

Министерство сельского хозяйства Российской Федерации
ФГБОУ ВО «Красноярский государственный аграрный университет»

И.В. Миндалёв

Моделирование бизнес-процессов с помощью IDEF0, DFD, BPMN за 7 дней

Рекомендовано научно-методическим советом федерального государственного бюджетного образовательного учреждения высшего образования «Красноярский государственный аграрный университет» для внутривузовского использования в качестве учебного пособия для студентов, обучающихся по направлениям подготовки 38.03.05 «Бизнес-информатика», 09.03.03 «Прикладная информатика»

Красноярск 2016

Рецензенты:

И.О. Егорушкин, канд. техн. наук, доцент кафедры математических и естественно-научных дисциплин Красноярского института железнодорожного транспорта

Л.Н. Шевцова, канд. с-х. наук, доцент кафедры математических методов и информационных технологий Сибирского федерального университета

Миндалёв, И.В.

Моделирование бизнес-процессов с помощью IDEF0, DFD, BPMN за 7 дней: учеб. пособие / И.В. Миндалёв; Краснояр. гос. аграр. ун-т. – Красноярск, 2016. – 123 с.

Издание представляет собой практическое руководство по созданию функциональных моделей с помощью CASE-средств – Ramus Education и Modelio. Содержит описание методов анализа и проектирования моделей бизнес-процессов в объеме, необходимом для практической работы. На конкретном примере рассмотрено применение CASE-средств по созданию функциональной модели предприятия по производству пельменей.

Предназначено для студентов, изучающих дисциплину «Моделирование бизнес-процессов» и выполняющих курсовые проекты по проектированию информационных систем по специальностям: 09.03.03 «Прикладная информатика», 38.03.05 «Бизнес-информатика».

© Миндалёв И.В., 2016

© ФГБОУ ВО «Красноярский государственный аграрный университет», 2016

ВВЕДЕНИЕ

Мировой опыт свидетельствует о том, что внедрению на предприятии информационных систем должно предшествовать серьезное функционально-информационное обследование предприятия с целью определения оптимальности процессов, распределения ресурсов между функциями и т. п. Созданная в результате такого обследования бизнес-модель может создаваться с помощью различных инструментов. В настоящем пособии рассматривается технология моделирования бизнес-процессов с помощью CASE-средств – Ramus Education и Modelio.

Реализация моделирования в Ramus Education базируется на методологии IDEF0, которая имеет в России и США статус стандарта [6], [19]. IDEF0 входит в семейство методологий IDEF, позволяющих исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем.

Метод IDEF0 предназначен для функционального моделирования, то есть моделирования выполнения функций объекта путем создания описательной графической модели, показывающей, что, как и кем делается в рамках функционирования предприятия. Функциональная модель представляет собой структурированное изображение функций производственной системы или среды, информации и объектов, связывающих эти функции.

CASE-средство Modelio использует методологии UML и BPMN. Метод BPMN – представляет собой графическую нотацию для отображения бизнес-процессов при моделировании потоков работ, происходящих в исследуемой системе [4].

На основе формализованного описания бизнес-процессов можно решать следующие задачи:

- анализ и реформирование бизнес-процессов;
- проектирование информационной инфраструктуры, процедур и регламентов информационного взаимодействия;
- разработка регламентов и процедур обеспечения качества продукции и создание систем обработки данных о качестве;
- разработка систем документооборота, должностных инструкций [7].

При создании информационных систем важно обеспечить эффективное взаимодействие между специалистами (менеджерами) и

разработчиками (программистами). Семейство IDEF и BPMN предоставляет средства отображения не только бизнес-процессов, но и методологию взаимодействия «специалист–разработчик». Поэтому изучение этих методов необходимо специалистам как в области информационных технологий, так и в области менеджмента.

Учебное пособие рассчитано на обучение в течение семи дней.

1-й день – вводит в мир моделирования бизнес-процессов.

2-й день – посвящен основным понятиям методологии функционального моделирования IDEF0.

3-й день – установка приложений Ramus Education и Modelio.

4-й день – создание функциональной модели деятельности компании по производству пельменей с помощью Ramus Education.

5-й день – основные элементы BPMN, создание диаграмм BPMN с помощью Modelio.

6-й день – реинжиниринг бизнес-процессов компании по производству пельменей с помощью Ramus Education и Modelio.

7-й день – основы методологии DFD, создание диаграмм потоков данных с помощью Ramus Education.

1-й день. МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ

Онегин Ленского спросил:

«Что ж, начинать?» – «Начнём, пожалуй», –
сказал Владимир.

А.С. Пушкин. Евгений Онегин, 6, XXVII

1.1. Задачи менеджера

С наступлением XXI века мы стали жить в компьютерном мире. Так, в России доля домашних хозяйств, имевших персональный компьютер и доступ к сети интернет, составляла в 2014 г. – 71,4 % и 65,1 % [9].

Энтузиасты компьютерной революции предрекали, что появление новой информационной технологии повлечет «куда более радикальные перемены, чем совершившийся сто лет назад переход на электроэнергию». Возможно, компьютерная революция действительно происходит, но она еще не изменила нашу экономику так радикально, как предсказывали ее адепты. Об этом свидетельствует тот факт, что распространение компьютеров не вызвало резкого роста производительности труда. Лауреат Нобелевской премии по экономике Роберт Солоу описал этот парадокс, заметив «что наступление компьютерной эры теперь сказывается буквально во всем, кроме... показателей производительности труда».

Но компьютеры продолжают наступать. И любая технология рано или поздно становится технологией информационной (ИТ). И сам бизнес, и системы управления, и даже культура – всё живет сейчас под знаком ИТ. Однако информационные технологии при всей своей революционности не отменили производственного процесса, не ликвидировали конкурентов и не отняли у человека право принимать решения. Скорее можно говорить об интенсификации всех процессов в компьютерном веке. Изменился инструментарий в управлении компанией. А инструментарий, в свою очередь, очень сильно повлиял на все процессы, к которым имеют отношение менеджеры: планирование, организация, руководство, контроль.

Менеджеру приходится решать кардинальные вопросы, связанные с положением компании на рынке, ответы на которые следует искать в возможностях информационных систем.

Какие же вопросы встают перед руководителем компании?

Как занять свое место на рынке? Ответ можно найти с помощью универсальных статистических приложений и маркетинговых информационных систем.

Как доказать инвесторам, что деньги они должны дать именно вам? Для этого необходимо использовать системы планирования и экспертизы инвестиций.

Как преобразовать предприятие с целью вывести его из неблагоприятной ситуации? Решение – в организационных изменениях, которые опираются на ИТ. К этим изменениям относятся: автоматизация бизнес-процессов, рационализация процедур и реинжиниринг (перепроектирование) бизнеса.

Как обеспечить выпуск продукции мирового уровня? Поскольку качество становится более техничным, процессный подход, составляющий основу ISO 9000, требует специальных средств для описания и классификации процессов, составляющих деятельность предприятий. А описание процессов деятельности предприятия требует использования компьютерной CASE-технологии.

Как отразить угрозы конкурентов? Как обеспечить высокую эффективность компании? Как обеспечить широкую известность своей продукции? Ответы на все эти вопросы руководитель находит с помощью ИТ.

В чем же цель ИТ? В повышении производительности, экономии финансов, подготовке обоснованных решений? Да. Но, это относится к способам достижения тактических, краткосрочных преимуществ. А стратегическая цель ИТ – способствовать менеджменту, реагировать на динамику рынка, создавать, поддерживать и углублять конкурентное преимущество [1].

1.2. Преобразование предприятия

Современные методы преобразования предприятия связаны с ИТ, поэтому преобразования обязательно затронут не только организацию, но и ее информационную систему. Создание ИС приводит к намного большим изменениям, чем смена оборудования и программного обеспечения – она включает изменения в работе, управлении и организации. Когда мы проектируем новую ИС, мы перепроектируем организацию.

Важно знать, что создание новой информационной системы является запланированным организационным изменением. Поэтому

разработчики должны понять, как система будет влиять на организацию в целом, как изменится существо работы под воздействием новой системы.

Информационные системы могут быть мощными инструментами организационных изменений, позволяя организациям перепроектировать свою структуру, область действия, коммуникации, трудовые процессы, изделия и услуги.

Выделяют следующие виды структурных организационных изменений, которые опираются на ИТ: автоматизацию, рационализацию, реинжиниринг и смену устремлений [1].

Что такое реинжиниринг? Реинжиниринг бизнес-процессов (BPR) – это не уменьшение размеров, это не автоматизация или стремление к автономности. Это не реструктуризация, не реорганизация, не уменьшение бюрократии, не снижение простоев и не сокращение уровней управления в организации. Реинжиниринг – это не комплексное управление качеством (TQM). Суть BPR – в реинтеграции процесса. Создатели метода, Хэммер и Чемли, определяют BPR как «идею объединения задач (которые были прежде разбиты) в связанные процессы в бизнесе».

Бизнес-процесс – это набор логически связанных задач, выполняемых для достижения определенного делового результата. Примеры деловых процессов: производство пельменей, заказ сырья у поставщика. При использовании ИТ организации могут переосмысливать бизнес-процессы и направления их движения, чтобы улучшить скорость, обслуживание и качество. Реинжиниринг использует ИТ для достижения конкурентных целей: минимизации числа операций и частей, минимизации издержек и работы, не вносящей вклада в добавленную стоимость, минимизации разделения труда и минимизации времени доставки и получения оплаты – короче говоря, для максимизации полезности.

Для реинжиниринга бизнеса компании нужно выполнить следующие действия:

- понять, что представляет собой бизнес-процесс компании – этому помогает моделирование бизнес-процессов;
- вложить новые идеи в реорганизацию компании – это задачи стратегического планирования;
- реализовать перепроектирование с помощью современных компьютерных методов и средств – CASE-технологий [1].

1.3. Способы разработки моделей

При разработке методик моделирования бизнес-процессов использовался опыт моделирования информационных процессов.

К традиционным способам разработки моделей относят следующие методы: структурный анализ и структурное проектирование (SA/SD), методология IDEF, методика SADT [1].

Эти методики, основанные на моделировании ИС, исходят из следующей парадигмы. При описании ИС предполагается, что она содержит два типа сущностей: некоторый аналог программы (активные сущности которые выполняют некоторую обработку) и данные (пассивные сущности, которые хранят информацию, доступную для поиска, чтения и замены). То есть ИС описывается как некая абстракция компьютера.

При моделировании сложные ИС разбиваются на составные части, каждая из которых рассматривается отдельно от других. Такой прием называется декомпозицией. Классическим является структурный подход к разработке сложных систем, при котором осуществляется декомпозиция системы по методу «сверху-вниз».

Стремление усовершенствовать структурный подход привело к развитию новых идей, основанных на объектной декомпозиции. Объектно-ориентированная декомпозиция заключается в представлении системы в виде совокупности классов и объектов предметной области. При этом иерархический характер сложной системы отражается в виде иерархии классов, а ее функционирование рассматривается как взаимодействие объектов. При данном подходе сложная система описывается наиболее естественным способом.

Такой подход к разработке программных систем получил название объектно-ориентированный подход. Универсальным языком объектного проектирования является UML (Unified Modeling Language), разработанный на основе популярных объектных методов ОМТ, OOSE и метода Буча.

Существуют немало CASE-средств, поддерживающих язык UML [20].

1.4. Технология Ramus Education

Для решения задач моделирования бизнес-процессов мы будем использовать программный продукт Ramus Education [21].

Ramus Education дает возможность наглядно представить любую деятельность или структуру в виде модели, что позволит оптимизировать работу организации, проверить ее на соответствие стандартам ISO 9000, спроектировать оргструктуру, снизить издержки, исключить ненужные операции, повысить гибкость и эффективность.

Ramus Education поддерживает две методологии (IDEF0, DFD), позволяющие анализировать бизнес с ключевых точек зрения:

- функциональности системы. В рамках методологии IDEF0 бизнес-процесс представляется в виде набора элементов-функций, которые взаимодействуют между собой, а также информационных, людских и производственных ресурсов, потребляемых каждой функцией;

- потоков информации (документооборота) в системе. Диаграммы DFD могут дополнить то, что уже отражено в модели IDEF0, так как они описывают потоки данных, позволяя проследить, каким образом происходит обмен информацией между бизнес-функциями внутри системы.

Ramus Education не поддерживает анализ с точки зрения последовательности выполнения функций. Поэтому предлагается использовать приложение Modelio [13] для создания диаграмм BPMN. Этот метод привлекает внимание к очередности выполнения событий. В него включены элементы логики, что позволяет моделировать и анализировать альтернативные сценарии развития бизнес-процесса.

CASE-средство Ramus Education позволяет ответить на следующие вопросы.

Как работает предприятие? В результате обследования предприятия создается функциональная модель существующей организации работы – «как есть». Эта модель позволяет выяснить, что мы делаем сегодня. Внедрение ИС неизбежно приведет к перестройке существующих бизнес-процессов. Анализ функциональной модели позволяет понять, где находятся наиболее слабые места, в чем будет состоять преимущество новых бизнес-процессов.

Как будет работать предприятие? Найденные в модели «как есть» недостатки можно исправить при создании модели «как будет». При этом, как правило, создается несколько моделей «как будет», из которых по какому-либо критерию выбирается лучшая.

Ramus Education поддерживает формат IDL, который является стандартом для экспорта и импорта моделей IDEF0, что дает возможность разрабатывать модели различными CASE-средствами.

1.5. Горячие слова

Термины, используемые при моделировании бизнес-процессов [3]:

CASE-средство (computer-aided software engineering) – программное средство, поддерживающее процесс создания и сопровождения программного обеспечения, в том числе анализ и формулировку требований, проектирование, генерацию кода, тестирование, документирование, обеспечение соответствия стандартам качества, управление проектом.

Анализ эффективности бизнес-процессов – это анализ результатов выполнения бизнес-процесса и/или параметров, характеризующих выполнение процесса в динамике, и сравнение полученных показателей с затратами (временными, финансовыми, материальными, человеческими), необходимыми для осуществления данного процесса, и/или целевыми показателями эффективности процесса.

Декомпозиция бизнес-процесса – последовательная детализация исходной модели бизнес-процесса до заданного уровня путем создания детальных моделей для каждого объекта процесса.

Деятельность (дело, бизнес) – совокупность процессов, выполняемых (протекающих) последовательно или/и параллельно, преобразующих множество материальных или/и информационных потоков во множество материальных или/и информационных потоков с другими свойствами.

Критерий оценки эффективности бизнес-процесса – качественный или количественный показатель, рассчитываемый по определенной методике и характеризующий результат и/или динамические параметры функционирования бизнес-процесса.

Метод OOSE (object-oriented software engineering) – метод описания информационной системы, предложенный Якобсоном (Ivar Jacobson) и применяющий диаграмму вариантов использования (Use Case Diagram).

Метод Буча (booch method) – объектно-ориентированный метод описания информационной системы, предложенный Бучем (Grady Booch).

Метод ОМТ (object modeling technique) – метод описания информационной системы, предложенный Рамбо (James Rumbaugh), состоящий в построении диаграммы потоков данных для описания выполняемых функций и создания модели классов с целью программной реализации.

Моделирование бизнес-процесса – процесс отражения субъективного видения реально существующего в организации потока работ в виде модели, состоящей из взаимосвязанных объектов.

Модель «как будет» – модель бизнес-процесса, построенная на основе видения перспективной структуры процесса и удовлетворяющая требованиям по целевым значениям показателей эффективности.

Модель «как есть» – модель бизнес-процесса, построенная на основе видения существующего в организации потока работ.

Объект модели – объект в базе данных инструментальной среды моделирования, обладающий рядом атрибутов (параметров), предназначенный для отражения реально существующего объекта определенного типа (пример – функция, выполняемая сотрудником предприятия).

Объект создания бизнес-моделей – реально существующее или проектируемое предприятие.

Объектный подход – набор моделей, связанных с понятием класса/объекта, объединяющего данные (состояние) и поведение.

Оптимизация бизнес-процессов – частичное совершенствование существующих бизнес-процессов за счет ликвидации их явных недостатков (информационные петли, дублирование функций), оформленное соответствующими организационно-распорядительными и нормативными документами.

Организационное развитие – непрерывный процесс реорганизации системы управления, включающий оптимизацию и реинжиниринг бизнес-процессов и процессов управления, совершенствование функциональной и организационных структур и комплекс вопросов управления персоналом.

Проверка адекватности модели бизнес-процесса – совместное изучение модели бизнес-процесса руководителями и специалистами предприятия с целью проверки соответствия модели их видению реально существующего на предприятии потока работ.

Проект реорганизации бизнес-процессов – проект, основная цель которого состоит в построении на предприятии более эффективных бизнес-процессов.

