. При этом, если записей очень много, понадобится достаточно много времени на такое обновление и, кроме того, возможны ошибки в данных при переносе в измененную таблицу. Подход с разделением позволяет снизить время на внесение изменений и минимизировать ошибки в данных, т. к. работа ведется с небольшой таблицей.

○ **Связь типа «Один-ко-Многим»** – это связь между двумя сущностями X и Y, в соответствии с которой одна запись в X может быть связана со многими записями в Y, но запись в Y связана только с одной записью в X. На рисунке выше показан пример связи «Один-ко-Многим» (см. Рисунок 33 и Рисунок 34).

В соответствии с определением связи «Один-ко-Многим», из рисунков видно, что одной записи в сущности «Специалист»

может соответствовать несколько записей в сущности «Персональные данные», но не наоборот. Одной записи в сущности «Персональные данные» может соответствовать только одна запись в сущности «Специалист».

○ **Связь типа «Многие-ко-Многим»** – это связь между двумя сущностями X и Y, в соответствии с которой множество записей в X могут быть связаны со многими записями в Y и наоборот. На рисунке ниже (см. Рисунок 36) показан пример связи «Многие-ко-Многим».

○

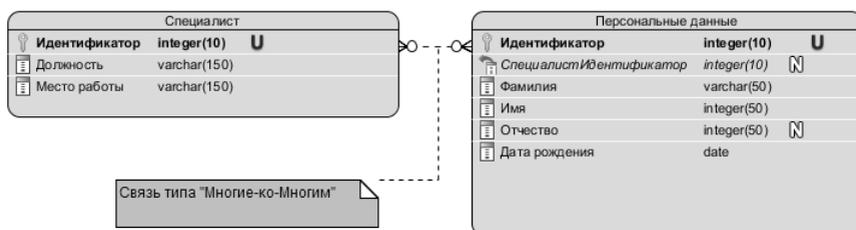


Рисунок 36 – Пример связи «Многие-ко-Многим»

Что изменилось относительно того, что демонстрирует Рисунок 33? Рисунок 36 показывает непростую ситуацию с точки зрения информационных бизнес-процессов. Т. е. то, что на нескольких предприятиях может занимать несколько должностей некоторое количество человек. И наоборот – некоторое количество человек может работать на нескольких предприятиях и занимать некоторое количество должностей. Не просто, но в жизни бывает еще сложнее.

○ **Явное объявление кардинальности**

Следует отметить, что концы связей несут дополнительную смысловую нагрузку (см. Таблица 13).

Таблица 13 – Особенности объявления кардинальности

Представление конца связи	Описание
∞ --	Кардинальность «Ноль или больше», т. е. количество записей, связываемых в сущности, может быть равно нулю (вырожденный случай) или сколь угодно большим (теоретически, но практически зависит от СУБД, на которой реализуется БД, а также от возможностей аппаратных средств, на которой установлена СУБД)
≥+ --	Кардинальность «Один или больше», т. е. количество записей, связываемых в сущности, не может быть меньше единицы или может быть сколь угодно большим (теоретически, но практически зависит от СУБД, на которой реализуется БД, а также от возможностей аппаратных средств, на которой установлена СУБД)
+0 --	Кардинальность «Ноль или один», т. е. количество записей, связываемых в сущности, может быть равно нулю (вырожденный случай) или единице
-  --	Кардинальность «Исключительно один», т. е. количество записей, связываемых в сущности, может быть только равно единице

- **Представление данных (View)** – сущность, которая может быть определена в ERD, а также в физической БД, и, которая формируется, как специфический запрос к одной или нескольким сущностям, входящим в ERD (или реальную БД). Т. е. в одной сущности объединяются атрибуты (в общем случае) нескольких сущностей так, как если бы это была одна сущность (см. Рисунок 37).

Представление данных обозначается в ERD аналогично сущностям, но не соединяется с другими элементами связями. Практическая реализация элемента «Представление данных» будет показана ниже. Пример формирования представления данных (сущность «Список специалистов») представлен ниже (см. Рисунок 38).

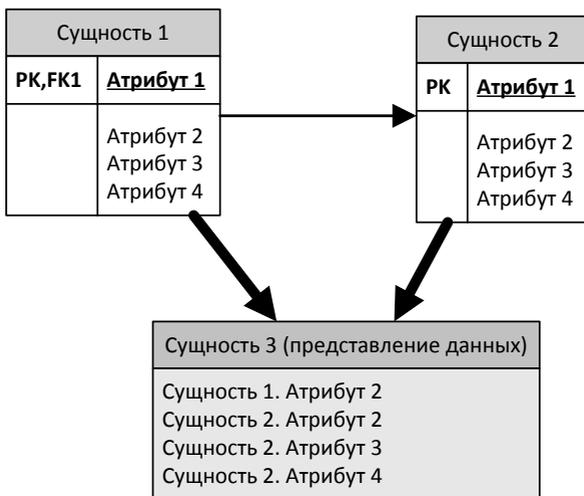


Рисунок 37 – Модель формирования представления данных

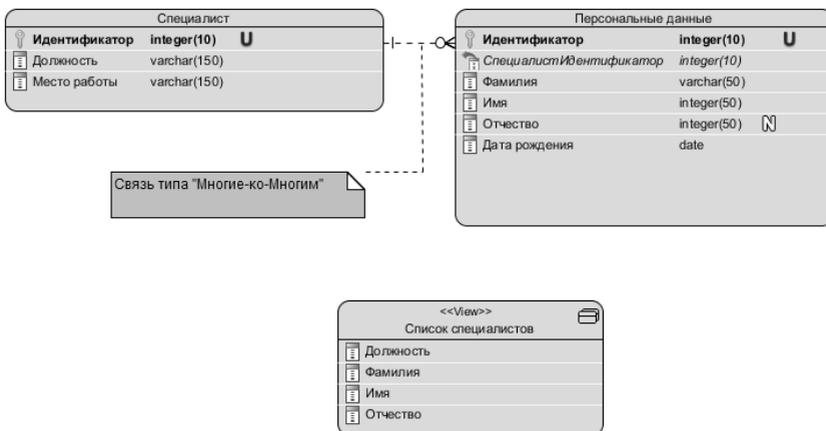


Рисунок 38 – Пример представления данных «Список специалистов»

В данном примере в одной сущности («Список специалистов») объединены атрибуты «Должность», «Фамилия», «Имя»,

«Отчество» (с их значениями) из сущностей «Специалист» и «Персональные данные».