Процесс (бизнес-процесс) – совокупность последовательно или/и параллельно выполняемых операций, преобразующая материальный или/и информационный потоки в соответствующие потоки с другими свойствами.

Реинжиниринг бизнес-процессов (BPR, business process reengineering) – «реинжиниринг – (1) «это фундаментальное переосмысление и радикальное перепроектирование деловых процессов для достижения резких, скачкообразных улучшений в решающих современных показателях деятельности компании – таких, как стоимость, сервис и темпы» (Хэммер и Чемпи); (2) деятельность, основанная на представлении любой организации в виде ряда взаимосвязанных бизнес-процессов и направленная на их анализ и реорганизацию.

Реорганизация бизнес-процесса – целенаправленное изменение бизнес-процесса за счет изменения состава его процедур и/или их параметров, логики процесса, системы принятия решений в рамках процесса, информационного обеспечения.

Структурный подход – раздельное построение модели функций (диаграммы потоков данных) и модели данных (диаграммы «сущность-связь»).

Эффективность бизнес-процесса – отношение конечного результата (выхода) процесса к затраченным на его получение ресурсам. Может измеряться на основе различных критериев.

2-й день. ОСНОВНЫЕ ПОНЯТИЯ IDEF0

Грамматика не предписывает законов языку,
но изъясняет и утверждает его обычаи.

А.С. Пушкин.

Заметки и афоризмы разных годов

2.1. Назначение

Постоянное усложнение производственно-технических и организационно-экономических систем – фирм, предприятий и других субъектов производственно-хозяйственной деятельности, а также необходимость их анализа с целью совершенствования функционирования и повышения эффективности обуславливает необходимость наличия специальных средств описания и анализа таких систем. Эта проблема приобретает особую актуальность с связи с появлением интегрированных компьютеризированных производств и автоматизированных предприятий [1].

Для решения подобных задач моделирования сложных систем существуют хорошо обкатанные методологии и стандарты. К таким стандартам относятся методологии семейства IDEF, позволяющие исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем. С их помощью можно эффективно отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах.

Общая методология IDEF включает частные методологии, основанные на графическом представлении систем: IDEF0, IDEF1, IDEF1X, IDEF2, IDEF3 и др. Подробные спецификации на эти стандарты можно найти на сайте <http://www.idef.com>.

Методология IDEF0 основана на подходе, разработанном Дугласом Россом в начале 70-х годов и получившем название SADT (Structured Analysis & Design Technique, – метод структурного анализа и проектирования) [3]. Основу подхода и, как следствие, методологии IDEF0 составляет графический язык моделирования систем, обладающий следующими свойствами:

- графический язык – полное и выразительное средство, способное наглядно представлять широкий спектр деловых, произ-

водственных и других процессов и операций предприятия на любом уровне детализации;

- язык обеспечивает точное и лаконичное описание моделируемых объектов, удобство использования и интерпретации этого описания;
- язык облегчает взаимодействие и взаимопонимание системных аналитиков, разработчиков и персонала изучаемого объекта (предприятия);
- язык может генерироваться рядом CASE-средств, например, Ramus [21], AI0 WIN (компания KBSI, <http://www.idef.com>).

Эти свойства предопределили выбор методологии IDEF0 в качестве базового средства анализа и синтеза производственно-технических и организационно-экономических систем, что нашло свое отражение в принятии методологии в качестве федерального стандарта США [19]. В 2000 г. методология IDEF0 была принята в качестве стандарта в Российской Федерации [6].

2.2. Концепция

Методология IDEF0 основана на следующих концептуальных положениях [6].

Модель – искусственный объект, представляющий собой отображение системы и ее компонентов. М моделирует А, если М отвечает на вопросы относительно А (М – модель, А – моделируемый объект). Модель разрабатывают для понимания, анализа и принятия решений о реконструкции (реинжиниринге) или проектировании новой системы.

Система представляет собой совокупность взаимосвязанных и взаимодействующих элементов, выполняющих некоторую полезную работу. Элементами системы могут быть любые комбинации разнообразных сущностей, включающих людей, информацию, программное обеспечение, оборудование, изделия, сырье или энергоносители. Модель описывает, что происходит в системе, как ею управляют, какие сущности она преобразует, какие средства использует для выполнения своих функций и что производит.

Основной концептуальный принцип методологии IDEF – представление любой изучаемой системы в виде набора взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции,

действия, происходящие в изучаемой системе. В IDEF0 все, что происходит в системе и ее элементах, принято называть функциями. Каждой функции ставится в соответствие блок. На IDEF0-диаграмме, основном документе при анализе и проектировании систем, блок представляет собой прямоугольник. Связи, посредством которых блок взаимодействует с другими блоками или с внешней по отношению к моделируемой системе средой, представляются стрелками, входящими в блок или выходящими из него. Входящие стрелки показывают, какие условия должны быть одновременно выполнены, чтобы функция, описываемая блоком, осуществилась.

Разработка моделей IDEF0 требует соблюдения ряда строгих формальных правил, обеспечивающих преимущества методологии в отношении однозначности, точности и целостности сложных многоуровневых систем.

Разработка модели в IDEF0 представляет собой итеративную процедуру. На каждом шаге итерации разработчик предлагает вариант модели, который подвергают обсуждению и последующему редактированию, после чего цикл повторяется.

При разработке моделей следует избегать изначальной привязки функций исследуемой системы к существующей организационной структуре моделируемого объекта. Организационная структура должна явиться результатом использования модели. Сравнение результата с существующей структурой позволяет: оценить адекватность модели, предложить решения, направленные на совершенствование этой структуры.

2.3. Компоненты языка

Компоненты синтаксиса языка IDEF0 – блоки, стрелки, диаграммы, правила [1].

Блоки представляют функции, определяемые как деятельность, процесс, операция, действие или преобразование. Стрелки представляют данные или материальные объекты, связанные с функциями. Правила определяют как следует применять компоненты. Диаграммы обеспечивают формат графического и словесного описания модели.

Пример блока показан на рисунке 1. Внутри блока помещается его имя «Почистить картофель» и номер 1.

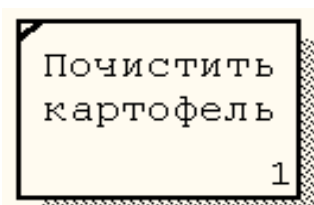


Рисунок 1 – Блок

Стрелка формируется из одного или более отрезков прямых и наконечника на одном конце (рис. 2). Стрелки могут ветвиться или сливаться

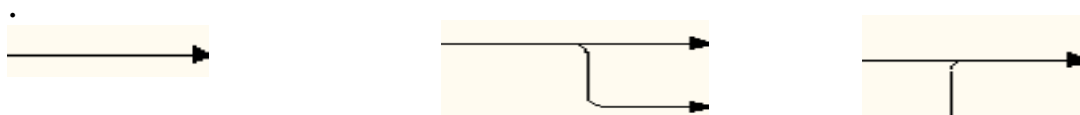


Рисунок 2 – Стрелки

2.4. Имена и метки

Так как IDEF0 – методология функционального моделирования, то имя блока, описывающее функцию, должно быть глаголом или глагольным оборотом. Примеры имен функций: производить детали, наблюдать за выполнением, планировать ресурсы.

Стрелки и их сегменты помечаются существительными или оборотами существительных. Примеры меток стрелок: менеджер, бюджет, конструкция детали.

2.5. Блоки и стрелки

Каждая сторона блока имеет своё определенное значение с точки зрения связи блок-стрелка. Верхняя сторона имеет значение «управление», левая – «вход», правая – «выход», а нижняя – «механизм». В свою очередь, сторона блока, к которой присоединена стрелка, однозначно определяет ее роль.

В IDEF0 различают пять классов стрелок – стрелка входа, стрелка выхода, стрелка управления, стрелка механизма, стрелка вызова.

Стрелка входа – это материал или данные, которые преобразуются или расходуются функцией, чтобы создать то, что появится на ее выходе. Стрелка входа рисуется как входящая в левую грань блока. Допускается, что функция может не иметь ни одной стрелки входа. Часто бывает сложно определить, являются ли данные входом, или управлением. В том случае, когда данные изменяются или перерабатываются, это вход, если нет – управление.

Стрелка управления – это правила, стратегии, процедуры, стандарты, которые определяют условия, необходимые функции, чтобы произвести правильный выход. Стрелка управления рисуется как входящая в верхнюю грань блока. Каждая функция должна иметь хотя бы одну стрелку управления. Управление влияет на функцию, но не преобразуется функцией. Если цель функции – изменить процедуру, то такая процедура будет для функции входом. В случае возникновения неопределенности в классифицировании стрелки (вход или управление) рекомендуется создавать стрелку управления.

Стрелка выхода – это данные или материальные объекты, произведенные функцией. Стрелка выхода рисуется как выходящая из правой грани блока. Каждая функция должна иметь хотя бы одну стрелку выхода. Функция без выхода не имеет смысла и не должна моделироваться.

Стрелка механизма – это ресурсы (персонал, техника, оборудование), поддерживающие выполнение функции. Стрелка механизма рисуется как входящая в нижнюю грань блока. Стрелка механизма может не изображаться на модели.

Стрелка вызова – это стрелка, указывающая на другую модель. Стрелка вызова рисуется как исходящая из нижней грани блока. Такая стрелка используется как указание на то, что некоторая функция выполняется за пределами моделируемой системы.

Стандартное расположение стрелок показано на рисунке 3.

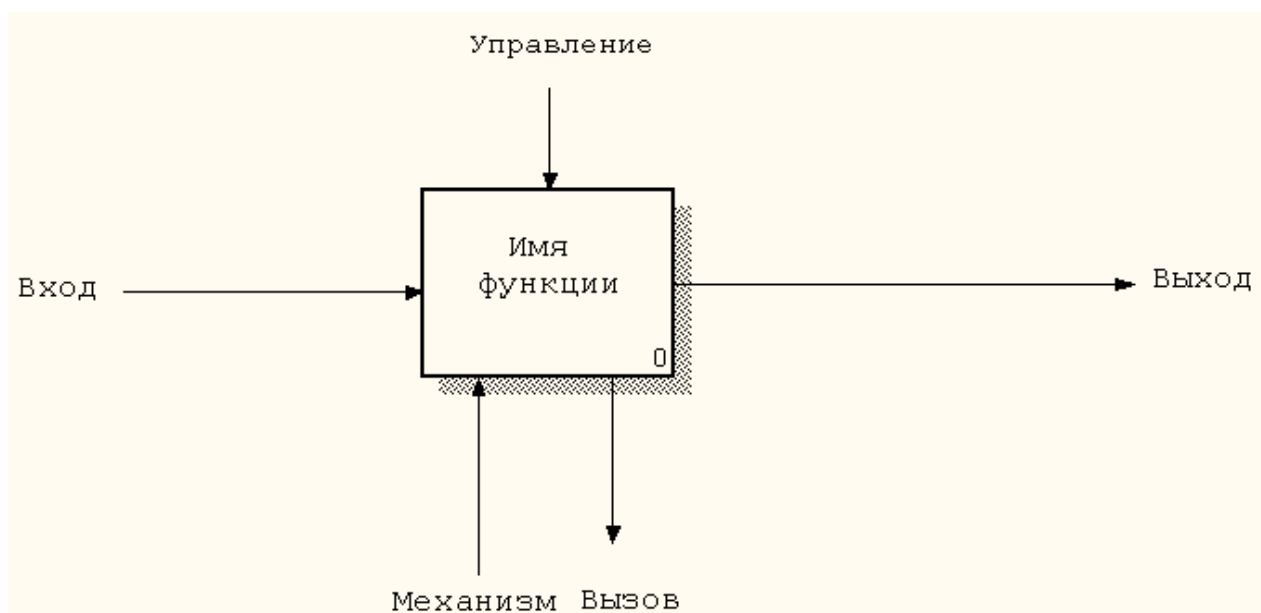


Рисунок 3 – Расположение стрелок блока

2.6. Диаграммы

2.6.1. Типы диаграмм

IDEFO-модели состоят из трех типов документов: графических диаграмм, текста, глоссария. Эти документы имеют перекрестные ссылки друг на друга [6].

Графическая диаграмма – главный компонент модели, содержащий блоки, стрелки, соединения блоков и стрелок. Диаграмма является единицей описания системы и располагается на отдельном листе.

Блоки представляют основные функции моделируемого объекта. Эти функции могут быть декомпозированы на составные части и представлены в виде более подробных диаграмм. Процесс декомпозиции продолжается до тех пор, пока объект не будет описан на уровне детализации, необходимом для достижения целей конкретного проекта. Диаграмма верхнего уровня обеспечивает наиболее общее или абстрактное описание объекта моделирования. За этой диаграммой следует серия дочерних диаграмм, дающих более детальное представление об объекте.

Модель может содержать следующие типы графических диаграмм: контекстная, декомпозиции, дерева узлов, иллюстрации.

2.6.2. Контекстная диаграмма

Каждая модель должна иметь контекстную диаграмму верхнего уровня, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется А-0 (А минус ноль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой. Пример такой диаграммы показан на диаграмме А (приложение 2).

Диаграмма А-0 устанавливает область моделирования и ее границу. Необходимо установить, что входит в систему, а что лежит за ее пределами, то есть определить, что будет рассматриваться как компоненты системы, а что – как внешнее воздействие.

В пояснительном тексте к контекстной диаграмме должна быть указана цель построения диаграммы и зафиксирована точка зрения.

Цель – это краткая формулировка причины создания модели. Цель должна отвечать на вопросы: почему этот процесс должен быть смоделирован, что должна показать модель, что может получить читатель? Примером цели может быть следующее утверждение: «Определить роли и ответственность служащих для написания должностных инструкций».

Точка зрения – это указание на должностное лицо или подразделение организации, с позиции которого разрабатывается модель. Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему. Например, модели одного и того же предприятия с точек зрения главного технолога и финансового директора будут существенно различаться. Это связано с тем, что в конечном итоге финансового директора не интересуют аспекты обработки сырья на производственном оборудовании, а главному технологу ни к чему прорисованные схемы финансовых потоков.

2.6.3. Дочерняя и родительская диаграммы

Единственная функция, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции посредством создания дочерней диаграммы. В свою очередь, каждая из этих подфункций может быть разложена на составные части посредством создания дочерней диаграммы следующего, более низкого уровня. Каждая дочерняя диаграмма содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.

Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский блок, но описывает ее более подробно. Таким образом, дочерняя диаграмма как бы вложена в свой родительский блок. Эта структура представлена на рисунке 4.

Любая обычная диаграмма (не контекстная) может быть как родительской диаграммой (содержать родительские блоки), так и дочерней (подробно описывать собственный родительский блок).

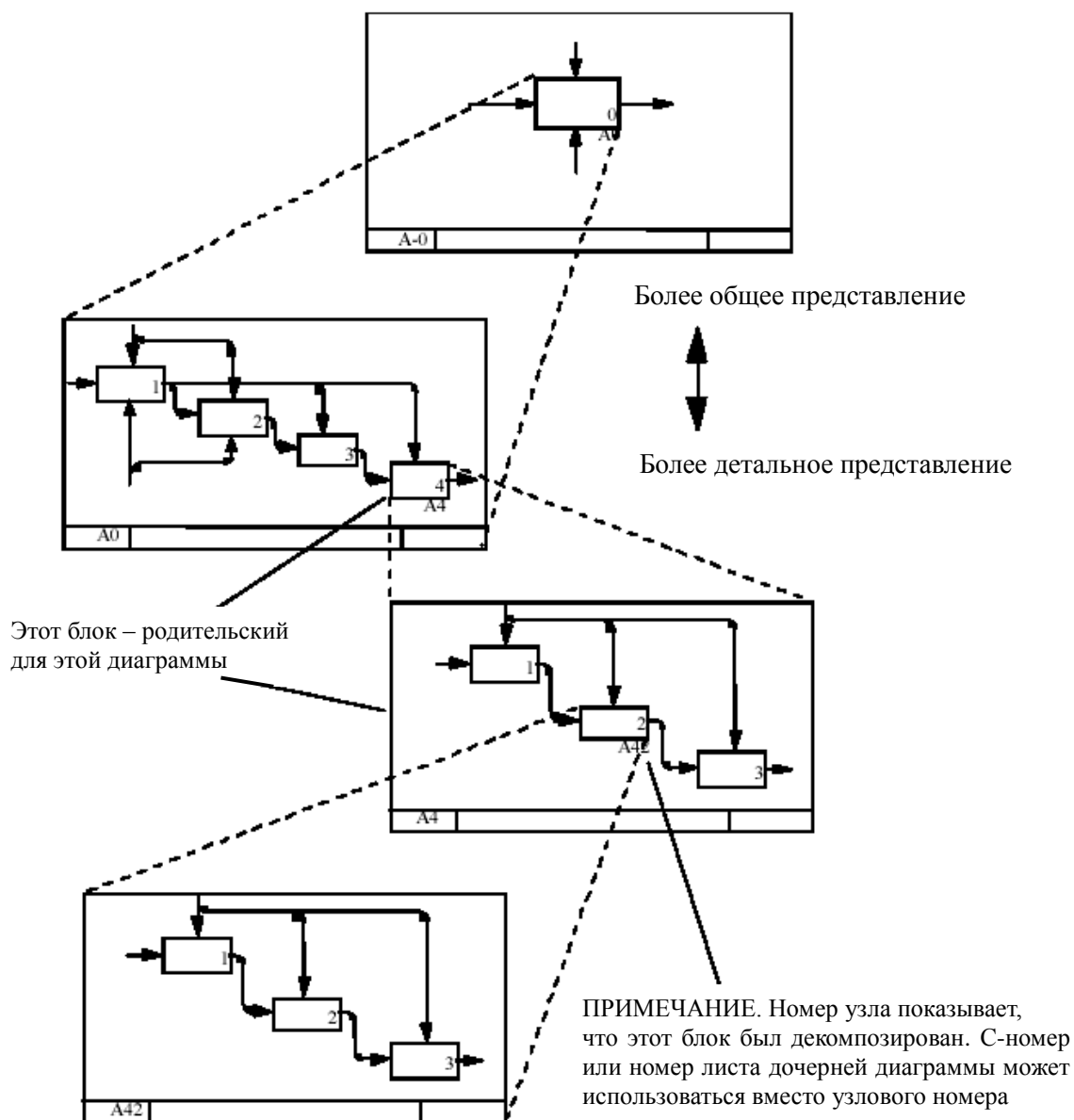


Рисунок 4 – Структура декомпозиции

2.6.4. Текст и глоссарий

Диаграмме может быть поставлен в соответствие текст, представляющий собой краткий комментарий к содержанию диаграммы. Текст не должен использоваться для описания и без того понятных блоков и стрелок на диаграмме.