- **Хранимая процедура** – это заранее написанный код процедуры, который можно выполнять снова и снова для проверки или быстрого извлечения данных. Использование хранимых процедур помогает поддерживать согласованную реализацию логики в программных модулях и приложениях. Это также облегчает проектирование, кодирование и тестирование, потому что логика находится в одном месте – хранимой процедуре. В ERD список хранимых процедур размещается в контейнере так, как показано ниже (см. Рисунок 39).

- 

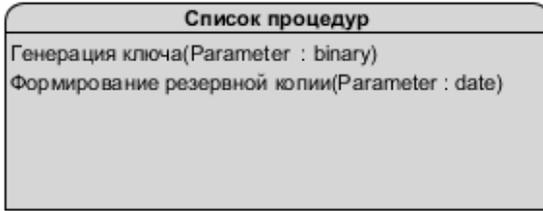


Рисунок 39 – Пример описания списка хранимых процедур

Каждую из хранимых процедур можно, для большей наглядности и читаемости модели, с помощью соединителя (сплошная линия) связать с сущностью, именно для которой эта процедура выполняется.

- **Триггеры** – это процедура, которая автоматически выполняется в ответ на определенные события в БД. Обычно триггеры используются для аудита БД. Триггер, который записывает вставку, изменение и удаление важных данных, сообщит вам, когда и почему были внесены определенные изменения в базу данных. В ERD триггеры располагаются в контейнере так, как показано ниже (см. Рисунок 40).

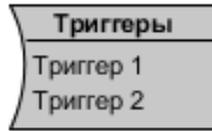


Рисунок 40 – Пример контейнера с триггерами

- **Примечание** – это текстовый блок, который содержит пояснения к элементам, формирующим ERD (см., например, Рисунок 41).

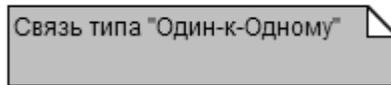


Рисунок 41 – Пример примечания в ERD

### С. Типы моделей ERD

Как уже упоминалось выше, существует три типа моделей ERD: концептуальная, логическая, физическая. Хотя все три типа ERD содержат сущности, включая атрибуты, и отношениями, они различаются по целям, для которых они созданы, и аудитории, на которую они рассчитаны. Общее понимание трех типов моделей данных состоит в том, что бизнес-аналитик использует концептуальную и логическую модели для моделирования бизнес-объектов, которые существуют в бизнес-архитектуре объекта автоматизации, и, которые должны быть в разрабатываемой системе, в то время, как разработчик БД или, другими словами, инженер БД<sup>2</sup> разрабатывают на основе

---

<sup>2</sup> По сложившейся практике, разработка физической модели в нотации ERD возлагается на системного аналитика, которому передает результаты своей разработки (концептуальную и логическую модели) бизнес-аналитик. Однако в небольших компаниях или несложных проектах часто роль «бизнес-аналитик» и «системный

концептуальной и логической моделях ERD физическую модель, которая представляет физическую БД, готовую к реализации в реальной СУБД.

В таблице, представленной ниже, показаны элементы нотации, которые обычно используются в каждом из типов модели ERD (см. Таблица 14).

Таблица 14 – Сравнение используемых элементов нотации ERD в различных типах моделей

Элемент нотации ERD	Тип модели ERD		
	Концептуальная	Логическая	Физическая
Сущность	Да	Да	Да
Атрибут сущности	При необходимости	Да	Да
Тип атрибута	Нет	При необходимости	Да
Связь	Да	Да	Да
Первичный ключ	Нет	Нет	Да
Внешний ключ	Нет	Нет	Да
Представление данных	При необходимости	При необходимости	При наличии
Хранимые процедуры	При необходимости	При необходимости	При наличии
Триггеры	При необходимости	При необходимости	При наличии
Примечание	При необходимости	При необходимости	При необходимости

- **Концептуальная модель ERD** – моделирует бизнес-объекты, которые должны существовать в системе, и связи между ними. Концептуальная модель разрабатывается с тем, чтобы представить общую картину системы путем различия используемых бизнес-объектов. Эта модель определяет именно сущности, а не таблицы БД. Например, таблицы со связями типа «Многие-ко-Многим» могут быть описаны, как сущности, в логической или физической модели данных, но в концептуальной модели такие сущности просто показаны, как отношения без кардинальности. Пример концептуальной модели ERD показан ниже (см. Рисунок 42).

---

аналитику» совмещены в одну роль «аналитику», который, собственно, формирует все три типа моделей ERD.

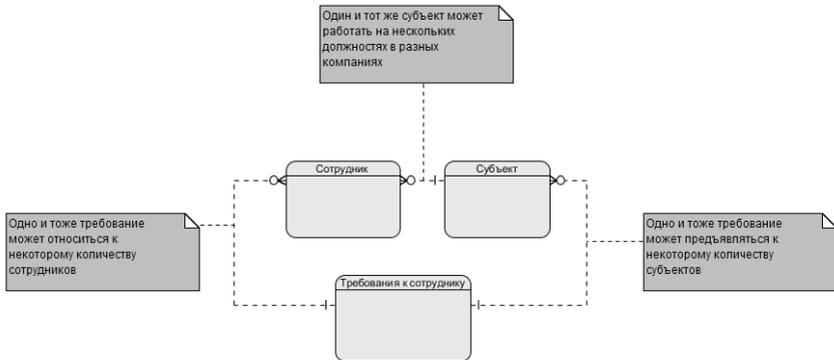


Рисунок 42 – Пример концептуальной модели ERD

В примере выше бизнес-аналитик рассмотрел некоторые данные, в которых отражается то, что может быть некоторое штатное расписание (сущность «Сотрудник»), с которым связаны требования к сотрудникам (сущность «Требования к сотруднику»), но, при этом, одни и те же требования могут предъявляться к нескольким должностям или не более, чем к одной должности. Поэтому используется связь «Один-ко-Многим». В то же время существуют субъекты (сущность «Субъект»), занимающие определенные должности, и один субъект может совмещать должности, что отражено связью «Один-ко-Многим» между сущностями «Субъект» и «Сотрудник». Но поскольку к каждой должности предъявляются различные требования, то и к субъекту может быть предъявлено несколько требований (см. связь «Один-ко-Многим» между сущностями «Требования к сотруднику» и «Субъект»).

- **Логическая модель ERD** – это детализированный вариант концептуальной ERD. Логическая модель разработана для расширения концептуальной модели путем явного определения атрибутов в каждой сущности и введения операционных и транзакционных сущностей. Хотя логическая модель данных по-прежнему не зависит от реальной СУБД, в которой будет создаваться БД, но особенности конкретной СУБД можно принять (а чаще всего и нужно!) во внимание, если это повлияет на дизайн ERD. Пример, на котором показаны отличия концепту-

альной и логической моделей, а также переход от одного типа модели к другому показаны ниже (см. Рисунок 43).

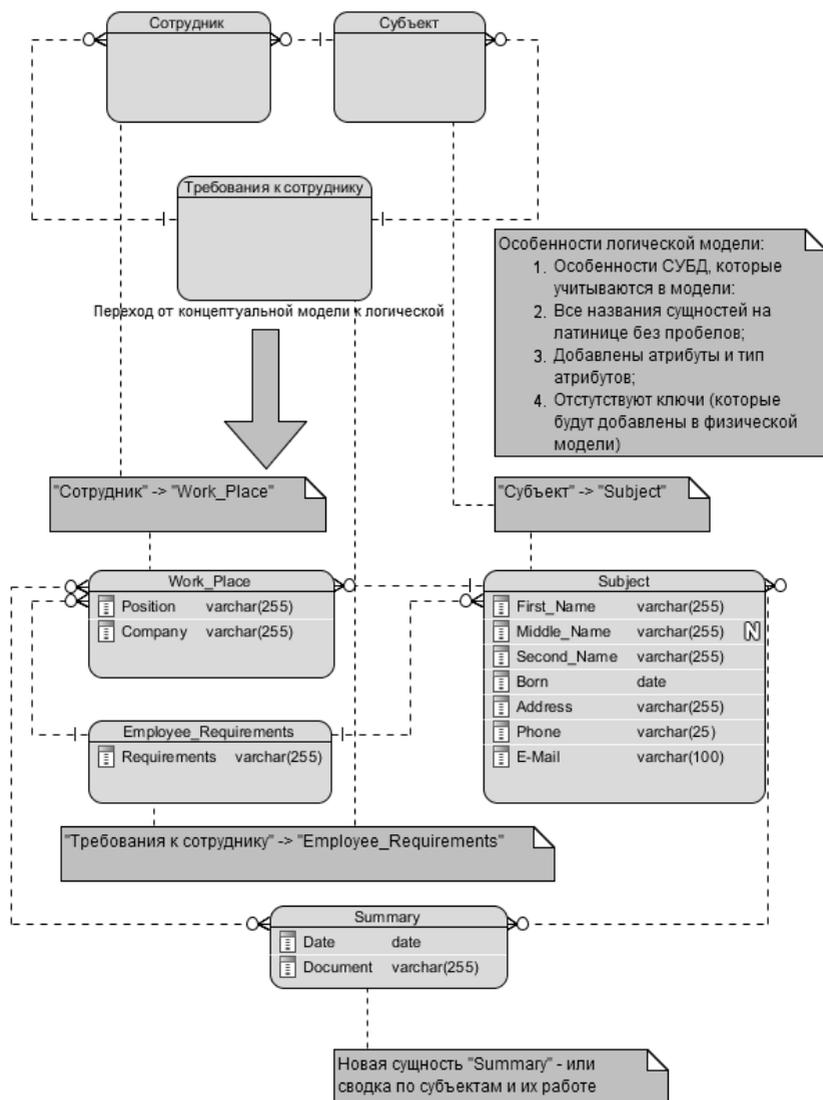


Рисунок 43 – Различия между концептуальной и логической моделями

При переходе от концептуальной модели к логической можно заметить ряд различий:

1. Если в концептуальной модели рассматриваются сущности, но приводятся атрибуты этих сущностей, то в логической модели сущности приведены с атрибутами, а также с указанием типа для значений атрибутов. Как уже указывалось выше, логическая модель может быть независимой от того, на какой СУБД будет реализована БД, но в большинстве случаев лучше учитывать эту особенность. Например, в приведенном примере, предполагается, что БД будет реализована на СУБД PostgreSQL или Microsoft SQL Server, а отсюда типы для атрибутов;

2. В концептуальной модели допустимо писать имена сущностей на языке, используемом в конкретной стране (в данном случае на русском). В принципе, логическая модель тоже допускает такое наименование, но, если выбрана конкретная СУБД для реализации (см. п. 1), то необходимо учитывать ее особенности. В данном случае сущности переименованы на латиницу и, если необходимо в названии использовать несколько слов, то эти слова объединяют с помощью символа «\_»;

3. Знатоки в области теории БД могут спросить – а где ключи? Аналогично концептуальной модели, в логической модели ключи, как правило, не указывают, но указание ключей, если это по каким-либо причинам требуется для моделирования, не будет ошибкой.

• **Физическая модель ERD** – это фактически проект реляционной БД. Физическая модель данных развивает логическую модель через указание каждому атрибуту типа, длины, допустимости нулевого значения и т. д. Поскольку физическая ERD представляет то, как данные должны быть структурированы и связаны в конкретной СУБД, важно учитывать соглашения и ограничения фактической СУБД, в которой будет создана БД. Необходимо убедиться, что типы атрибутов поддерживаются в СУБД, а зарезервированные слова не используются при именовании сущностей и атрибутов. В настоящей работе рассматриваются задачи, решаемые бизнес-аналитиком, а формирование физической модели, как правило, лежит в сфере деятельности системного аналитика и подробно не рассматривается.

## **Д. Выводы или «Как разработать ERD»**

В этом разделе автор дает несколько советов по разработке ERD. Читателю рекомендуется следовать этим рекомендациям, чтобы понять, как эффективно разработать ERD.

1. Нужно четко представить себе всю архитектуру, разрабатываемой системы, т. е. какие информационные объекты моделируются, и в каких информационных бизнес-процессах будет использована информация об этих объектах;

2. Нужно убедиться, что хорошо понятна область применения модели данных или, другими словами, видение разработчика совпадает с видением функционального заказчика. Это позволит не включать избыточные сущности в модель;

3. Первой разрабатывается концептуальная модель, которая включает сущности, на языке понятном заказчику, и связи между ними. И обязательно – комментарии, комментарии, комментарии, в которых лаконично указывается, ЧТО это за сущность и, ПОЧЕМУ выбрана та или иная связь. Эта модель должна обязательно сопровождаться пояснительной запиской, в которой описывается логика выбора элементов модели. Хорошая практика – представить для рецензирования функциональному заказчику модель и пояснительную записку. И получить согласующую подпись ответственного сотрудника со стороны заказчика;

4. Следующей разрабатывается логическая модель путем добавления в сущности атрибутов;

5. Необходимо внимательно изучить ERD и убедиться, что сущностей и атрибутов достаточно для хранения данных. В противном случае, необходимо рассмотреть возможность вставки дополнительных сущностей и атрибутов. На этом этапе можно идентифицировать некоторые транзакционные, операционные и событийные сущности;

6. Необходимо рассмотреть связи между всеми сущностями и соотнести их с надлежащим количеством сущностей (например, отношение «Один-ко-многим» между сущностями «Заказчик» и «Заказ»). Если есть одиночные сущности, то не стоит считать это ошибкой. Хотя это не распространено, но допустимо;

7. Необходимо применять нормализацию БД, чтобы реструктурировать сущности таким образом, чтобы уменьшить избыточность и улучшить целостность данных. Например, сведения о производителе могут изначально храниться в сущности «Продукт». В процессе нормализации можно обнаружить, что одно и то же значение повторяется в разных записях. Можно выделить, например, отдельную сущность «Производитель» и через внешний ключ связать сущности «Продукт» и «Производитель».