Глоссарий – это список определений для ключевых слов, фраз, аббревиатур, связанных с блоками, стрелками или с моделью в целом. Глоссарий определяет термины, которые должны одинаково понимать все участники разработки.

2.6.5. Диаграммы иллюстраций и дерева узлов

Диаграммы иллюстрации используются в качестве дополнений, поясняющих специфику содержания основных диаграмм, для иллюстрации альтернативной точки зрения. Такие диаграммы могут не подчиняться правилам IDEF0.

Диаграммы дерева узлов показывают иерархию функций в модели и позволяют рассмотреть всю её целиком. На диаграммах узлов нет стрелок. Диаграмм узлов может быть в модели сколько угодно, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

2.7. Свойства диаграмм

2.7.1. Стрелки-условия

Стрелки на диаграмме, представляя данные или материальные объекты, одновременно задают условия. Входные и управляющие стрелки блока, стрелки блока, соединяющие его с другими блоками или с внешней средой, описывают условия, которые должны быть выполнены для реализации функции, записанной в качестве имени блока.

Например, на рисунке 5 представлен случай, при котором функция 3 может быть выполнена только после получения данных, выработанных функцией 1 и функцией 2.

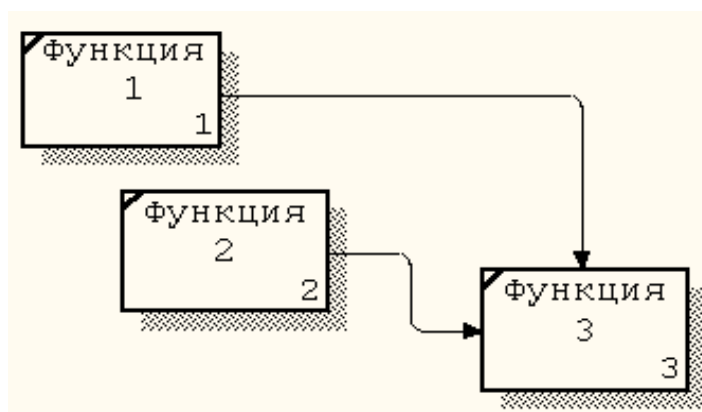


Рисунок 5 – Стрелки-условия

2.7.2. Параллельное функционирование

Различные функции в модели могут быть выполнены параллельно, если выполнены необходимые условия. Например, функция 1 обеспечивает работу функций 2 и 3 (рис. 6).

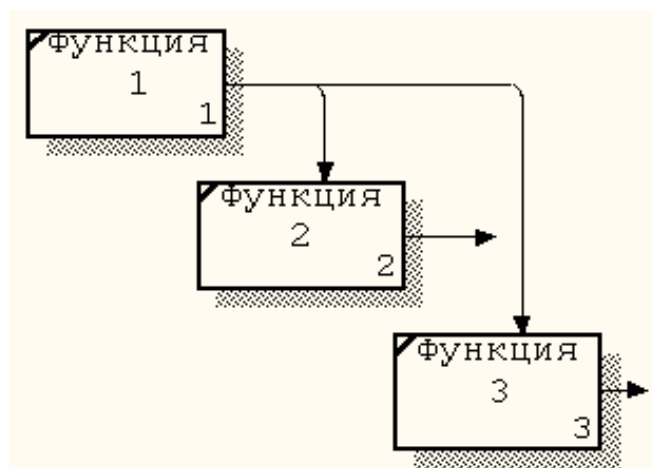


Рисунок 6 – Параллельные функции

2.7.3. Связи блоков на диаграмме

IDEF0 имеет шесть типов связей между блоками в пределах одной диаграммы: доминирование, управление, выход-вход, обратная связь по управлению, обратная связь по входу, выход-механизм.

Доминирование определяется взаимным расположением блоков на диаграмме. Предполагается, что блоки, расположенные на диаграмме выше и левее, влияют на блоки, расположенные ниже и правее. Блоки на диаграммах декомпозиции обычно располагаются по диагонали от левого верхнего угла к правому нижнему. Такой порядок называется порядком доминирования. Поэтому в левом верхнем углу располагается самая важная функция или функция, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже функции.

Остальные пять типов описывают связи между блоками и изображаются стрелками. Для связи блоков между собой используются внутренние стрелки, то есть стрелки, не касающиеся границы диаграммы, которые начинаются у одного и кончаются у другого блока. Такая стрелка отличается от граничной стрелки.

Связь по управлению и связь по входу являются простейшими, поскольку отражают прямые взаимодействия.

Связь по управлению (рис. 7) возникает тогда, когда выход одного блока служит управляющим воздействием на блок с меньшим доминированием. Данные или объекты выхода вышестоящей функции не меняются в нижестоящей.

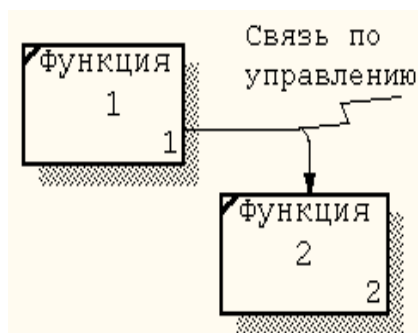


Рисунок 7 – Связь по управлению

Связь выход-вход (рис. 8) возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием.

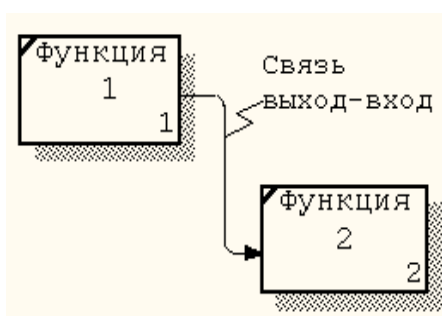


Рисунок 8 – Связь выход-вход

Обратная связь по входу и обратная связь по управлению являются более сложными типами связей, поскольку они представляют итерацию (выход функции влияет на будущее выполнение других функций с большим доминированием, что впоследствии скажется на исходной функции).

Обратная связь по входу (рис. 9) имеет место тогда, когда выход блока становится входом другого блока с большим доминированием. Такая связь, как правило, используется для описания циклов.

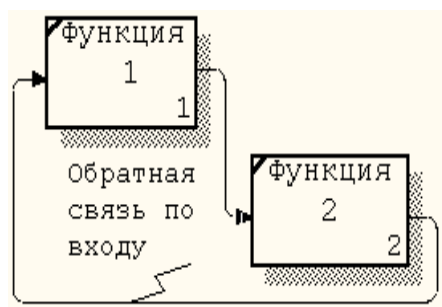


Рисунок 9 – Обратная связь по входу

Обратная связь по управлению возникает в том случае, когда выход одного блока создает управляющее воздействие на блок с большим доминированием (рис. 10). Такая связь часто свидетельствует об эффективности бизнес-процесса.

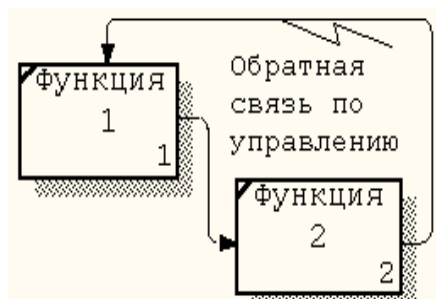


Рисунок 10 – Обратная связь по управлению

Связь выход-механизм (рис. 11) отражает ситуацию, при которой выход одной функции становится средством достижения цели для другой, то есть выход одного блока направляется на механизм другого. Такая связь показывает, что одна функция подготавливает ресурсы, необходимые для другой функции. Связи выход-механизм возникают при отображении в модели процедур выполнения и распределения ресурсов, создания или подготовки средств для выполнения функций системы (например, приобретение или изготовление инструментов и оборудования, обучение персонала, финансирование, закупка материалов).

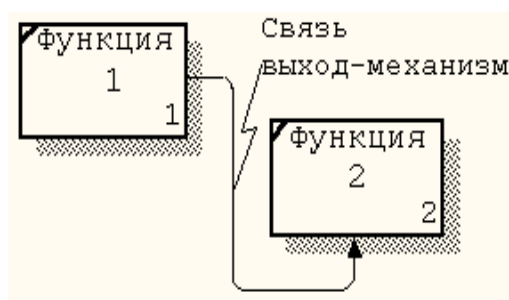


Рисунок 11 – Связь выход-механизм

2.8. Правила построения диаграмм

- В составе модели должна присутствовать контекстная диаграмма А-0, которая содержит только один блок. Номер единственного блока на контекстной диаграмме А-0 должен быть 0.

- Блоки на диаграмме должны располагаться по диагонали – от левого верхнего угла диаграммы до правого нижнего в порядке присвоенных номеров. Блоки на диаграмме, расположенные вверху слева доминируют над блоками, расположенными внизу справа.
- Не контекстные диаграммы должны содержать не менее трёх и не более шести блоков.
- Имена блоков и метки стрелок должны быть уникальными. Если метки стрелок совпадают, это значит, что стрелки отображают тождественные данные.
- Блоки всегда должны иметь хотя бы одну управляющую и одну выходную стрелку, но могут не иметь входных стрелок.
- Если одни и те же данные служат и для управления, и для входа, рисуется только стрелка управления. Этим подчеркивается управляющий характер данных и уменьшается сложность диаграммы.
- Стрелки связываются (сливаются), если они представляют сходные данные и их источник не указан на диаграмме (рис. 12).

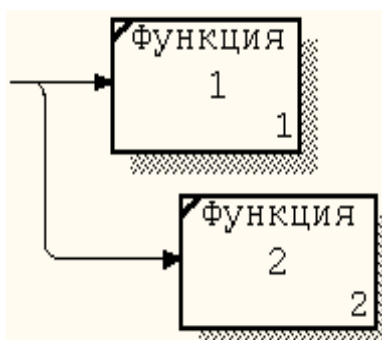


Рисунок 12 – Связанные стрелки

- Циклические обратные связи (рис. 13) для одного и того же блока изображаются только для того, чтобы их выделить. Обычно обратную связь изображают на диаграмме, декомпозирующей блок.

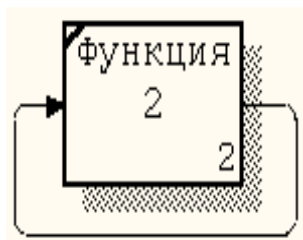


Рисунок 13 – Обратная связь

- Стрелки объединяются, если они имеют общий источник или приемник, или если они представляют связанные данные. Общее название лучше описывает суть данных (рис. 14).

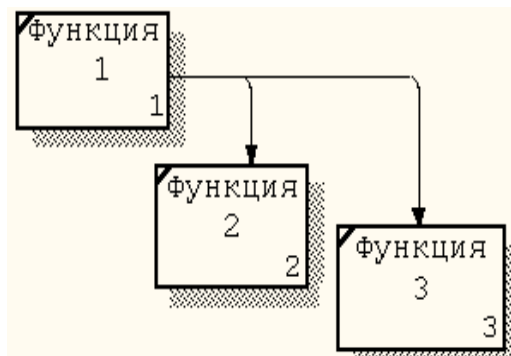


Рисунок 14 – Объединенные стрелки

2.9. Классификация функций

Практика построения моделей требует введения классификации явлений и событий с целью облегчения построения и понимания функциональных моделей. Классификация, предложенная в [6], делит все функции производственно-технических и организационно-экономических систем на четыре основных вида: деятельность, процесс, операция, действие.

Деятельность (дело, бизнес) – это совокупность процессов, выполняемых (протекающих) последовательно или/и параллельно, преобразующих множество материальных или/и информационных потоков во множество материальных или/и информационных потоков с другими свойствами. Деятельность осуществляется в соответствии с заранее определенной и постоянно корректируемой целью, с потреблением финансовых, энергетических, трудовых и материальных ресурсов, при выполнении ограничений со стороны внешней среды.

В модели IDEF0 деятельность описывается блоком А0 на основной контекстной диаграмме А-0.

Процесс (бизнес-процесс) – это совокупность последовательно или/и параллельно выполняемых операций, преобразующая материальный или/и информационный потоки в соответствующие потоки с другими свойствами. Процесс протекает в соответствии с управляющими директивами, вырабатываемыми на основе целей деятельности. В ходе процесса потребляются финансовые, энергетические, трудовые и материальные ресурсы и выполняются ограничения со стороны других процессов и внешней среды.

Операция – это совокупность последовательно или/и параллельно выполняемых действий, преобразующих объекты, входящие в состав материального или/и информационного потока, в соответствующие объекты с другими свойствами. Операция выполняется: (1) в соответствии с директивами, вырабатываемыми на основе директив, определяющих протекание процесса, в состав которого входит операция; (2) с потреблением всех видов потребных ресурсов; (3) с соблюдением ограничений со стороны других операций и внешней среды.

Действие – это преобразование какого-либо свойства материального или информационного объекта в другое свойство. Действие выполняется в соответствии с командой, являющейся частью директивы на выполнение операции, с потреблением необходимых ресурсов и с соблюдением ограничений, налагаемых на осуществление операции.

Введенные выше понятия образуют естественную иерархию блоков на IDEF0-диаграммах при декомпозиции, предусматривая четыре уровня последней.

Уровни декомпозиции, детализирующие действия, естественно считать состоящими из элементарных или простых функций.

2.10. Управление

Один из общих принципов методологии IDEF0 требует: к каждому блоку на диаграмме должна быть присоединена хотя бы одна стрелка управления, отображающая условия правильного функционирования блока. В связи с этим в [6] сформулирован ряд определений, которыми следует руководствоваться при отражении управления на функциональных моделях.

Управление деятельностью – это процесс, состоящий, как минимум, из следующих операций:

- формулирование целей деятельности;
- оценивание ресурсов, необходимых для осуществления деятельности и их сопоставление с имеющимися ресурсами;
- сбор информации об условиях протекания и фактическом состоянии деятельности (глобальная обратная связь);
- выработка и принятие решений, направленных на достижение целей по п.1, в частности, решений о распределении ресурсов по процессам, входящим в состав деятельности;
- оформление решений в виде директив на управление процессами;
- реализация решений (исполнение директив) и оценка их результатов (локальная обратная связь);

- корректировка (в случае необходимости, например, при нехватке ресурсов) ранее сформулированных целей (самонастройка, адаптация).

Именно решения и их реализация суть те стимулирующие воздействия на систему, о которых говорилось выше.

Управление процессом – это операция, состоящая как минимум из следующих действий:

- анализ директивы на управление процессом, ее декомпозиция на директивы управления операциями;
- сбор (по каналам связи) информации о ходе выполнения операций, ее обобщение и формирование сведений о состоянии процесса;
- передача данных в подсистему управления деятельностью;
- сопоставление информации о ходе операций с данными директив и выработка локальных решений, направленных на устранение отклонений;
- корректировка (в случае необходимости) директив на выполнение операций.