Впрочем, у каждого разработчика БД появляются свои секреты и секретки, которые автор, возможно, не знает. Так, что – держайте.

## **Е. Вопросы и задачи для повторения материала**

1. Заказчиком ИС является государственная структура или заказчик в ТЗ в требованиях к документированию написал, что документация должна оформляться в соответствии с группой ГОСТ 34\*. Почему недопустим часто практикуемый в настоящее время подход к разработке программного обеспечения, известный, как «code first», т. е. структура данных формируется программистом по своему усмотрению в процессе написания приложения?

2. В некоторой организации существует следующая структура: руководитель, заместитель руководителя, бухгалтер, эксперты. Эксперты могут быть, как штатными сотрудниками, так и совместителями. Штатные сотрудники получают фиксированный оклад, а совместители получают вознаграждение в зависимости от отработанного времени в соответствии со стоимостью нормо-часа. Руководитель формирует список заданий для экспертов (с описанием задачи и сроков выполнения), а заместитель руководителя распределяет задания между исполнителями. Список заданий передается бухгалтеру для начисления зарплаты с учетом следующих особенностей: штатные сотрудники получают ежемесячно зарплату, но, если в течение полугода было сорвано более двух заданий – сотрудник должен получить уведомление о возможности увольнения, если в течение полугода было сорвано три и более заданий – сотрудника увольняют;

совместители получают вознаграждение, если задание отмечается руководителем или заместителем руководителя, как выполненное. Задания передаются исполнителям. *Необходимо разработать модель данных в нотации ERD (концептуальная модель), которая будет отражать информационную структуру организации и содержание информационных потоков в процессе выполнения задач участниками процессов. Концептуальную модель преобразовать в физическую.* При решении задачи необходимо представить, что вы взаимодействуете с сотрудниками организации, как бизнес-аналитик, согласно процессам, которые описывает ГЛАВА 2. Недостающие данные, которые, как вам кажется, необходимы для построения модели, добавьте самостоятельно – как будто вы получили недостающую информацию от сотрудников организации.

3. Для организации, описанной в разделе «ГЛАВА 4», необходимо разработать описание данных в виде концептуальной модели ERD для BPMN-моделей «Как будет». Преобразовать в логическую модель и обосновать выбор типа данных для каждой из сущностей. Также описать выбор того или иного типа связей между сущностями. Для продвинутых знатоков теории БД, понимающих роль функций, процедур и триггеров, предлагается включить в ERD-модель эти объекты, описать их и обосновать необходимость использования.

## ЗАКЛЮЧЕНИЕ К ТОМУ 1

Настоящий том двухтомника «Теория и практика бизнес-анализа в ИТ. Учебное пособие. Том I» содержит введение в предмет, а также описывает основные нотации, которые необходимы бизнес-аналитику для отражения некоторой ПрО в виде набора документов, которые должны позволить членам группы разработки ИС однозначно понять требования заказчика и, самое главное, правильно реализовать эти требования.

Автор надеется, что становится понятной роль бизнес-аналитика, который берется за очень непростые задачи: понять истинные потребности заказчика ИС и добиться того, чтобы эти потребности были реализованы в полном объеме в конце разработки.

Для простых задач может оказаться, что достаточно документации, которая содержит иллюстрации в нотациях BPMN и ERD. Но полное описание постановки задачи должно содержать описание перехода от модели «Как есть» к системной модели верхнего уровня «Как будет», которая, в значительной степени, абстрагируется от ПрО заказчика и становится понятной для реализации остальным членам группы разработки. А тут не обойтись без нотации универсального языка моделирования (UML и SysML), модели которого позволяют увидеть контуры будущей системы архитектуру ИС (или, как думается автору, и, как вернее отражает суть его деятельности, системному аналитику<sup>3</sup>).

Кроме того, для того, чтобы донести до остальных членов группы разработки авторское видение, желательно (а вернее сказать: необходимо) представить пользовательские сценарии и прототипы пользовательских интерфейсов. По сути две стороны одной медали.

Задачи моделирования в нотации UML, а также сценарии и прототипы – это задачи, которые близки к искусству<sup>4</sup>. Поэтому автор посчитал необходимым вынести этот материал в отдельный том.

---

<sup>3</sup> Не буду спорить ибо, как гласит русская народная мудрость: «Назови хоть горшком – только в печь не сажай»...

<sup>4</sup> Если читателю не хватит фантазии писать сценарии для ИС, то его вполне возьмут к себе кинематографисты – по сравнению с «сюжетом» ИС сценарии для кино покажутся сущей ерундой. Прощу прощения за небольшой плагиат из известной шутки.

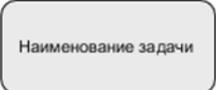
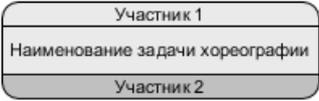
## ПРИЛОЖЕНИЕ 1 ЭЛЕМЕНТЫ НОТАЦИИ ВРМН

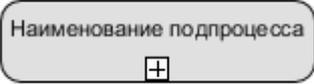
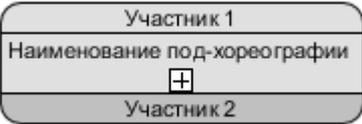
Элементы нотации ВРМН, соответствующие документу [15], представлены ниже (см. Таблица 15).

Таблица 15 – Элементы нотации ВРМН

Элемент	Описание	Нотация
Событие	Событие – это то, что «происходит» в ходе процесса или хореографии. Эти события влияют на переходы в модели и обычно имеют причину (триггер) или исход (результат). События – обозначаются кружками с открытыми центрами, что позволяет размещать внутренние маркеры, которые позволяют различать разные триггеры или результаты. Существует три типа событий, в зависимости от того, когда они влияют на поток: начальное, промежуточное и конечное.	
<b>Направление потока</b> (т. е. определяет «начальное событие», «промежуточное событие» или «конечное событие»)	Как следует из названия («Начальное событие»), этот элемент указывает на то, где расположено начало в моделях типа «Оркестровка» или «Хореография».	
Начальное событие		
Промежуточное событие	Промежуточные события происходят между начальным и конечным событиями. Они будут влиять на ход процессов в моделях типа «Оркестровка» или «Хореография», но не будут запускать или (напрямую) прерывать процесс.	
Конечное событие	Как следует из названия, Конечное событие указывает, где закончится модель типа «Оркестровка» или «Хореография».	

Элемент	Описание	Нотация				
<p><b>Тип направления потока</b> (т. е. «не типизирован», «сообщение», «по таймеру», «ошибка», «отмена», «компенсация», «по условию», «ссылка», «по сигналу», «распараллеливание», «завершение»)</p>	<p>Начальные и некоторые промежуточные события могут иметь триггеры, которые определяют причину события. Существует несколько способов определения триггеров. В конечных событиях <b>МОГУТ</b> быть определены «результаты», которыми завершаются последовательности.</p> <p><b>Стартовые события</b> могут реагировать только на входные триггеры.</p> <p><b>Конечные события</b> могут создавать только выходные результаты.</p> <p><b>Промежуточные события</b> могут содержать входные триггеры или выходные результаты. Для событий, с входными триггерами, маркеры не заполнены, а для выходных триггеров и результатов, маркеры заполнены.</p>	Тип триггера или события				
		Улавливающий	Выходной	Без прерывания		
		Сообщение (стр. 254)				
		По таймеру				
		Ошибка				
		Расширение				
		Отмена				
		Компенсация				
		По условию				
		Ссылка				
		По сигналу				
Многократное событие						
Распараллеливание многократных событий						
Завершение						

Элемент	Описание	Нотация
Деятельность	Деятельность – это общий термин для той работы, которая выполняется в процессе. Деятельность может быть атомарной или неатомарной (составной). Типы деятельности, которые являются частью процессной модели: подпроцесс и задача, которые отображают в виде прямоугольника со скругленными углами. Элемент «Деятельность» используются как в моделях типа «Оркестровка», так и в моделях типа «Хореография».	
Задача (атомарная деятельность)	Задача – это атомарное действие, включаемое в процесс. Задача используется в том случае, когда работа в Процессе не разбита на более тонкий уровень детализации Процесса.	
Задача хореографии	Задача хореографии – это атомарная деятельность в моделях типа «Хореография». Она представляет собой набор из одного или более обменов сообщениями. В каждой хореографической задаче участвуют два участника. Наименование задачи хореографии и каждого из участников отображаются в разных полосах, которые составляют графическое обозначение фигуры.	

Элемент	Описание	Нотация
Процесс/подпроцесс (неатомарный)	Подпроцесс – это составная деятельность, включаемая в модель типа «Оркестровка» или модель типа «Хореография». Подпроцесс можно разбить на более мелкий уровень детализации («Оркестровка» или «Хореография») с помощью наборов задач.	Различные типы подпроцессов показаны на следующих четырех рисунках
Свернутый подпроцесс	В данном представлении детали подпроцесса не видны на диаграмме. Знак «плюс» в нижнем центре фигуры указывает, что деятельность содержит подпроцесс и имеет нижний уровень детализации.	
Раскрытый подпроцесс	Граница подпроцесса расширена и детали (процессы) видны внутри его границы. Заметим, что последовательности не должны пересекать границу подпроцесса.	
Свернутая под-хореография	Детали под-хореографии не видны на диаграмме. Знак «плюс» в центре в нижней части полосы наименования задачи указывает на то, что деятельность содержит подпроцесс и имеет нижний уровень детализации.	

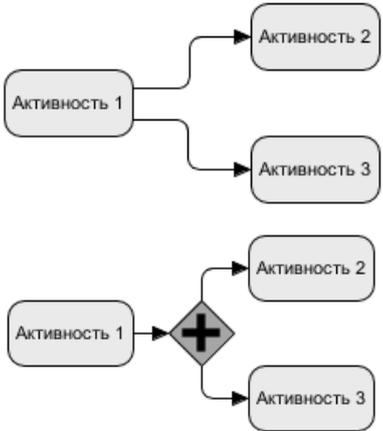
Элемент	Описание	Нотация
Раскрытая под-хореография	Граница под-хореографии расширена и детали (хореографии) видны внутри ее границы. Заметим, что последовательности не должны пересекать границу под-хореографии	
Шлюз	Шлюз используется для управления разветвлением и сходимостью потоков в моделях типа «Оркестровка» и в моделях типа «Хореография». Таким образом, шлюз определяет ветвление, разветвление, слияние и присоединение потоков. Внутренние маркеры указывают тип поведения	
Типы управления в шлюзах	<p>Значки внутри ромба, обозначающего шлюз, будут указывать тип управления потоком. Типы управления могут быть следующие:</p> <ul style="list-style-type: none"> <li>• Исключительное разделение и слияние. Как исключительные, так и основанные на событиях, шлюзы выполняют исключительное разделение и слияния;</li> <li>• Шлюзы на основе событий и параллельных событий могут запускать новый экземпляр процесса;</li> <li>• Разделяющий и объединяющий включающий шлюз;</li> </ul>	<p>Исключающий шлюз </p> <p>Шлюз на основе событий </p> <p>Включающий шлюз </p> <p>Комплексный шлюз </p>

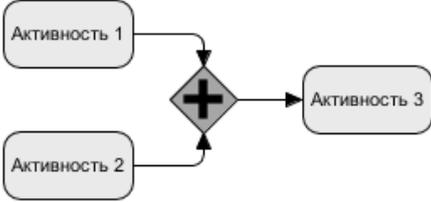
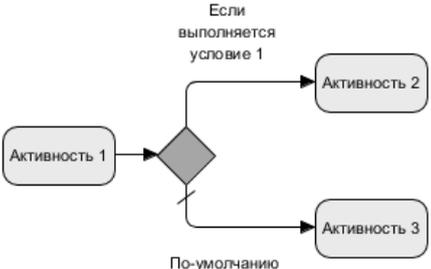
Элемент	Описание	Нотация
	<ul style="list-style-type: none"> <li>• Комплексный шлюз предназначен для сложных условий и ситуаций;</li> <li>• Параллельный шлюз разветвляет и присоединяет.</li> </ul> <p>Каждый тип управления влияет как на входящий, так и на исходящий потоки</p>	<p>Параллельный шлюз</p> 
Поток последовательности	Поток последовательности используется для отображения порядка выполнения операций в моделях типа «Оркестровка» и типа «Хореография»	Конкретные типы потоков последовательностей на следующих семи рисунках
Обычный поток	Обычный поток указывает направления потока последовательности, но, который не может начинаться с промежуточного события, связанного с границей активности	
Неуправляемый поток	Неуправляемый поток соответствует потоку, на который не влияют какие-либо условия, или, который не проходит через Шлюз. Простейшим примером этого является одиночный Поток Последовательностей, соединяющий две Активности. Это также может применяться к множественным Потокам Последовательностей, которые сходятся к Активности или расходятся от неё. Для каждого неуправляемого Потока Последовательности направление распространения будет направлено от исходного объекта	