Управление операцией – это действие, состоящее в выработке на основании директивы на управление операцией команд на управление действиями, в реализации этих команд, в оценке результатов выполнения, в передаче необходимой информации в комплекс управления процессом, в корректировке команд в случае необходимости.

Блоки управления должны присутствовать на каждой IDEF0-диаграмме (кроме тех, которые являются декомпозициями самих таких блоков). Через них осуществляются управляющие воздействия на остальные блоки диаграммы. Именно эти блоки воспринимают ограничивающую и управляющую информацию и преобразуют ее в соответствующие директивы и команды. Имена блоков управления, как правило, содержат глагол «управлять...».

2.11. Типизация функциональных моделей

Эффективность и производительность труда разработчиков функциональных моделей могут быть повышены за счет применения типовых моделей ориентированных на применение в конкретных предметных областях [6], а также использованием типовых бизнес-процессов.

Перечень типовых бизнес-процессов предприятия по всем видам деятельности представлен в приложении 1. Эта классификация разработана Американским центром производительности и качества (<http://www.apqc.org>).

2.12. Горячие слова

Термины, используемые при функциональном моделировании IDEF0 [1]:

Блок (box) – прямоугольник, содержащий имя и номер и используемый для описания функции.

Ветвление (branch) – разделение стрелки на два или большее число сегментов.

Виды функций – деятельность, процесс (бизнес-процесс), операция, действие.

Внутренняя стрелка (internal arrow) – стрелка входа, управления или выхода, концы которой связывают источник и потребитель, являющиеся блоками одной диаграммы.

Глоссарий (glossary) – список определений для ключевых слов, фраз и аббревиатур, связанных с блоками, стрелками или с моделью в целом.

Граничная стрелка (boundary arrow) – стрелка, один из концов которой связан с источником или потребителем, а другой не присоединен ни к какому блоку на диаграмме. Отображает связь диаграммы с другими блоками системы.

Декомпозиция (decomposition) – разделение моделируемой функции на функции-компоненты; позволяет представить модель в виде иерархической структуры отдельных диаграмм.

Деятельность (дело, бизнес) – совокупность процессов, преобразующих множество материальных или/и информационных потоков во множество материальных или/и информационных потоков с другими свойствами.

Процесс (бизнес-процесс) – совокупность последовательно или/и параллельно выполняемых операций, преобразующая материальный или/и информационный потоки в соответствующие потоки с другими свойствами.

Операция – совокупность последовательно или/и параллельно выполняемых действий, преобразующих объекты, входящие в состав материального или/и информационного потока, в соответствующие объекты с другими свойствами.

Действие – преобразование какого-либо свойства материального или информационного объекта в другое свойство.

Диаграмма (diagram) – часть модели, описывающая декомпозицию блока.

Диаграмма A–0 – специальный вид контекстной диаграммы IDEF0, состоящей из одного блока, описывающего функцию верхнего

уровня, ее входы, выходы, управления и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.

Диаграмма иллюстрация (FEO diagram) – графическое описание, используемое для сообщения специфических фактов о диаграмме IDEF0; при ее построении можно не придерживаться правил IDEF0.

Доминирование – блоки, расположенные на диаграмме выше и левее, влияют на блоки, расположенные ниже и правее.

Дочерний блок (child box) – блок на дочерней диаграмме.

Дочерняя диаграмма (child diagram) – диаграмма, детализирующая родительский блок.

Идея IDEF0 – бизнес-процессы (функции реального объекта бизнеса) представляются как некие преобразования входного потока в выходной под контролем (управлением) управляющего потока с использованием для преобразования механизма.

Имя блока (box name) – глагол или глагольный оборот, помещенный внутри блока и описывающий моделируемую функцию.

Классы стрелок – стрелка входа, стрелка выхода, стрелка управления, стрелка механизма, стрелка вызова.

Код-ICOM – код, обеспечивающий соответствие граничных стрелок дочерней диаграммы со стрелками родительского блока; используется для ссылок.

Контекст (context) – окружающая среда, в которой действует функция (или комплект функций на диаграмме).

Контекстная диаграмма (context diagram) – диаграмма, имеющая узловой номер A–n (n0), которая представляет контекст модели и располагается на отдельном листе. Диаграмма A–0, состоящая из одного блока, является обязательной контекстной диаграммой; диаграммы с узловыми номерами A–1, A–2,... – это дополнительные контекстные диаграммы.

Метка стрелки (arrow label) – существительное или оборот существительного, связанные со стрелкой или сегментом стрелки и определяющие их значение.

Модель IDEF0 (IDEF0 model) – графическое описание системы, разработанное с определенной целью и с выбранной точки зрения; комплект одной или более диаграмм IDEF0, которые изображают функции системы с помощью графики, текста и глоссария.

Назначение IDEF0 – перестройка структуры функций, которая позволит повысить производительность и эффективность системы.

Номер блока (box number) – число (0–6), помещаемое в правом нижнем углу блока и однозначно идентифицирующее блок на диаграмме.

Обратная связь по входу – выход блока становится входом другого блока с большим доминированием.

Обратная связь по управлению – выход одного блока создает управляющее воздействие на блок с большим доминированием.

Основной принцип IDEF0 – представление системы в виде набора взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции, действия, происходящие в системе.

Основные понятия IDEF0 – блок, стрелка, декомпозиция, глоссарий.

Родительская диаграмма (parent diagram) – диаграмма, которая содержит родительский блок.

Родительский блок (parent box) – блок, который подробно описывается дочерней диаграммой.

С-номер (c-number) – номер, создаваемый в хронологическом порядке и используемый для идентификации диаграммы и прослеживания ее истории.

Связывание/развязывание (bundling/unbundling) – объединение значений стрелок в составное значение (связывание в пучок) или разделение значений стрелок (развязывание пучка), выраженное синтаксисом слияния или ветвления стрелок.

Связь выход-вход – возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием.

Связь выход-механизм – выход одного блока направляется на механизм другого.

Связь по управлению – выход одного блока служит управляющим воздействием на блок с меньшим доминированием.

Сегмент стрелки (ветвь) – сегмент линии, который начинается или заканчивается на стороне блока, в точке ветвления или слияния, или на границе.

Слияние (join) – объединение двух или большего числа сегментов стрелок в один сегмент.

Стрелка (arrow) – направленная линия, состоящая из одного или нескольких сегментов, которая моделирует открытый канал или канал, передающий данные или материальные объекты от источника (начальная точка стрелки) к потребителю (точка с наконечником), то есть описывает данные, неформальное управление или что-либо другое, «оказывающее влияние» на функцию, изображенную блоком.

Стрелка входа (input arrow) – класс стрелок, которые отображают вход IDEF0-блока, то есть данные или материальные объекты, которые преобразуются функцией в выход; связываются с левой стороной блока IDEF0.

Стрелка вызова (call arrow) – вид стрелки механизма, который обозначает обращение из блока данной модели к блоку другой модели и обеспечивает связь между моделями.

Стрелка выхода (output arrow) – класс стрелок, которые отображают выход IDEF0-блока, т. е. данные или материальные объекты, произведенные функцией в выход; связываются с правой стороной блока IDEF0.

Стрелка механизма (mechanism arrow) – класс стрелок, которые отображают механизмы IDEF0 (т. е. средства, используемые для выполнения функции, персонал, станки); связываются с нижней стороной блока IDEF0.

Стрелка управления (control arrow) – класс стрелок, которые в IDEF0 отображают управление (т. е. условия, при выполнении которых выход блока будет правильным, например, стратегии, стандарты); связываются с верхней стороной блока IDEF0.

Текст (text) – любой текстовый комментарий к графической модели IDEF0.

Тильда (squiggle) – небольшая ломаная линия, используемая для соединения метки с конкретным сегментом стрелки или примечания модели – с компонентом диаграммы.

Типы связей между блоками – доминирование, управление, выход-вход, обратная связь по управлению, обратная связь по входу, выход-механизм.

Точка зрения (viewpoint) – указание на должностное лицо или подразделение организации, с позиции которого разрабатывается модель.

Тоннельная стрелка (tunneled arrow) – стрелка, не удовлетворяющая обычному требованию, согласно которому каждая стрелка на дочерней диаграмме должна соответствовать стрелкам на родительской диаграмме.

Функция (activity) – деятельность, процесс или преобразование (моделируемые блоком IDEF0), идентифицируемое глаголом.

Цель (purpose) – краткая формулировка причины создания модели.

3-й день. УСТАНОВКА JDK, Ramus Education, Modelio

Поздно воротился он в смиренный свой уголок; долго не мог заснуть, и, когда сон им овладел, ему пригрезились карты, зеленый стол, кипы ассигнаций и груды червонцев.

А.С. Пушкин. Пиковая дама, 1

3.1. Установка JDK

Java – язык программирования и платформа вычислений, которые были впервые выпущены компанией Sun Microsystems в 1995 г (в последующем приобретённой компанией Oracle). Компания предоставляет на своем сайте для свободного доступа спецификацию языка (описывающая лексику, типы данных, основные конструкции) и инструментальные средства разработки приложений – Java Development Kit (JDK).

Главная задача, которую преследовали разработчики – создание надежного платформи-независимого объектного языка, который позволял бы разрабатывать небольшие мобильные приложения для web – апплеты. Технология разработки и использования java-апплетов основана на двух стандартизованных компанией компонентах: на мобильных Java-байт кодах – формате представления результатов компиляции исходного программного кода java-апплета – и на спецификации виртуальной машины Java (JVM) – интерпретаторе мобильных java-байт кодов. Скомпилированные апплеты должны храниться на web-сервере. Их вызов на машину клиента обеспечивается браузером при просмотре HTML-страницы, в которой с помощью специальных тегов встроен вызов апплета с передачей ему фактических параметров. После вызова мобильных java-байт кодов на сторону web-клиента их исполнение осуществляется JVM, встроенной в браузер.

Наряду с использованием java для создания апплетов широко используются системы программирования с входным языком java, основанные на традиционной технологии.

В последние годы интерес к Java был обусловлен появлением таких технологий, как Java EE, включая JSP (Java Server Pages), Java ME (подмножество платформы Java для устройств, ограниченных в ресурсах), сделавших Java наиболее популярной платформой для создания корпоративных решений, поддерживаемой почти всеми произ-

водителями ПО. Основная сфера применения Java – это приложения масштаба предприятия и многозвенные распределенные системы, базирующиеся на Java EE-совместимых серверах приложений.

Внутри Java существуют несколько основных семейств технологий:

- Java SE – Java Standard Edition, основное издание Java, содержит компилятор, API, Java Runtime Environment; подходит для создания пользовательских приложений, в первую очередь – для настольных систем.

- Java EE – Java Enterprise Edition, представляет собой набор спецификаций для создания программного обеспечения уровня предприятия.

- Java ME – Java Micro Edition, создана для использования в устройствах, ограниченных по вычислительной мощности, например, в мобильных телефонах, КПК, встроенных системах;

- Java FX – технология, являющаяся следующим шагом в эволюции Java как Rich Client Platform; предназначена для создания графических интерфейсов корпоративных приложений и бизнеса.

JRE (среда выполнения для Java) – минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора и других средств разработки. Если пользователь хочет только запускать программы, это именно то, что ему нужно.

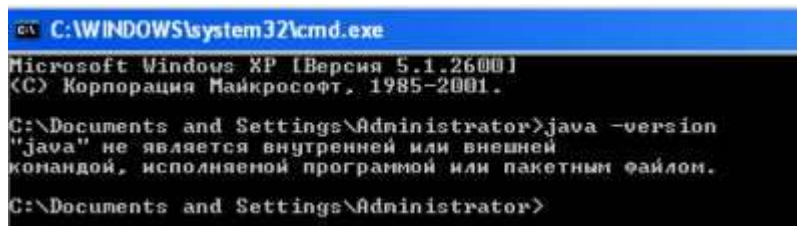
Java Development Kit (сокращенно JDK) – бесплатно распространяемый компанией Oracle комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и JRE.

JDK долгое время было базовым средством разработки приложений. Оно не содержит никаких текстовых редакторов, а оперирует только уже существующими java-файлами. Компилятор представлен утилитой javac (java compiler). Виртуальная машина реализована программой java. Для тестовых запусков апплетов существует специальная утилита appletviewer. Наконец, для автоматической генерации документации на основе исходного кода прилагается средство javadoc.

Все современные интегрированные среды разработки приложений на Java, такие, как NetBeans IDE, IntelliJ IDEA, Eclipse опираются на сервисы, предоставляемые JDK.

Для Ramus Education установим JDK 6 [15].

1. Запустите Java и узнайте ее версию: выберите **Пуск, Программы, Выполнить**, введите команду **cmd**, в открывшейся консоли введите команду **java -version**.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\Administrator>java -version
"java" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Documents and Settings\Administrator>
```

Рисунок 15 – Версия Java

Если сообщение как на рис. 15, то необходимо установить java. Если сообщение примерно как на рис. 16, то сразу переходим к установке Ramus.



```
Command Prompt
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\Administrator>java -version
java version "1.6.0_20-ea"
Java(TM) SE Runtime Environment (build 1.6.0_20-ea-b02)
Java HotSpot(TM) Client VM (build 17.0-b12, mixed mode, sharing)