Элемент	Описание	Нотация
	<p>посредством Поточков Последовательности к целевому объекту.</p>	
Условный поток	<p>Поток последовательности может иметь выражение условия, которое оценивается во время выполнения, чтобы определить, будет ли использоваться поток последовательности (т. е. будет ли перемещение по данному потоку последовательности). Если условный поток является исходящим из элемента «Действие», то поток последовательности будет иметь мини-ромб в начале соединителя. Если условный поток исходит от шлюза, то на линии не будет мини-ромба (как для неуправляемого потока).</p>	
Поток по умолчанию	<p>Для исключающих шлюзов на основе данных или включающих шлюзов одним из типов потока является условный поток по умолчанию. Этот поток будет использоваться только в том случае, если все остальные исходящие условные потоки не соответствуют условиям во время выполнения. Эти потоки последовательности будут иметь диагональную косую черту, которая будет добавлена в начало соединителя.</p>	

Элемент	Описание	Нотация
Поток исключение	Поток исключение происходит, если процесс завершается с отклонениями от нормального течения, и основан на Промежуточном событии, прикрепленном к границе действия, которое происходит во время выполнения Процесса.	
Поток сообщений	Поток сообщений используется для отображения потока сообщений между двумя участниками, которые готовы отправлять и получать их.	
Компенсационная Ассоциация	Компенсационная Ассоциация происходит вне обычного потока Активности и основана на Промежуточном событии типа «Компенсация», которое инициируется в результате сбоя транзакции или события, требующем компенсации исключения. Целью Компенсационной Ассоциации ДОЛЖНА быть Активность, помеченная, как компенсационная.	

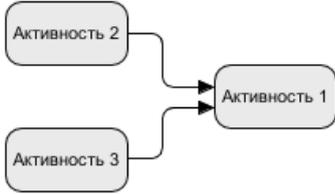
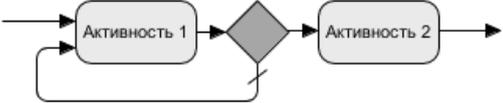
Элемент	Описание	Ноотация
Объект данных	Объекты данных предоставляют информацию о том, какие данные нужно получить объекту типа «Деятельность», или, какие данные он производит. Объекты данных могут представлять отдельный объект или набор объектов. Ввод и вывод данных предоставляют одинаковую информацию для процессов.	 <p>Объект данных</p>  <p>Объект данных (набор данных)</p>  <p>Ввод данных</p>  <p>Вывод данных</p>
Сообщение	Элемент «Сообщение» используется, чтобы изобразить какими сообщениями обмениваются два Участника.	

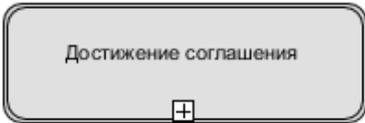
Элемент	Описание	Нотация
Разветвление	<p>В BPMN используется термин «разветвление» для обозначения разделения пути на два или более параллельных пути (также известное как разделение AND). Это место в процессе, где действия могут выполняться одновременно, а не последовательно.</p> <p>Существует два варианта:</p> <ul style="list-style-type: none"> <li>• Несколько исходящих потоков. «Неуправляемый» поток является предпочтительным методом для большинства ситуаций;</li> <li>• Параллельный шлюз. Этот метод используется редко, обычно в сочетании с другими шлюзами</li> </ul>	

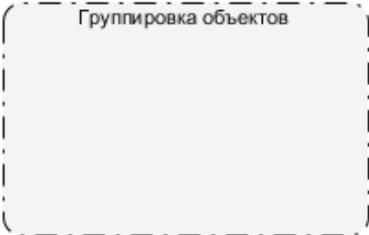
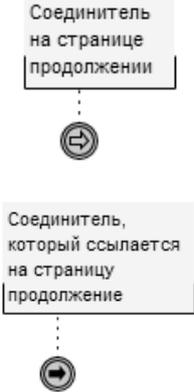
Элемент	Описание	Нотация
Слияние	<p>В BPMN используется термин «слияние» для обозначения объединения двух или более параллельных путей в один путь (также известный, как «Слияние-И» или синхронизация).</p> <p>Чтобы показать слияние нескольких потоков, используется параллельный шлюз.</p>	
Решение, Точка ветвления	<p>Решение – это шлюз, используемый в моделях «Оркестровка» или «Хореография», который может определить один или несколько альтернативных путей.</p>	<p>См. ниже пять вариантов использования</p>
Исключающая точка ветвления	<p>Это решение представляет собой точку ветвления, в которой альтернативы основаны на условных выражениях, содержащихся в исходящих потоках. Может быть выбрана только одна из альтернатив.</p>	

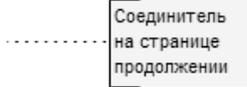
Элемент	Описание	Нотация
<p>Точка ветвления на основе события</p>	<p>Это точка ветвления, в которой выбор альтернативы основан на событии, которое произошло перед этой точкой. Конкретное событие (обычно получение сообщения) определяет, какой из путей будет выбран. Могут быть использованы другие типы событий, такие как, например, таймер. В результате будет выбрана только одна из альтернатив.</p> <p>Возможны два варианта получения сообщений из точки ветвления на основе событий:</p> <ul style="list-style-type: none"> <li>• Можно использовать элемент «Активность» типа «Принимающая задача»;</li> <li>• Можно использовать промежуточные события типа «Сообщение».</li> </ul>	
<p>Включающая точка ветвления</p>	<p>Эта точка ветвления, в которой альтернативы основаны на условных выражениях, содержащихся в исходящих потоках последовательностей.</p> <p>В некотором смысле это группа связанных независимых двоичных (ДА/НЕТ) решений. Поскольку каждый путь независим, все комби-</p>	