C:\Documents and Settings\Administrator>
```

Рисунок 16 – Версия Java

2. Запустите установку 32-разрядной JDK: выберите **jdk-6u37-windows-i586.exe, M2**.

3. Разработчики программы приветствуют вас, выберите **Next**.



Рисунок 17 – License Agreement

4. В окне **License Agreement** (рис. 17) предлагается ознакомиться с условиями лицензионного соглашения, выберите **Accept**.

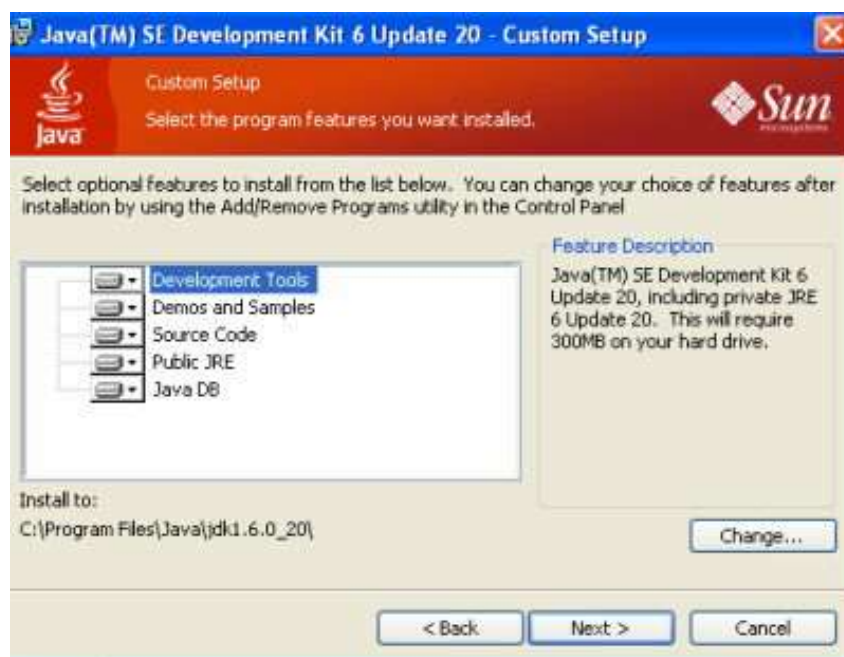


Рисунок 18 – Custom Setup

5. В окне **Custom Setup** (рис. 18) предлагается выбрать программы для установки, выберите по умолчанию, то есть все, обратите внимание на каталог **C:\Program Files\Java\jdk1.6.0_37** где будет размещаться приложение, нажмите **Next**. Начинается установка (рис. 19).

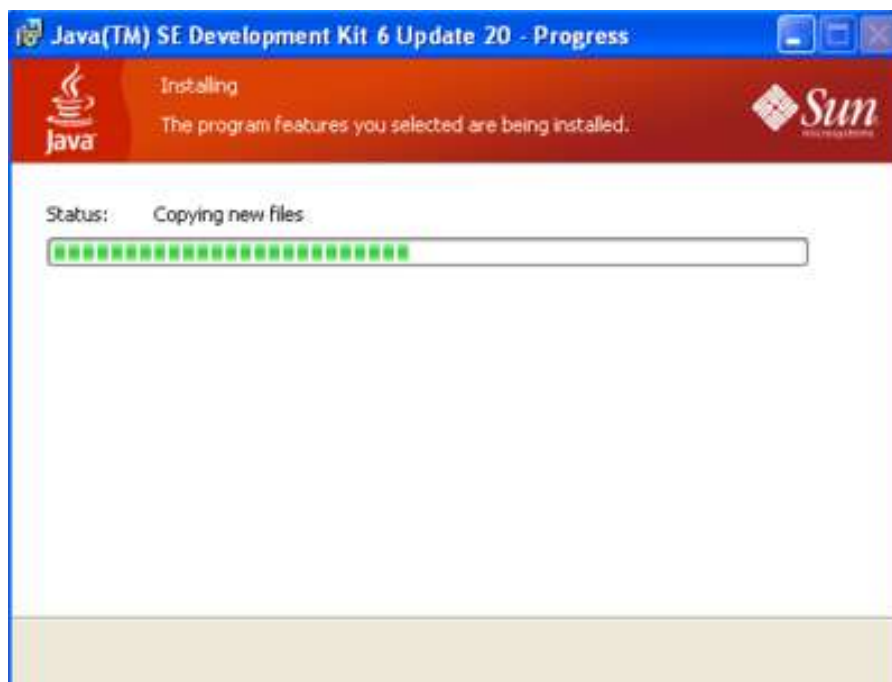


Рисунок 19 – Установка

6. После завершения установки (рис 20): выберите **Finish**.

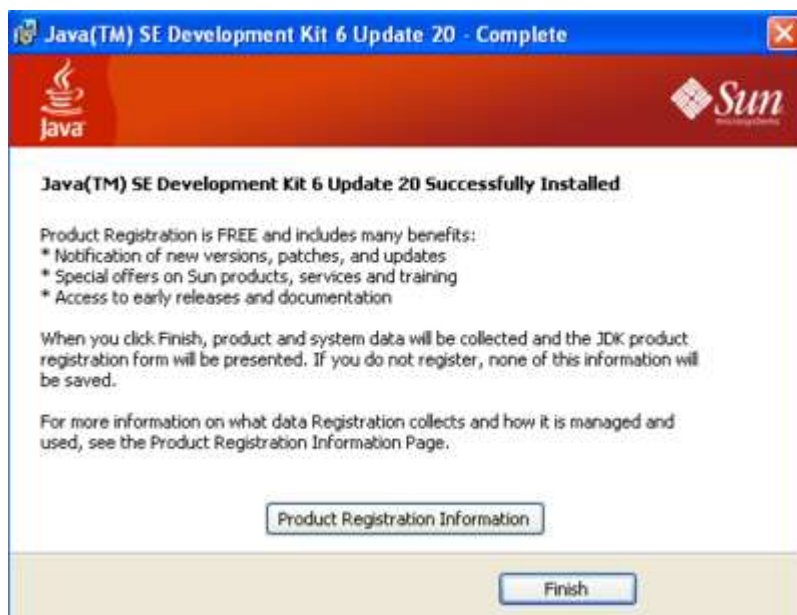


Рисунок 20 – Complete

7. Войдите в каталог **jdk1.6.0_37**, там должны быть следующие папки: **bin** (здесь находятся основные утилиты, например компилятор и JVM); **demo** (примеры программ); **jre** (файлы, имеющие отношение непосредственно к JRE); **lib** (библиотеки JDK). Кроме того, здесь также должен быть файл **src.zip**. В нем находятся исходные кода библиотек, которые вы при желании можете изучить самостоятельно.

8. Перезагрузите компьютер.

9. Запустите **Java** и узнайте ее версию: выберите **Пуск, Программы, Выполнить**, наберите **cmd**, в открывшейся консоли введите команду: **java-version**.

Если результат примерно такой (рис. 21), то можно себя поздравить – все установлено правильно.

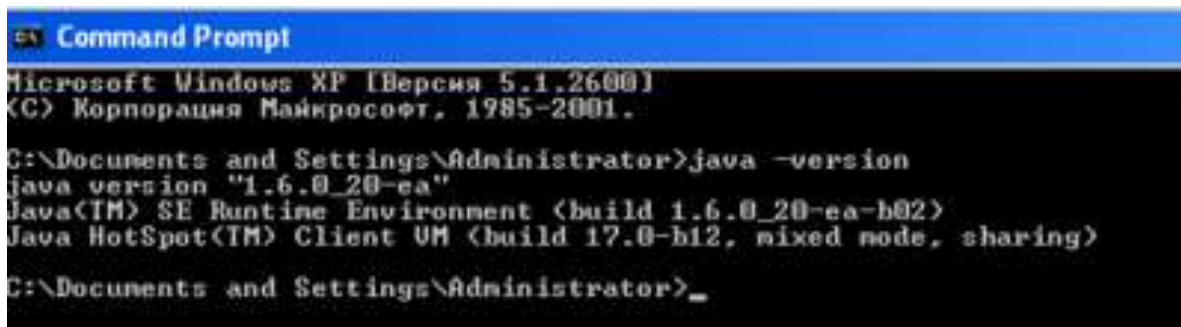


Рисунок 21 – Версия Java

3.2. Установка JDK (Linux)

Для загрузки программы Java следует открыть следующую страницу [15] и сохранить `jdk-6u6-linux-i586-rpm.bin` на диск.

10. Скопируйте **jdk-6u6-linux-i586-rpm.bin** в каталог **/home/student/Загрузка**

11. Запустите консоль: **Система, Терминалы, Консоль.**

12. Войдите почти как администратор: введите команду **sudo su**

13. Смените текущий каталог: введите команду

cd /home/student/Загрузка

14. Проверьте, где вы: введите команду **ls**.

15. Set the executable permissions:

```
chmod a+x jdk-6u6-linux-i586-rpm.bin
```

Если команда выполнена, то сообщений не должно быть. Если ошибка, то читайте и заново выполняйте команду.

16. Запустите установку: **./jdk-6u6-linux-i586-rpm.bin**

17. Прочитайте лицензионное соглашение нажимая **Enter**, выберите **Yes**, наблюдайте процесс установки.

18. Проверьте работу java. Узнайте ее версию: введите

java -version. Если результат есть, то установка прошла успешно.

19. Запустите java-программу: введите три команды

```
cd /usr/java/jdk1.6.0_06/demo/jfc/SwingSet2
```

```
chmod +x SwingSet2.jar
```

```
./SwingSet2.jar
```

Наблюдайте красоту.

Устанавливаем `jdk-6u20-linux-i586-rpm.bin` по инструкции [19].

3.3. Установка Ramus Education

Версии ОС: Windows 2000, XP, Vista, 7, 2003 Server, 2008 Server.

Загружаем Ramus Educational со страницы [20].

20. Запустите установку: выберите **ramus-educational-1.1.1-setup.exe, M2**.

21. Разработчики программы приветствуют вас (рис. 22), выберите **Далее**.



Рисунок 22 – Установка

22. В окне License Agreement (рис. 23) предлагается ознакомиться с условиями лицензионного соглашения, выберите **Принимаю**.

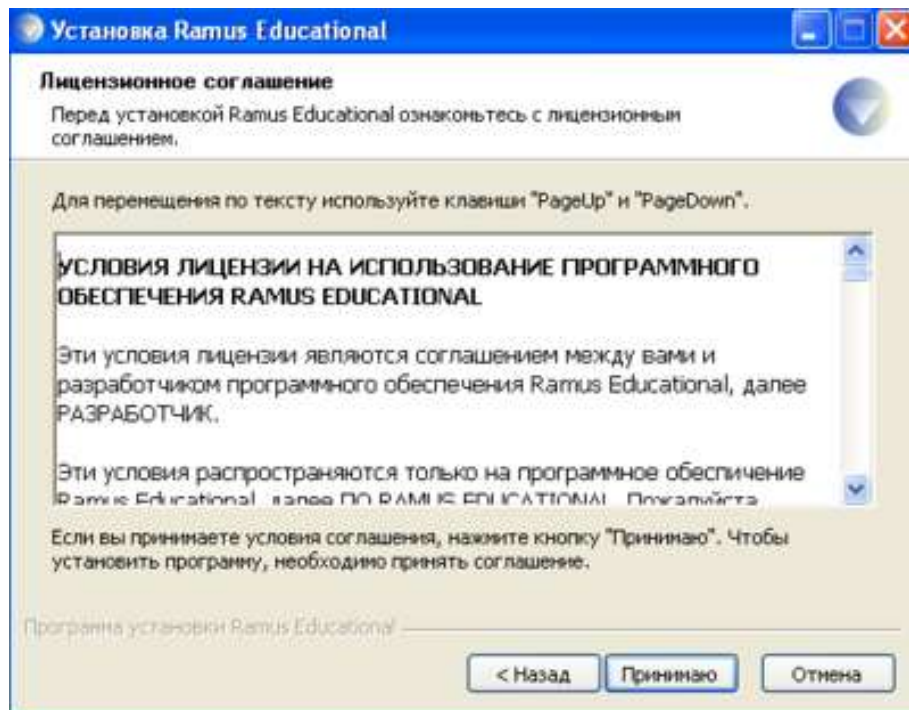


Рисунок 23 – License Agreement

23. В окне **Выбор папки установки** (рис. 24) выберите по умолчанию, нажмите **Далее**.

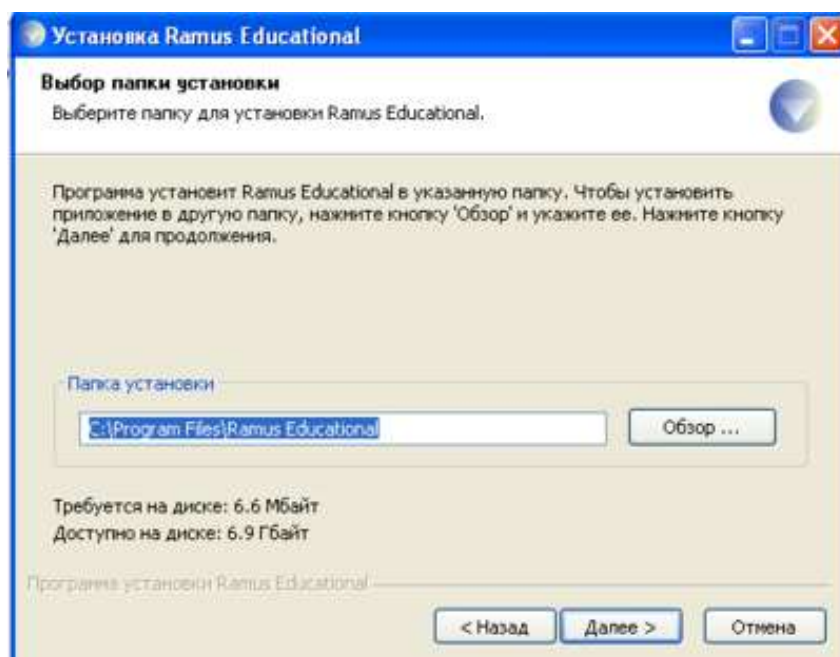


Рисунок 24 – Выбор папки установки

24. В окне **Папка в меню Пуск** (рис. 25) выберите по умолчанию, нажмите **Установить**.

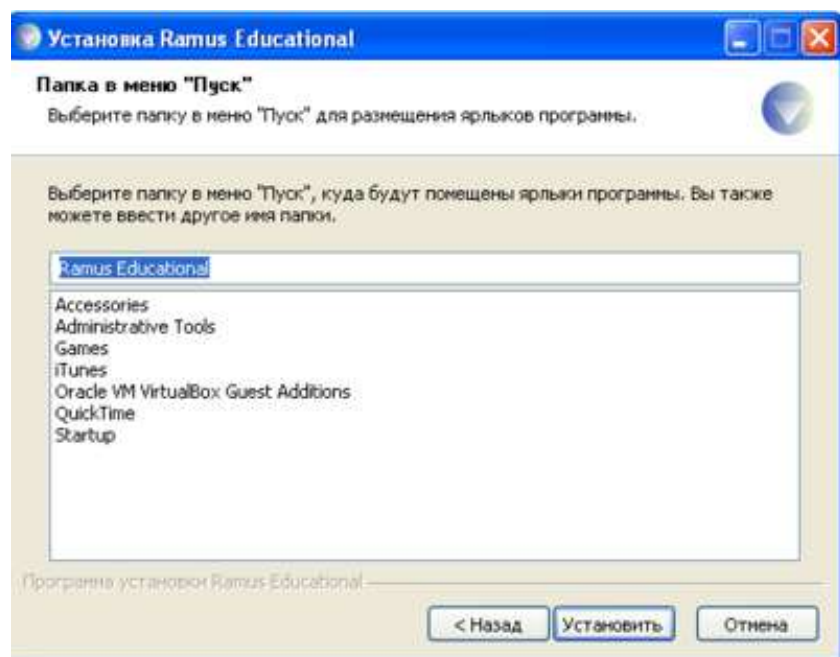


Рисунок 25 – Папка в меню Пуск

25. После окончания установки нажмите **Далее**, затем **Готово** (рис. 26).



Рисунок 26 – Финиш

26. Если установка прервется и сообщение будет похожее на сообщение на рисунке 27, то необходимо установить java.



Рисунок 27 – Ошибка

3.4. Установка Ramus Education (Linux)

Версии ОС: Любая ОС, поддерживающая Java версии 1.6 (Linux, Solaris).

Загружаем Ramus Educational со страницы [21].

27. Откройте консоль и выполните команду:

java -jar ramus-educational-1.1.1-install.jar

3.5. Установка JDK 8

В требованиях к установке Modelio [21] указано, что для Modelio 3.3 необходима Java 8. Загрузите последнюю версию комплекта разработки JDK [15].

28. Запустите установку JDK: выберите **jdk-8u5-windows-x64.exe**, **M2** и далее см. раздел 3.1.

3.6. Установка переменных среды JAVA_HOME и PATH

Необходимо прописать путь до файлов JDK в путях операционной системы. Это позволит нам запускать основные файлы из командной строки.

29. Установите переменную JAVA_HOME: в меню **Пуск** выберите **Компьютер**, нажмите **МП**, в появившемся меню выберите пункт **Свойства**, **Дополнительные параметры системы**, выберите **Переменные среды**, нажмите кнопку **Создать** (рис. 28).

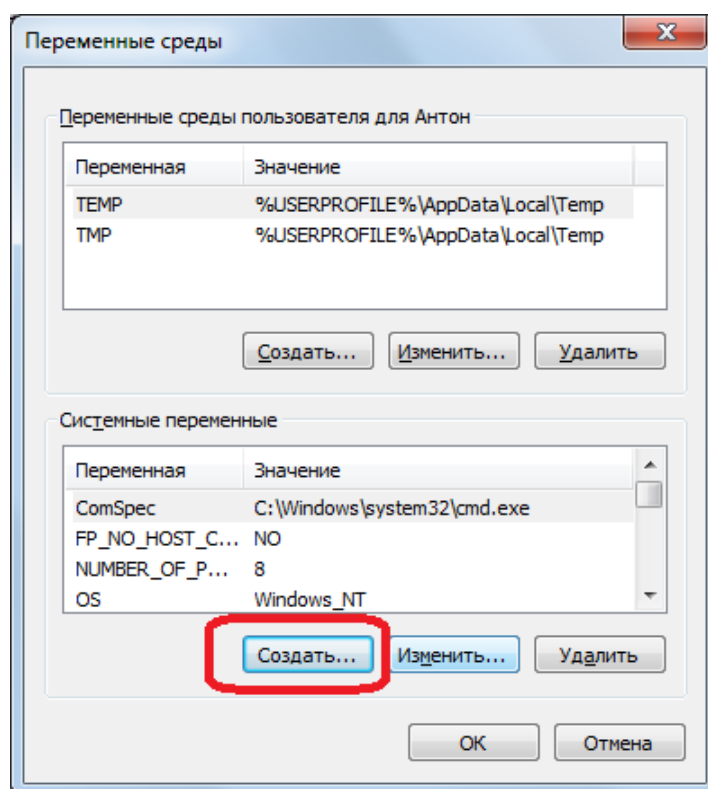


Рисунок 28 – Переменные среды

30. Введите значение переменной JAVA_HOME (рис. 29).

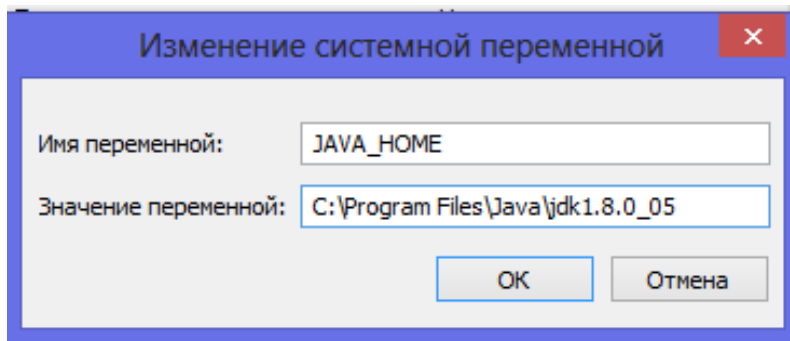


Рисунок 29 – JAVA_HOME

31. Для установки переменной Path скорее всего ее создавать не надо – она уже обычно есть. Так что надо ее найти в списке и нажать кнопку **Изменить**. Но будьте внимательны. Во-первых, здесь надо вводить строку предварительно поставив “;” в конце списка всех путей. И, во-вторых, надо добавить строку “\bin” после. Пример показан на рисунке 30.

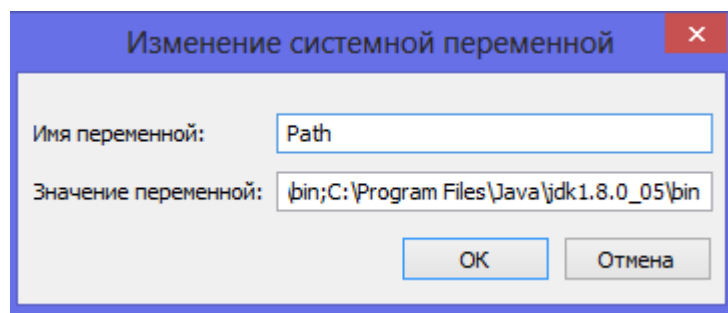


Рисунок 30 – Path

3.7. Hello World!

32. Создайте исходный файл с программой на языке Java: выберите Пуск, Программы, Блокнот, введите следующий код

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

33. Сохраните файл: выберите **Файл, Сохранить как**, выберите папку **myapplication** на диске C, в поле **Имя файла** введите **HelloWorldApp.java**, в поле **Тип файла** выберите **Текстовые документы (*.txt)**, в поле **Кодировка** выберите **ANSI**, нажмите **Сохранить**.

34. Откройте каталог: выберите **Пуск, Программы, Выполнить**, введите команду **cmd**, в открывшейся консоли введите команду **cd C:\myapplication**

35. Просмотрите каталог: введите в консоли команду (рис. 31)
C:\myapplication>dir

```
Microsoft Windows [Version 6.2.9200]
(c) Корпорация Майкрософт, 2012. Все права защищены.

C:\windows\system32>cd C:\myapplication

C:\myapplication>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 937A-6EF0

Содержимое папки C:\myapplication

12.07.2015  20:34    <DIR>          .
12.07.2015  20:34    <DIR>          ..
12.07.2015  20:36                105 HelloWorldApp.java
                1 файлов          105 байт
                2 папок   48 466 026 496 байт свободно

C:\myapplication>
```

Рисунок 31 – Каталог

36. Выполните компиляцию исходного файла: введите в консоли команду

javac HelloWorldApp.java

При появлении ошибок исправляем (рис. 32). И снова компилируем.

37. Просмотрите каталог: введите в консоли команду (рис. 32)

C:\myapplication>dir

```
C:\myapplication>javac HelloWorldApp.java
HelloWorldApp.java:2: error: <identifier> expected
    public static void<String[] args>{
                        ^
1 error

C:\myapplication>javac HelloWorldApp.java

C:\myapplication>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 937A-6EF0

Содержимое папки C:\myapplication

12.07.2015  20:41    <DIR>          .
12.07.2015  20:41    <DIR>          ..
12.07.2015  20:41                432 HelloWorldApp.class
12.07.2015  20:41                111 HelloWorldApp.java
                2 файлов          543 байт
                2 папок   48 465 989 632 байт свободно
```

Рисунок 32 – Каталог

38. Запускаем программу: введите в консоли команду
java -cp . HelloWorldApp

3.8. Установка JDK 8 (Linux)

Для загрузки программы Java следует открыть следующую страницу [14]. Выберите rpm пакет (jdk-8u45-linux-i586.rpm, jdk-8u45-linux-x64.rpm, jre-8u45-linux-i586.rpm or jre-8u45-linux-x64.rpm).

Сохраните jdk-8u45-linux-x64.rpm на диск.

Устанавливаем jdk-8u45-linux-x64.rpm по инструкции [18], [16].

39. Скопируйте **jdk-8u45-linux-x64.rpm** в каталог

/home/student/Загрузка

40. Запустите консоль: выберите **Консоль**.

41. Войдите почти как администратор: введите команду **sudo su**

42. Смените текущий каталог: введите команду

cd /home/student/Загрузка

43. Проверьте, где вы: введите команду **ls**

44. Запустите установку: в консоли введите команду

urpmi jdk-8u45-linux-x64.rpm

45. Определите ссылки на java, javaws, libjavaplugin.so (для Firefox/Mozilla) и javac: в консоли введите команды

```
## java ##
```

```
alternatives --install /usr/bin/java java
```

```
/usr/java/jdk1.8.0_45/jre/bin/java 200000
```

```
## javaws ##
```

```
alternatives --install /usr/bin/javaws javaws
```

```
/usr/java/jdk1.8.0_45/jre/bin/javaws 200000
```

```
## Java Browser (Mozilla) Plugin 32-bit ##
```

```
alternatives --install /usr/lib/mozilla/plugins/libjavaplugin.so libjavaplugin.so /usr/java/jdk1.8.0_45/jre/lib/i386/libnpjp2.so 200000
```

```
## Java Browser (Mozilla) Plugin 64-bit ##
```

```
alternatives --install /usr/lib64/mozilla/plugins/libjavaplugin.so
```

```
libjavaplugin.so.x86_64
```

```
/usr/java/jdk1.8.0_45/jre/lib/amd64/libnpjp2.so 200000
```

```
## javac only if you installed JDK (Java Development Kit) package
```

```
##
```

```
alternatives --install /usr/bin/javac javac  
/usr/java/jdk1.8.0_45/bin/javac 200000
```

```
alternatives --install /usr/bin/jar jar /usr/java/jdk1.8.0_45/bin/jar  
200000
```

46. Просмотрите переключение между OpenJDK и Sun/Oracle Java JDK/JRE 6, 7, 8 (+ отмечена текущая программа): в консоли введите команду

```
alternatives --config java
```

There are 4 programs which provide 'java'.

```
Selection  Command
```

```
1          /usr/java/jdk1.6.0_26/jre/bin/java  
2          /usr/lib/jvm/jre-1.6.0-openjdk/bin/java  
3          /usr/java/jdk1.7.0_55/jre/bin/java  
*+ 4       /usr/java/jdk1.8.0_45/jre/bin/java
```

Enter to keep the current selection[+], or type selection number: 4

```
alternatives --config javaws
```

```
alternatives --config libjavaplugin.so
```

```
alternatives --config libjavaplugin.so.x86_64
```

```
alternatives --config javac
```

47. Добавляем переменную JAVA_HOME в файл /etc/profile или \$HOME/.bash_profile

```
## export JAVA_HOME JDK ##
```

```
export JAVA_HOME="/usr/java/jdk1.8.0_45"
```

48. Проверьте текущие версии java, javac, javaws, libjavaplugin.so: в консоли введите команды

```
java -version
```

```
java version "1.8.0_45"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_45-b26)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.40-b25, mixed mode)
```