Элемент	Описание	Нотация
	<p>нации путей МОГУТ быть использованы: от нуля до использования всех путей. Однако этот элемент должен быть спроектирован так, чтобы был выбран хотя бы один путь. Условие по умолчанию может использоваться для гарантии того, что выбран хотя бы один путь.</p> <p>Есть два подхода для точки ветвления этого типа:</p> <ul style="list-style-type: none"> <li>• Первый подход использует коллекцию условных потоков, отмеченных мини-ромбами;</li> <li>• Второй подход использует включающий шлюз.</li> </ul>	
Объединяющая точка ветвления	<p>BPMN использует термин «объединение» для обозначения исключительного объединения двух или более путей в один путь (также известный как Исключение-ИЛИ). Исключающий шлюз используется для отображения объединения нескольких потоков последовательностей.</p> <p>Если все входящие потоки являются альтернативными, то шлюз не нужен. Неуправляемый поток</p>	

Элемент	Описание	Нотация
	<p>обеспечивает такое же поведение точки ветвления.</p>	
Цикличность	<p> BPMN предоставляет два механизма отображения цикличности.</p>	<p>См. два элемента нотации ниже</p>
Циклическая деятельность	<p>Атрибуты задач и подпроцессов определяют, будут ли они повторяться или выполняться один раз. Существует два типа циклов: Стандартный и Множественный. Небольшой индикатор цикличности будет отображаться внизу в центре активности.</p>	
Цикл потока последовательностей	<p>Цикл может быть создан путем замыкания потока выходящего потока последовательностей на объект, расположенный в начале замыкаемого потока, при выполнении некоторого условия.</p>	

Элемент	Описание	Нотация
<p>Множество экземпляров</p>	<p>Атрибуты процессов и подпроцессов определяют, будут ли они повторяться или выполняться один раз. Набор из трех вертикальных линий будет отображаться внизу в центре деятельности для параллельных множественных экземпляров.</p>	
<p>Прерывание процесса (некое воздействие делает паузу)</p>	<p>Прерывание процесса – это место в последовательности активностей, в котором прямо указано, где в последовательности произойдет ожидаемая задержка. Промежуточное событие используется для отображения фактического поведения. Кроме того, с событием можно связать артефакт разрыва процесса, предусмотренный разработчиком модели или инструментом моделирования, чтобы выделить место задержки в потоке.</p>	
<p>Транзакция</p>	<p>Транзакция – это подпроцесс, который поддерживается специальным протоколом, и, который гарантирует, что все вовлеченные стороны имеют полное согласие с тем, что действие должно быть завершено или отменено. Атрибуты деятельности будут определять, является ли действие транзакцией. Граница с двойной линией означает, что подпроцесс является транзакцией.</p>	

Элемент	Описание	Нотация
<p>Группировка (граница вокруг группу объектов, входящих в одну категорию)</p>	<p>Группировка – это группа графических элементов, которые могут быть отнесены к одной категории. Этот элемент не влияет на потоки последовательности внутри группировки.</p> <p>Название группировки отображается на диаграмме как метка. Группировки могут быть использованы для документирования или анализа. Это один из способов визуального отображения категорий объектов в модели</p>	
<p>Междустраничный соединитель</p>	<p>Этот элемент используется для учета возможностей печати и показывает где поток последовательности покидает одну страницу, а затем перезапускается на следующей странице. Промежуточное событие «Междустраничный соединитель» может быть использовано в качестве «соединителя» страниц</p>	
<p>Ассоциация</p>	<p>Ассоциация используется для связи информации и артефактов с графическими элементами BPMN,</p>	

Элемент	Описание	Нотация
	<p>например, текстовыми примечаниями и объектами данных. Стрелка на конце ассоциации, если применимо, показывает направление потока (как правило, это связано с данными)</p>	
<p>Текстовое примечание (прикрепляемое посредством ассоциации)</p>	<p>Текстовое примечание является механизмом, который использует разработчик модели для пояснений читателю модели BPMN</p>	
<p>Пул</p>	<p>Пул – это графическое представление участника, участвующего в совместной деятельности. Он также может выступать в качестве «плавательной дорожки» и графического контейнера для разделения набора операций из других пулов, обычно в контексте ситуаций В2В. Пул МОЖЕТ иметь внутренние детали в форме Процессов, которые могут выполняться. Или пул МОЖЕТ не иметь внутренних деталей, то есть это может быть «черный ящик».</p>	

Элемент	Описание	Нотация
Дорожки	<p>Дорожка – это часть пула, которая расширяет пул по вертикали и горизонтали.</p> <p>Дорожки используются для организации и классификации деятельности.</p>	 <p>The notation diagram consists of a vertical bar on the left side. The bar is divided into two sections: the top section is labeled 'Директор' and the bottom section is labeled 'Заместитель'. To the right of this bar is a large gray rectangular area, which is divided into two horizontal sections by a single line, representing the organizational structure associated with the 'Дорожки' element.</p>

# СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

БА	Бизнес-анализ
БД	База данных
БП	Бизнес-процесс
ГОСТ	Государственный стандарт
ЖЦ	Жизненный цикл
ИС	Информационная система
ИТ	Информационные технологии
НСИ	Нормативно-справочная информация
ПК	Персональный компьютер
ПО	Программное обеспечение
ПрО	Предметная область
РА	Рациональный агент (см. [8])
СУБД	Система управления базами данных
ТЗ	Техническое задание
ЭВМ	Электронная вычислительная машина
BPEL	Business Process Execution Language – язык выполнения бизнес-процессов, см. [13]
BPML	Business Process Modeling Language – язык моделирования бизнес-процессов, см. [12]
BPMN	Business Process Model and Notation – нотация и модель бизнес-процесса, см. [15]
DFD	Data Flow Diagrams – диаграммы потоков данных (см. [22])
ERD	Entity Relationship Diagram или в русской интерпретации: диаграмма сущность-связь
IDEF	Семейство стандартов IDEF0...IDEF14 (см., например, [6])
MOF	Microsoft Operations Framework или в русской интерпретации: набор документов Microsoft по эксплуатации
MSF	Microsoft Solutions Framework или в русской интерпретации: набор документов Microsoft по разработке
OMG	Международная организация Object Management Group – некоммерческий консорциум по разработке технологический стандартов. Интернет ресурс: <a href="https://www.omg.org/">https://www.omg.org/</a>
SADT	Structured Analysis and Design Technique – методология структурного анализа и проектирования (одна из методологий, на основе которой была сформирована группа методологий IDEF*)
SysML	Systems Modeling Language – язык моделирования систем, см. [17]
UML	Unified Modeling Language – унифицированный язык моделирования, см. [16]
XML	eXtensible Markup Language – расширяемый язык разметки, см. [11]
XPDL	Process Definition Language – язык описания процессов, см. [14]