```
javaws
```

```
Java(TM) Web Start 11.40.2.26-fcs
```

```
[...]
```

```
javac -version
```

```
javac 1.8.0_45
```

49. Проверьте libjavaplugin.so с помощью перезагрузки Mozilla Firefox, выберите *Plugins*.

3.9. Установка Modelio

Modelio – среда объектно-ориентированного проектирования на языке UML и BPMN.

Для установки следует загрузить архив со страницы загрузки [12] и разархивировать его.

Следует убедиться, что в переменной среды Path прописан путь к файлу java.exe.

Запуск может быть неудачен, если в качестве рабочей директории используется папка, в которую нет права записи у текущего пользователя. Следует запускать среду, используя директорию, на которую достаточно прав.

50. Запустите Modelio: в папке Modelio 3.3 выберите файл modelio.exe, M2.

4-й день. СОЗДАНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ДЕЯТЕЛЬНОСТИ КОМПАНИИ ПО ПРОИЗВОДСТВУ ПЕЛЬМЕНЕЙ

Чтоб уха была по сердцу,
Можно будет в кипяток
Положить немного перцу,
Луку маленький кусок.

А. С. Пушкин. Из письма к Соболевскому

4.1. Пользовательский интерфейс









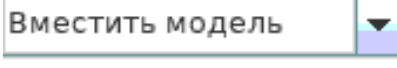
Интерфейс Ramus Education простой и интуитивно понятный. Ниже представлено описание интерфейса версии Ramus Education 1.1.1 на основе встроенной в программу справки [21].

4.1.1. Панель инструментов

Панель инструментов обеспечивает быстрый доступ к часто выполняемым задачам.











В таблице 1 показаны элементы стандартной панели.

Таблица 1 – Описание элементов управления стандартной панели инструментов Ramus Education

Элемент управления	Описание
	Новый проект
	Открыть проект
	Сохранить проект
	Отменить последнее действие
	Повторить последнее действие
	Параметры страницы
	Предварительный просмотр
	Печать
	Выбор масштаба

Палитра инструментов содержит инструменты для работы с объектами в диаграмме Ramus Education. Вид палитры зависит от выбранной нотации (IDEF0, DFD). В таблице 2 показаны основные элементы палитры инструментов.

Таблица 2 – Описание элементов управления палитры инструментов Ramus Education

Элемент управления	Описание
	Режим курсора
	Режим добавления функциональных блоков
	Режим работы со стрелками
	Режим размещения тильд
	Режим добавления внешних ссылок
	Режим добавления хранилищ данных
	Режим добавления текста
	Сетка
	Перейти к родительской диаграмме
	Перейти к дочерней диаграмме

- «Режим курсора» – основной режим при работе с диаграммами. Режим позволяет манипулировать объектами диаграмм: перетягивать, менять размеры, отсоединять стрелки от краёв диаграммы или функционального блока и присоединять эти стрелки к блокам и левому краю диаграммы.

- «Режим добавления функциональных блоков» – позволяет добавлять на диаграммы новые блоки путём клика левой кнопки мыши на диаграмме.

- «Режим работы со стрелками» – позволяет добавлять стрелки на диаграммы.

- «Режим размещения тильд» – позволяет размещать на диаграмме тильды связующие название стрелки и саму стрелку на диаграмме, для этого достаточно в данном режиме кликнуть левой кнопкой мыши на стрелку в месте, где должна начинаться тильда. Тильды

не являются обязательным атрибутом IDEF0-диаграмм, но упрощают чтение диаграмм.

- «Режим добавления внешних ссылок» – позволяет размещать на диаграмме сущность диаграмм DFD «ссылка». Пункт доступен только для редактирования диаграмм DFD.

- «Режим добавления хранилищ данных» – позволяет размещать на диаграмме сущность диаграмм DFD «хранилище данных». Доступен только для редактирования диаграмм DFD.

- «Режим добавления текста» – данный режим позволяет добавить произвольный текст в любом месте диаграммы. Режим используется для добавления комментариев непосредственно на диаграмму.

- «Сетка» – данный пункт позволяет включать и отключать видимость сетки, по узлам которой выравниваются функциональные блоки диаграммы и текстовые рамки (названия стрелок, комментарии).

- «Перейти к родительской диаграмме» – данный пункт позволяет переместиться из текущей диаграммы в родительскую.

- «Перейти к дочерней диаграмме» – данный пункт позволяет перейти из текущей диаграммы, в одну из дочерних. Для этого на диаграмме должен быть выделен необходимый блок.

4.1.2. Контекстное меню блока

Для редактирования свойств блока следует щелкнуть по нему правой кнопкой мыши (МП). Появляется контекстное меню (рис. 33).



Рисунок 33 – Контекстное меню

Контекстное меню блока содержит следующие пункты [21]:

- «Удалить» – позволяет удалить блок, доступен только для блоков без декомпозиции.
- «Применить параметры внешнего вида к дереву объектов» – позволяет применить цвет и шрифт текущего блока для дочерних ему блоков.
- «Открыть в новой вкладке» – позволяет открыть диаграмму декомпозиции текущего блока в новой вкладке окна.
- «Собственник» – позволяет открыть список, в котором можно выбрать собственника текущего блока. В данный список попадают элементы, которые являются механизмами данного блока, и только те элементы, которые принадлежат к классификаторам, которые выбраны как содержащие наименования собственников процессов.
- «Тип» – позволяет открыть список, в котором можно выбрать тип текущего блока (комплекс процессов; процесс; подпроцесс; операция; действие).
- «Центровать присоединенные стрелки» – позволяет упорядочить размещение присоединенных к данному блоку стрелок.
- «Добавить уровень» – позволяет «вклинить» дополнительный уровень иерархии в существующую иерархию блоков.
- «Удалить уровень» – позволяет удалить лишний уровень иерархии блоков. При этом текущий блок должен иметь только один дочерний блок.
- «Копировать в файл» – позволяет создать новый файл, который будет содержать дочерние к выделенному блоку диаграммы.
- «Вставить из файла» – позволяет вставить модель из другого файла, при этом все диаграммы данного файла станут дочерними к выделенному блоку.
- «Редактировать активный элемент» – позволяет открыть окно со свойствами блока. Это окно можно открыть также двойным щелчком левой кнопки мыши по блоку. Данное окно содержит ряд стандартных вкладок, которые позволяют изменить внешний вид блока.

4.1.3. Контекстное меню стрелки

Контекстное меню стрелки (рис. 34) содержит следующие пункты:

- «Удалить» – позволяет удалить стрелку.
- «Тильда» – позволяет на диаграмме связать название стрелки и саму стрелку тильдой.

- «Прозрачный текст» – позволяет отключить или включить прозрачность фона поля названия стрелки на диаграмме.

- «Туннель» – позволяет использовать на диаграммах туннелирование стрелок. Пункт доступен только при вызове контекстного меню стрелки на отрезке, обрамлённом скобками. При выборе данного пункта откроется небольшое диалоговое окно с возможными вариантами действий. В окне представлены два основных действия: «Создать стрелку» – позволяет отменить туннелирование и создать отрезок стрелки на родительской диаграмме или внутри функционального блока (зависит от расположения туннеля); «Обозначить стрелку круглыми скобками» – позволяет, собственно, туннелировать стрелку. В случае если туннелируются стрелки на входе в блок (стрелки управления, входов, механизмы) то доступен и третий вариант: «Обозначить стрелку круглыми скобками (показывать в отчётах для дочерних элементов)». Выбор данного варианта обозначает, что данная стрелка будет считаться стрелкой управления, входа или механизма соответственно, для всех дочерних блоков данного блока, хотя на диаграммах этого и не будет показано. Данную функциональность следует использовать с целью повышения читабельности диаграмм, если на диаграммах присутствует множество объектов затрудняющих их восприятие.

- «Редактировать активный элемент» – позволяет открыть окно со свойствами стрелки. Это окно можно открыть, также, двойным щелчком левой кнопки мыши по стрелке. Данное окно содержит ряд вкладок с настройками внешнего вида стрелки, а также вкладку «Поток».

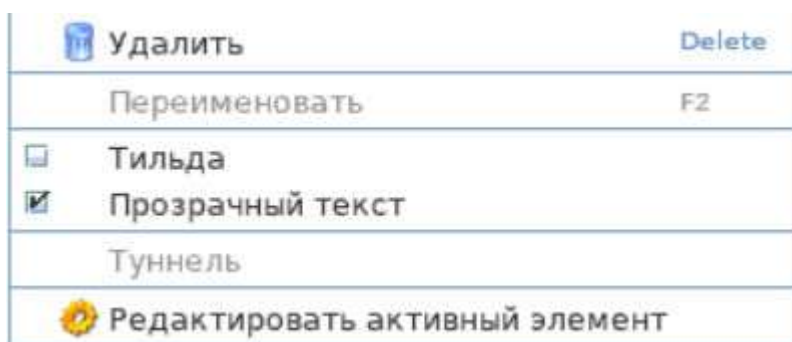


Рисунок 34 – Контекстное меню стрелки

На вкладке «Поток» (рис. 35) можно указать название стрелки, подпись текущего сегмента. Сегмент стрелки – это часть стрелки между двумя перекрёстками стрелки, между перекрёстком и краем диа-

граммы, между перекрестком и функциональным блоком, между функциональным блоком и другим функциональным блоком на одной диаграмме. Стрелка с одним названием может иметь много сегментов с разными подписями. Если подпись сегмента не задана, то на диаграмме отражается название стрелки.

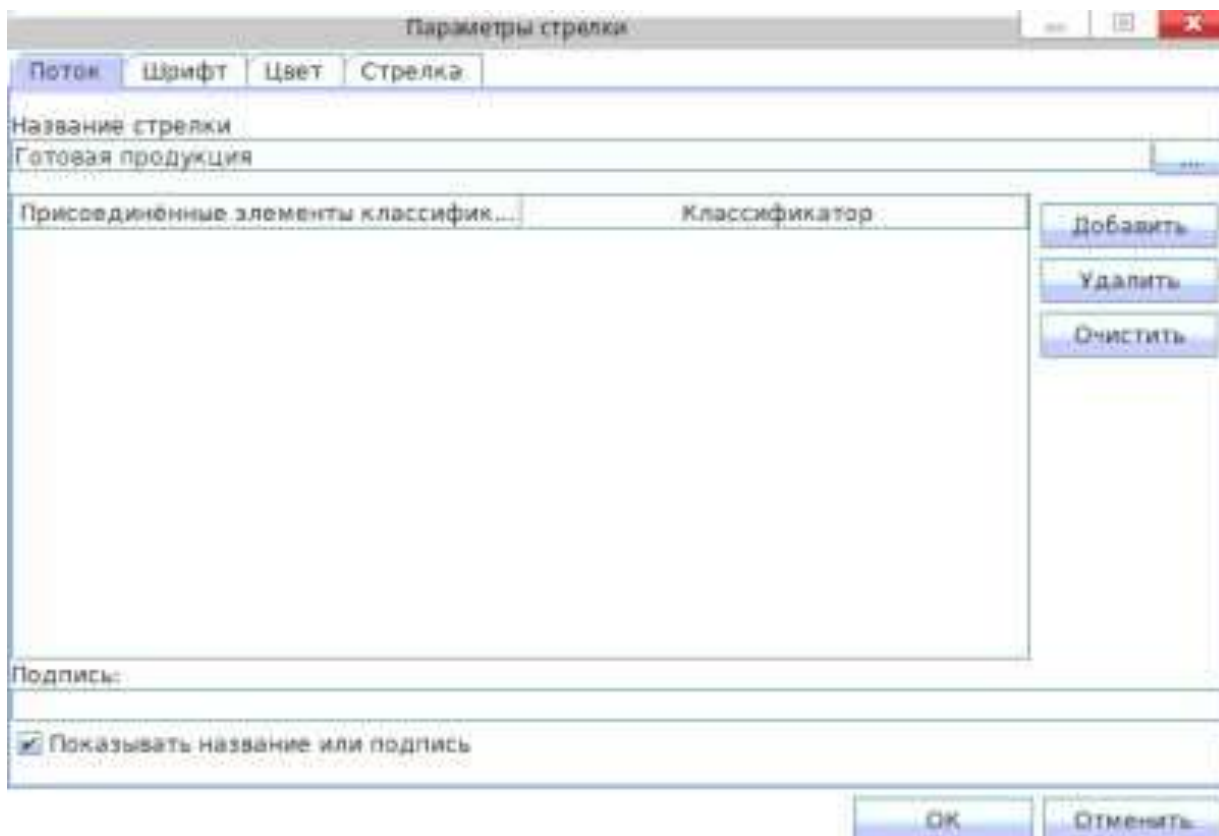


Рисунок 35 – Вкладка «Поток»

4.1.4. Меню диаграммы

Кликнув в верхней части диаграммы правой кнопкой, можно вызвать контекстное меню диаграммы. В основной части диаграммы клик правой кнопкой зарезервирован под функцию сброса режима. В данном меню присутствуют пункты изменения свойств диаграммы и модели.

Пункт вызова окон «Свойства модели» и «Свойства диаграммы» открывают окна для настройки основных свойств модели и диаграммы, а именно: указать автора диаграммы; указать, кем используется данная диаграмма; указать статус диаграммы; указать читателя диаграммы. Наличие всех этих свойств диаграммы предусматривает методология функционального моделирования IDEF0.

4.2. Построение моделей IDEF0

Рекомендуется следующая последовательность при построении функциональной модели [11]:

- формирование цели моделирования;
- выбор точки зрения;
- определение области моделирования;
- создание блока контекстной диаграммы;
- создание стрелок на контекстной диаграмме;
- создание диаграмм декомпозиции;
- создание диаграмм узлов и иллюстраций.

Модель не должна создаваться без ясного осознания цели моделирования. При выборе цели следует ответить на такие вопросы: почему моделируется данный процесс, что выявит данная модель, как ее смогут применять? Вот пример цели моделирования: выявить задачи каждого сотрудника компании для разработки руководства по обучению новых сотрудников.

С методической точки зрения при моделировании полезно использовать мнение специалистов, имеющих разные взгляды на предметную область. Но модель должна разрабатываться исходя из единой точки зрения. Другие точки зрения отражаются в виде диаграмм иллюстраций.

Основой для выбора точки зрения должна служить поставленная цель моделирования. Наименованием точки зрения может являться название должности или подразделения (например, менеджер по продажам). Как и в случае с определением цели моделирования, четкое определение точки зрения необходимо для обеспечения внутренней целостности модели и предотвращения постоянного изменения ее структуры.

При определении области моделирования необходимо учитывать два компонента – широту и глубину. Широта подразумевает определение границ модели – мы определяем, что будет рассматриваться внутри системы, а что снаружи. Глубина определяет, на каком уровне детализации модель является завершенной.

После определения области предполагается, что новые объекты не должны вноситься в систему, так как все объекты модели взаимосвязаны и внесение нового объекта может изменить существующие взаимосвязи.

Стрелки удобно проектировать в таком порядке: выход, вход, механизм, управление.

Каждый блок обозначает отдельную функцию, и она часто имеет четко описываемые результаты работы. Наличие неясностей при анализе выходов блока – возможный сигнал необходимости проведения реинжиниринга рассматриваемого бизнес-процесса.

После идентификации возможных выходов полезно провести анализ модели на предмет предвидения всех возможных сценариев поведения процесса. Это означает, что если существует вероятность возникновения той или иной ситуации в ходе процесса, модель должна ее отражать. Важно не забыть отразить негативные результаты работы блоков. Негативные результаты часто используются в качестве обратных связей; их анализ должен проводиться для каждого блока.

Входы можно рассматривать как особым образом преобразуемые блоками сырье или информацию для получения выходов.

После создания входов и выходов можно приступить к рассмотрению механизмов исполнения или ресурсов, относящихся к блоку.

Наконец, должно быть определено управление, контролирующее ход работы блока. Все блоки должны иметь хотя бы одну стрелку управления.

После завершения работы над контекстной диаграммой следует задать следующие вопросы. Обобщает ли диаграмма моделируемый бизнес-процесс? Согласуется ли диаграмма с областью, точкой зрения и целью моделирования? Подходит ли выбранный уровень детализации стрелок для контекстного блока?

Блок декомпозируется, если необходимо детально описать его работу. При декомпозиции блока полезно рассмотреть его жизненный цикл – это поможет определить блоки дочерней диаграммы. Например, жизненный цикл блока «Поджарить картофель» может быть таким: подготовить продукты, почистить картофель, подогреть масло и т. д.

Важно помнить, что граница дочерней диаграммы есть граница родительского блока. Это означает, что вся работа выполняется блоками самого нижнего уровня. В отличие от иерархии, применяемой в структурном программировании, блоки верхнего уровня не являются субъектами управления для блоков нижнего уровня. Это значит, что дочерние объекты – это те же объекты, что и их родители, только показанные с большей детализацией. Действия генерального директора компании могут отражаться на IDEF-диаграммах рядом с действиями простых рабочих.

Модели могут проектироваться как с использованием подхода в ширину, когда каждая диаграмма максимально детализируется перед своей декомпозицией, так и с подходом в глубину, когда сначала определяется иерархия блоков, а потом создаются соединяющие их стрелки.

Сформулированная цель моделирования содержит вопросы, на которые должна отвечать модель. Когда становится возможным получение ответов на них с помощью модели, последняя считается удовлетворяющей поставленным требованиям и рассматривается как завершенная. При построении декомпозиции нужно следить, чтобы все блоки на диаграмме лежали внутри определенных ранее границ моделирования. Еще одно правило состоит в том, что моделирование следует продолжаться до тех пор, пока стрелки входа и выхода преобладают на диаграммах.

4.3. Модели «как есть» и «как будет»

Целью построения функциональных моделей обычно является выявление наиболее слабых и уязвимых мест деятельности компании, анализе преимуществ новых бизнес-процессов и степени изменения существующей структуры организации бизнеса [7].

Анализ начинают с построения модели «как есть» (as-is), то есть модели существующей организации работы. Модель «как есть» может создаваться на основе изучения документации (должностных инструкций, положений о предприятии, приказов, отчетов), анкетирования и опроса служащих предприятия и других источников.

С помощью синтаксического анализа модели можно легко обнаружить «бесполезные» (не имеющие выхода), «неуправляемые» (не имеющие управления) и «простаивающие» функции. Более тонкий анализ позволяет выявить дублирующие, избыточные или неэффективные функции. Модель дает целостное представление о работе системы в целом и возможность понять взаимосвязи всех составляющих системы. При этом выясняется, что обработка информации и использование ресурсов неэффективны, важная информация не доходит до соответствующего рабочего места. Признаком неэффективности организации работ является, например, отсутствие обратных связей по входу и управлению для важных функций.

Исправление недостатков, перенаправление информационных и материальных потоков приводит к созданию модели «как будет» (to-be).

Только на основе модели «как будет» проектируется модель данных и затем информационная система. Построение модели на основе модели «как есть» приводит к тому, что информационная система автоматизирует несовершенные бизнес-процессы и дублирует, а не заменяет существующий документооборот.

4.4. Описание деятельности компании

Для создания учебной модели рассмотрим деятельность вымышленной компании, которая занимается изготовлением и продажей полуфабрикатов – быстрозамороженных пельменей и вареников.

Последовательность основных процедур в нашей компании такова:

- менеджеры принимают заказы от клиентов;
- рабочие изготавливают продукцию;
- рабочие проводят контроль качества продукции;
- кладовщик отгружает клиентам заказы.

Компания использует бухгалтерскую информационную систему, которая позволяет оформить заказ, выписать счет и отследить платежи по счетам.

4.5. Создание контекстной диаграммы

При запуске «Ramus» появляется окно, в котором предлагается создать новый проект (по умолчанию) или же открыть уже существующий файл проекта. Данное окно не будет выводиться в дальнейшем если поставить галочку «Использовать выбор по умолчанию и больше не спрашивать».

При выборе пункта «Создать новый файл» автоматически запускается мастер создания нового проекта. Этот мастер можно закрыть и заполнить необходимые сведения позже, но рекомендуется выполнить все его шаги.

На первом шаге предлагается внести сведения об авторе, названии проекта и модели. Также можно выбрать тип нотации модели: IDEF0 или DFD.

На втором шаге предлагается внести сведения о том, в какой организации используется данный проект.

На третьем – предлагается дать короткое описание проекту.

На четвертом – предлагается создать несколько основных классификаторов проекта. Например: «Документы», «Персонал» и т. д.

На пятом, заключительном, шаге предлагается выбрать, какие классификаторы из созданных будут содержать перечень собственников процессов. Это важно для дальнейшей работы над проектом. Например, это мог бы быть классификатор «Персонал»

После завершения работы мастера, откроется рабочее пространство «Диаграммы», в котором можно приступить к рисованию графической модели.

4.5.1. Контекст

1. Запустите CASE-средство: выберите **Пуск, Программы, Ramus**

Education, Ramus Education.

2. Создайте новую модель: в окне «Начало работы» выберите **Создать новый файл**, нажмите **ОК**, в поле «Автор» введите **Куприянова К.А.**, в поле «Название проекта» введите **МБП**, в поле «Название модели» введите **Деятельность компании**, выберите тип модели **IDEFO**, нажмите **Дальше**, в поле «Используется в» введите **кафедра ИСиТ КрасГАУ**, нажмите **Дальше**, в области «Описание» введите **Это учебная модель, описывающая деятельность компании**, нажмите **Дальше**, раздел классификаторы оставьте незаполненным, нажмите **Дальше**, нажмите **Окончить**.

Через меню **Диаграмма, Свойства модели** можно отредактировать метаданные модели, а именно: название модели, описание, место ее использования.

3. Создайте текстовый объект: нажмите на кнопку **T**, выберите мышью в левом нижнем углу области моделирования.

4. Перейдите в режим редактирования: выберите текстовый объект, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Название**, введите **Цель: Моделировать текущие бизнес-процессы компании**, нажмите **ОК**.

5. Создайте текстовый объект: нажмите на кнопку **T**, выберите мышью в левом нижнем углу области моделирования.

6. Перейдите в режим редактирования: выберите текстовый объект, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Название**, введите **Точка зрения: Директора**, нажмите **ОК**.

4.5.2. Блок

7. Активируйте окно модели, кликнув мышью на область моделирования.

8. Создайте блок контекстной диаграммы: нажмите на кнопку **Режим добавления функциональных блоков**.

9. Перейдите в режим редактирования: выберите блок, нажмите **МП**, выберите **Редактировать активный элемент**, в закладке **Название** введите **Деятельность компании**.

10. Выберите цвет: выберите вкладку **Цвет фона**, выберите **розовый**.

11. Выберите начертание: выберите вкладку **Шрифт**, в окне **Стиль** выберите **Жирный**, нажмите **ОК**.

На экране должна быть контекстная диаграмма, состоящая из одного блока: см. диаграмму А (приложение 2).

4.5.3. Граничные стрелки

12. Создайте граничную стрелку выхода: выберите на палитре инструментов кнопку **Режим работы со стрелками**, переместите курсор на правую грань блока, пока не появится треугольник, щелкните мышью по правой грани, переместите курсор к правой стороне экрана, пока не появится полоска, щелкните мышью по полоске, выберите в палитре инструментов кнопку курсор .