## СПИСОК ЛИТЕРАТУРЫ

1. Берталандфи Л. фон. Общая теория систем – критический обзор // Исследования по общей теории систем: Сборник переводов / Общ. ред. и вст. ст. В. Н. Садовского и Э. Г. Юдина. – М.: Прогресс, 1969.
2. J. Zachman, «A Framework for Information Systems Architecture» // IBM Systems Journal 26, (No. 3, 1987).
3. B. L. Meek, P. M. Heath. Guide to Good Programming Practice (Computers and Their Applications) / Series: Ellis Horwood Series in Computers and Their Applications / Prentice Hall Europe (a Pearson Education company) / May 11, 1980 / ISBN-10: 0853121524 / ISBN-13: 978–0853121527.
4. ГОСТ 34.003–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения» // Москва – Стандартиформ – 2009.
5. Ожегов С. И., Шведова Н. Ю. Толковый словарь русского языка. – М.: «Азъ Ltd.», 1992, – 960 с.
6. Р 50.1.028–2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования // Госстандарт России – 2001.
7. Smith H., Fingar P. Business Process Management: The Third Wave // – Meghan-Kiffer Press, Tampa, Florida, USA, 2003.
8. Stuart Russell and Peter Norvig (2003) Artificial Intelligence: A Modern Approach, Second Edition, Prentice-Hall, 2003.
9. Система // Большой Российский энциклопедический словарь. – М.: БРЭ. – 2003, с. 1437.
10. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. – 8-е изд. – М.: Вильямс, 2005. – 1328 с. – ISBN 5–8459–0788–8 (рус.) 0–321–19784–4 (англ.).
11. Extensible Markup Language (XML) 1.1 (Second Edition). W3C Recommendation 16 August 2006, edited in place 29 September 2006. URL: <https://www.w3.org/TR/xml11/>.

12. Business Process Modeling Language (BPML). URL: [https://www.service-architecture.com/articles/web-services/business\\_process\\_modeling\\_language\\_bpml.html](https://www.service-architecture.com/articles/web-services/business_process_modeling_language_bpml.html).
13. Web Services Business Process Execution Language Version 2.0. OASIS Standard. 11 April 2007. URL: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
14. Workflow Management Coalition Workflow Standard. Process Definition Interface -- XML Process Definition Language / Document Number WFMC-TC-1025 Document Status –Released / August 30, 2012 Version 2.2 / URL: [http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20\(2012-08-30\).pdf](http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20(2012-08-30).pdf).
15. Business Process Model and Notation (BPMN). Version 2.0. URL: <https://www.omg.org/spec/BPMN/2.0/>.
16. OMG® Unified Modeling Language® (OMG UML®). Version 2.5.1. URL: <https://www.omg.org/spec/UML/>.
17. OMG Systems Modeling Language™. Version 1.5. URL: <https://www.omg.org/spec/SysML/>.
18. Буч Г., Рамбо Д., Якобсон Н. «Язык UML. Руководство программиста» // 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2007 – 496 с.: ил.
19. Некрасов С. И., Некрасова Н. А. Философия науки и техники: тематический словарь. – Орёл: ОГУ. 2010.
20. Чередникова А. В., Землякова И. В. Введение в теорию графов // – Кострома, КГТУ, 2011.
21. Information technology – Object Management Group Business Process Model and Notation. International Standard ISO/IEC 19510 / Reference number ISO/IEC 19510:2013(E).
22. ГОСТ Р МЭК 62264–1-2014 Интеграция систем управления предприятием. Часть 1. Модели и терминология // М.: Стандартинформ, 2016.
23. Непейвода Н. Н., Гулиев Я. И., Цветков А. А. Методика анализа и синтеза моделей бизнес-процессов в медицинской организации / **А. А. Цветков** // – Врач и информационные технологии / 2015 / № 4/ стр. 14–23 / УДК 61:007.
24. ГОСТ Р 43.0.4-2009 Информационное обеспечение техники и операторской деятельности. Информация в технической деятельности. Общие положения // М.: Стандартинформ, 2018.

25. ГОСТ Р 43.2.1–2007 Информационное обеспечение техники и операторской деятельности. Язык операторской деятельности. Общие положения // М.: Стандартинформ, 2018.

26. ГОСТ Р 51275–2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения // Стандартинформ, 2018.

27. ГОСТ Р 50922–2006. Защита информации. Основные термины и определения // Стандартинформ, 2008.

28. Федеральном законе от 27.07.2006 N 149-ФЗ (ред. от 18.12.2018) «Об информации, информационных технологиях и о защите информации» // URL: <http://www.consultant.ru/cons/cgi/online.cgi?req=doc&base=LAW&n=313796&fld=134&dst=1000000001,0&rnd=0.18199737002639793#08591695246202944>.

29. ГОСТ Р ИСО/МЭК 15288–2005. Информационная технология. Системная инженерия. Процессы жизненного цикла систем // М.: Стандартинформ, 2006.

30. ГОСТ Р ИСО/МЭК 12207–2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств // М.: Стандартинформ, 2011.

31. Кумагина, Е.А., Неймарк, Е. А. Модели жизненного цикла и технологии проектирования программного обеспечения: учебно-методическое пособие / Е. А. Кумагина, Е. А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.

32. Гибкая методология разработки программного обеспечения. © Microsoft Corporation, 2006–2008 // Москва – Изд-во «Русская редакция» – 2008. URL: [http://download.microsoft.com/documents/rus/msdn/msfa2009\\_w.pdf](http://download.microsoft.com/documents/rus/msdn/msfa2009_w.pdf).

## ОБ АВТОРЕ



**Цветков Алексей Анатольевич**, научный сотрудник Института программных систем им. А. К. Айламазяна Российской академии наук.

Круг научных интересов: теория и практика разработки ИС, системы искусственного интеллекта, теория сверхбольших баз данных, исследования в области недифференцируемых функций.

Автор многочисленных научных работ в отечественных и зарубежных научных журналах.

**Алексей Анатольевич Цветков**

**Теория и практика  
бизнес-анализа в ИТ**

**Том I**

*Учебное пособие*

Ответственный редактор *А. Иванова*

Корректор *М. Глаголева*

Верстальщик *М. Глаголева*

Издательство «Директ-Медиа»  
117342, Москва, ул. Обручева, 34/63, стр. 1  
Тел/факс + 7 (495) 334–72–11  
E-mail: [manager@directmedia.ru](mailto:manager@directmedia.ru)  
[www.biblioclub.ru](http://www.biblioclub.ru)  
[www.directmedia.ru](http://www.directmedia.ru)