На экране должна быть граничная стрелка выхода.

13. Введите метку: выберите выходную стрелку, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Поток**, в поле **Название стрелки** введите **Проданная продукция**.

14. Определите свойства стрелки: выберите вкладку **Шрифт**, в поле **Стиль** выберите **жирный**, в поле **Размер** выберите **10**, выберите вкладку **Цвет**, выберите **зеленый**, выберите вкладку **Стрелка**, выберите тип **потолще**, нажмите **ОК**.

15. Создайте граничную стрелку входа: выберите на палитре инструментов кнопку **Режим работы со стрелками**, переместите курсор к левой стороне экрана, пока не появится полоска, щелкните мышью по полоске, переместите курсор на левую грань блока, пока не появится треугольник, щелкните мышью по левой грани, выберите в палитре инструментов кнопку курсор.

На экране должна быть граничная стрелка входа.

16. Введите метку: выберите выходную стрелку, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Поток**, в поле **Название стрелки** введите **Звонки клиентов**.

17. Выберите свойства стрелки: выберите вкладку **Шрифт**, в поле **Стиль** выберите **жирный**, в поле **Размер** выберите **10**, выберите вкладку **Цвет**, выберите **красный**, выберите вкладку **Стрелка**, выберите тип **потолще**, нажмите **ОК**.

18. Создайте граничные стрелки по таблице 3.

Результат выполнения должен быть как на диаграмме А (приложение 2).

19. Сохраните модель: выберите **Файл, Сохранить как**, в поле **Имя** файла введите **Пельмени_as_is**, в поле **Тип** выберите ***.rsf**, выберите **Сохранить**.

Таблица 3 – Стрелки контекстной диаграммы

Имя	Описание	Тип
Бухгалтерская система	Оформление счетов, оплата счетов, работа с заказами	Механизм
Правила и процедуры	Правила продаж, инструкции по производству, процедуры контроля качества	Управление

4.6. Создание диаграммы декомпозиции

4.6.1. Декомпозиция

20. Создайте диаграмму декомпозиции: выберите блок, выберите кнопку **Перейти к родительской диаграмме** в панели инструментов, в диалоговом окне укажите число работ на диаграмме нижнего уровня **3**, выберите нотацию декомпозиции **IDEF0**, выберите шаблон **простой**, нажмите **ОК**.

Автоматически будет создана диаграмма декомпозиции. На экране должна быть диаграмма декомпозиции, которая содержит три блока и несвязанные стрелки.

21. Введите свойства каждого блока по таблице 4.

Таблица 4 – Блоки диаграммы декомпозиции А0

Имя	Описание
Продажи и маркетинг	Маркетинговые исследования
Изготовление и контроль качества	Изготовление и контроль качества пельменей и вареников
Отгрузка и получение	Отгрузка заказов клиентам и получение сырья от поставщиков

4.6.2. Связывание

При декомпозиции блока входящие и исходящие стрелки автоматически появляются на диаграмме декомпозиции, но при этом не касаются блоков. Такие стрелки называются несвязанными и воспринимаются как ошибки.

22. Свяжите стрелку: выберите на палитре инструментов кнопку **Режим курсора**, выберите наконечник стрелки **Звонки клиентов**, поместите курсор на левую грань блока **Продажи и маркетинг**, когда появится треугольник щелкните мышью по левой грани блока **Продажи и маркетинг**, стрелка должна соединиться с блоком.

23. Свяжите стрелку **Правила и процедуры** и верхнюю грань блока **Продажи и маркетинг** (см. 22).

24. Свяжите стрелку **Бухгалтерская система** и нижнюю грань блока **Продажи и маркетинг** (см. 22).

25. Свяжите стрелку: выберите на палитре инструментов кнопку **Режим курсора**, выберите основание стрелки **Проданная продукция**, поместите курсор на правую грань блока **Отгрузка и получение**, когда появится треугольник, щелкните мышью по правой грани блока **Отгрузка и получение**, стрелка должна соединиться с блоком.

4.6.3. Разветвление

Одни и те же данные или объекты могут использоваться, с одной стороны, сразу в нескольких функциях. С другой стороны, данные или объекты, порожденные в разных функциях, могут в дальнейшем использоваться в одном месте. Для моделирования таких ситуаций используют ветвление и слияние стрелок.

Смысл разветвляющихся и сливающихся стрелок передается метками каждого сегмента (ветви) стрелки. Метку обычно связывают с сегментом (ветвью) с помощью тильды.

Существуют следующие варианты именования таких стрелок.

- Не помеченные ветви содержат все объекты, указанные в метке стрелки перед ветвлением, то есть все объекты принадлежат каждой ветви.

- Ветви, помеченные после точки ветвления, содержат все объекты, указанные в метке стрелки перед ветвлением, или их часть, описываемую меткой каждой ветви.

- При слиянии не помеченных ветвей объединенная ветвь стрелки содержит все объекты, принадлежащие сливаемым ветвям и указанные в общей метке стрелки после слияния.

- При слиянии помеченных ветвей объединенная ветвь содержит все или некоторые объекты, принадлежащие сливаемым ветвям и перечисленные в общей метке после слияния. Если общая метка после слияния отсутствует, это означает, что общая ветвь передает все объекты, принадлежащие сливаемым ветвям.

26. Выполните разветвление стрелки: выберите на палитре инструментов кнопку **Режим работы со стрелками**, выберите стрелку **Правила и процедуры, М2**, выберите верхнюю грань блока **Отгрузка и получение**, стрелка должна соединиться с блоком, выберите в палитре инструментов кнопку **Режим курсора**.

27. Выполните разветвление стрелки **Правила и процедуры** и свяжите с верхней гранью блока **Изготовление и контроль качества**.

28. Выполните разветвление стрелки **Бухгалтерская система** и свяжите с нижней гранью блока **Отгрузка и получение**.

29. Определите свойства стрелки: выберите ветвь стрелки управления блока **Изготовление и контроль качества**, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Поток**, в поле **Название стрелки**, введите **Правила изготовления и контроля качества**, нажмите **ОК**.

4.6.4. Внутренние стрелки

30. Создайте внутреннюю стрелку связи выход-управление: выберите на палитре инструментов **Режим работы со стрелками**, выберите правую грань блока **Продажи и маркетинг**, выберите верхнюю грань блока **Изготовление и контроль качества**, стрелка должна соединиться с блоком, выберите в палитре инструментов кнопку **Режим курсора**.

31. Определите метку стрелки **Заказы клиентов**: см. 29.

32. Создайте внутреннюю стрелку связи выход-вход: выберите на палитре инструментов **Режим работы со стрелками**, выберите правую грань блока **Изготовление и контроль качества**, выберите левую грань блока **Отгрузка и получение**, стрелка должна соединиться с блоком, выберите в палитре инструментов кнопку **Режим курсора**.

33. Определите метку стрелки **Готовая продукция**: см. 29.

34. Создайте внутреннюю стрелку обратной связи по управлению: выберите на палитре инструментов **Режим работы со стрелками** выберите правую грань блока **Изготовление и контроль качества**, выберите верхнюю грань блока **Продажи и маркетинг**, стрелка должна соединиться с блоком, выберите в палитре инструментов кнопку **Режим курсора**.

35. Определите метку стрелки **Результаты изготовления и контроля качества**: см. 29.

36. Измените стиль стрелки: выберите стрелку **Результаты изготовления и контроля качества**, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Стрелка**, выберите тип **пунктир**.

4.6.5. Словарь стрелок

Имена стрелок автоматически заносятся в словарь стрелок, который предназначен для хранения имен.

37. Просмотрите имена стрелок: выберите любую стрелку, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Поток**, напротив поля Название стрелки выберите **многоточие**.

4.6.6. Туннель

Туннель – круглые скобки () в начале и/или окончании стрелки.

Туннельные стрелки означают, что данные, выраженные этими стрелками, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме.

Существуют следующие типы туннелирования:

- Стрелка, помещенная в туннель там, где она присоединяется к блоку, означает, что данные, выраженные этой стрелкой, не обязательны на следующем уровне декомпозиции.

- Стрелка, помещаемая в туннель на свободном конце, означает, что выраженные ею данные отсутствуют на родительской диаграмме.

Вновь внесенная граничная стрелка на диаграмме декомпозиции нижнего уровня изображается в квадратных скобках [] и автоматически не появляется на диаграмме верхнего уровня. Такая стрелка называется неразрешенной.

Для миграции такой стрелки на диаграмму верхнего уровня необходимо в окне «Туннелирование стрелки» выбрать опцию «Создать стрелку». Для туннелирования необходимо выбрать опцию «Обозначить туннель круглыми скобками». В этом случае стрелка не попадет на другую диаграмму. Если на диаграмме нижнего уровня необходимо изобразить малозначимые объекты, то изображающие их стрелки можно поместить в тоннель на самом нижнем уровне.

38. Создайте граничную стрелку выхода **Маркетинговые материалы** блока **Продажи и маркетинг**.

Созданная граничная стрелка автоматически не попадает на диаграмму верхнего уровня и имеет квадратные скобки [].

39. Выполните мигрирование стрелки на верхний уровень: выберите квадратные скобки [] на неразрешенной стрелке, **МП**, выберите **Туннель**, выберите **Создать стрелку**, нажмите **ОК**.

40. Результат должен быть, как на диаграмме **Б** (приложение 2).

4.7. Создание диаграммы декомпозиции **A2**

Для проведения декомпозиции блока «Изготовление и контроль качества» было проведено анкетирование сотрудников компании. В результате получена следующая информация:

- Производственный цех получает заказы клиентов из отдела продаж.
- Мастер координирует работу рабочих производственного цеха и дает указание на отгрузку продукции.
- Рабочие производственного цеха изготавливают пельмени и вареники в соответствии с рецептурой и указаниями мастера.
- Контролеры производят контроль качества готовой продукции. Забракованная продукция отправляется на переработку на другое предприятие.
- Контролеры направляют результаты контроля качества мастеру, который принимает решение о передаче готовой продукции на отгрузку.

41. Создайте диаграмму-декомпозицию: выберите блок **Изготовление и контроль качества**, выберите кнопку **Перейти к дочер-**

ней диаграмме, в счетчике выберите 4 новых блока, выберите нотацию декомпозиции **IDEFO**, выберите шаблон **простой**, нажмите **ОК**.

На экране должна быть диаграмма декомпозиции A2, которая содержит четыре блока и несвязанные стрелки.

42. Введите свойства каждого блока по таблице 5.

43. Введите стрелки по таблице 6.

Таблица 5 – Блоки диаграммы декомпозиции A2

Имя	Описание
Управление изготовлением и контролем качества	Просмотр заказов, просмотр результатов контроля качества, формирование групп заказов на изготовление и отгрузку
Изготовление пельменей	Изготовление пельменей в соответствии с рецептурой и указаниями мастера
Изготовление вареников	Изготовление вареников в соответствии с рецептурой и указаниями мастера
Контроль качества	Контроль качества готовой продукции. Отправка брака на переработку.

Таблица 6 – Стрелки диаграммы декомпозиции A2

№	Метка стрелки	Источник стрелки	Тип источника	Потребитель стрелки	Тип потребителя
1	2	3	4	5	6
1	Заказ от клиентов	Граница диаграммы	Control	Управление изготовлением и контролем качества	Control
2	Заказ на пельмени	Управление изготовлением и контролем качества	Output	Изготовление пельменей	Control
3	Заказ на вареники	Управление изготовлением и контролем качества	Output	Изготовление вареников	Control
4	Сырье	Tunnel	Input	Изготовление пельменей	Input
				Изготовление вареников	Input
5	Пельмени	Изготовление пельменей	Output	Контроль качества	Input

Окончание табл. 6

1	2	3	4	5	6
6	Вареники	Изготовление вареников	Output	Контроль качества	Input
7	Персонал производственного цеха	Tunnel	Mechanism	Изготовлениепельменей	Mechanism
				Изготовлениевареников	Mechanism
8	Мастер	Персонал производственного цеха		Управление изготовлением и контролем качества	Mechanism
9	Правила изготовления и контроля качества	Граница диаграммы		Изготовлениепельменей	Control
				Изготовлениевареников	Control
				Контроль качества	Control
10	Результаты изготовления и контроля качества	Изготовлениепельменей	Output	Граница диаграммы	Output
		Изготовлениевареников	Output		
		Контроль качества	Output		
11	Результаты контроля качества	Контроль качества	Output	Управление изготовлением и контролем качества	Input
12	Готовая продукция	Контроль качества	Output	Граница диаграммы	Output
13	Контролер	Персонал производственного цеха		Контроль качества	Mechanism
14	Указание передать продукцию на отправку	Управление изготовлением и контролем качества	Output	Контроль качества	Control
15	Отходы	Контроль качества	Output	Граница диаграммы	Output

44. Перенаправьте стрелку: выберите диаграмму **A0**, выберите стрелку **Отходы**, нажмите клавишу **Delete**, выберите **Да**, выберите на палитре инструментов **Режим работы со стрелками**, выберите правую грань блока **Изготовление и контроль качества**, выберите ле-

вую грань блока **Отгрузка и получение**, стрелка должна соединиться с блоком, выберите в палитре инструментов кнопку **Режим курсора**.

45. Введите метку стрелки **Отходы**.

46. Результат должен быть, как на диаграмме **В**.

4.8. Синтаксический контроль

Синтаксические ошибки IDEF0 в Ramus Education можно разделить на следующие типы:

- ошибки, которые RAMUS Education не в состоянии выявить, например, метка стрелки должна быть существительным. RAMUS Education не анализирует синтаксис естественного языка и смысл имен объектов, и поэтому игнорирует ошибки этого типа;

- есть синтаксические ошибки, которые RAMUS Education не допускает, например, невозможно создать на диаграмме IDEF0 внутреннюю стрелку, выходящую из левой грани блока и входящую в правую грань.

5-й день. ДОПОЛНЕНИЕ МОДЕЛИ ДИАГРАММОЙ BPMN

Обед, продолжавшийся около трёх часов, кончился; хозяин положил салфетку на стол – все встали и пошли в столовую, где ожидал их кофе, карты и продолжение попойки, столь славно начатой в столовой.

А. С. Пушкин. Дубровский, IX

5.1. Назначение

BPMN (Business Process Model and Notation, нотация и модель бизнес-процессов) – представляет собой графическую нотацию для отображения бизнес-процессов при моделировании потоков работ, происходящих в исследуемой системе [4].

Нотация BPMN была разработана организацией Business Process Management Initiative (BPMI), в настоящее время разработка BPMN ведётся консорциумом OMG (Object Management Group). В 2007 г. OMG выпустила спецификацию BPMN 2.0 [12].

Целью проекта BPMN является создание общей нотации разработки моделей бизнес-процессов для различных категорий специалистов: от аналитиков и экспертов, моделирующих бизнес-процессы, технических разработчиков, которые создают системы для выполнения этих процессов, до менеджеров различных уровней, которые должны понимать процессные диаграммы, чтобы принимать деловые решения.

Благодаря абстрактному представлению модели нотация BPMN позволяет наглядным образом описывать модели бизнес-процессов независимо от среды их функционирования. Для реализации нотации модели используются языки исполнения бизнес-процессов – BPMML (Business Process Modeling Language) и BPEL (Business Process Execution Language).

5.2. Основные понятия

Одной из причин создания BPMN явилась необходимость построения простого механизма для проектирования как простых, так и сложных моделей бизнес-процессов. Для удовлетворения двух этих противоречащих требований был применен подход систематизации графических элементов нотации по категориям. Результатом явился небольшой перечень категорий нотаций, позволивший людям, работаю-

щим с диаграммами BPMN, без труда распознавать основные типы элементов и осуществлять корректное чтение схем. Компания ELMA выполнила перевод спецификации BPMN 2.0 на русский язык [8].

Существуют пять основных категорий элементов:

- элементы потока (flow objects);
- данные (data);
- соединяющие элементы (connecting objects);
- зоны ответственности (swimlanes);
- артефакты (artifacts).

Элементы потока являются важнейшими графическими элементами, определяющими ход бизнес-процесса. Элементы потока, в свою очередь, делятся на:

- события (events);
- действия (activities);
- шлюзы (gateways).

Данные на диаграмме могут быть представлены любыми из следующих четырех элементов:

- объект данных (data objects);
- входные данные (data inputs);
- выходные данные (data outputs);
- хранилища данных (data stores).

Выделяют четыре вида соединяющих элементов потока, связывающихся друг с другом и с другими элементами:

- поток операций (sequence flow);
- поток сообщений (message flow);
- ассоциация (association);
- ассоциация данных (data associations).

Существуют два способа группировки основных элементов моделирования с помощью зон ответственности:

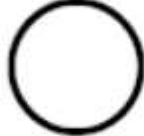

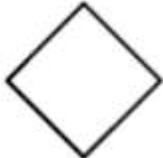


- группировка с помощью пула (pool);
- группировка с помощью дорожки (lane).

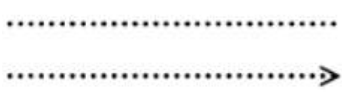



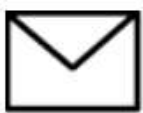

Артефакты используются для добавления дополнительной информации о процессе. На данный момент текущий перечень артефактов включает в себя следующие элементы:

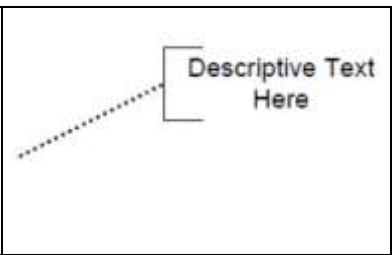
- группа (group);
- текстовая аннотация (text annotation).

5.3. Основные графические элементы моделирования

Основные графические элементы моделирования изображены при помощи графических нотаций [11].

Событие (event)	Событие – это то, что происходит в течение бизнес-процесса или его хореографии. Событие оказывает влияние на ход бизнес-процесса и чаще всего имеет причину (триггер) или воздействие (результат). Изображается в виде круга со свободным центром, предназначенным для дифференцировки внутренними маркерами различных триггеров или их результатов	
Действие (activity)	Действие – общий термин, обозначающий работу, выполняемую исполнителем в ходе бизнес-процесса. Действия могут быть либо элементарными, либо неэлементарными (составными). Выделяют следующие виды действий, являющихся частью модели процесса: подпроцесс (sub-process) и задача (task). Задача и подпроцесс изображаются в виде прямоугольников с закругленными углами	
Шлюз (gateway)	Шлюзы используются для контроля расхождений и схождений потока операций, как в процессе, так и в хореографии. Таким образом, данный термин подразумевает ветвление, раздвоение, слияние и соединение маршрутов. Внутренние маркеры указывают тип контроля развития бизнес-процесса	
Поток операций (sequence flow)	Поток операций служит для отображения того порядка, в котором организованы действия процесса или условия хореографии	
Поток сообщений (message flow)	Поток сообщений служит для отображения обмена сообщениями между двумя участниками, готовыми эти сообщения отсылать и принимать	

<p>Ассоциация (association)</p>	<p>Ассоциация служит для установления связи между информацией или артефактами (объектами, не относящимися к элементам потока) и элементами потока. Текстовые объекты, а также графические объекты, не относящиеся к элементам потока, могут соотноситься с элементами потока. При необходимости ассоциация может указывать направление потока (например, потока данных)</p>	
<p>Пул (pool)</p>	<p>Пул представляет собой участника взаимодействия. Пул также может выступать в качестве зоны ответственности или графического контейнера, отвечающего за разделение определенного набора действий, относящихся к другим пулам, что обычно встречается в ситуациях типа «бизнес для бизнеса» (B2B)</p>	
<p>Дорожка (lane)</p>	<p>Дорожка используется для отображения распределения ролей и может быть как вертикальной, так и горизонтальной. Служит для упорядочивания и категоризации действий</p>	
<p>Объект данных (data object)</p>	<p>Объект данных предоставляет информацию о том, какие действия необходимо выполнить и/или каков результат этих действий. Входные и выходные данные объекта данных представляют собой одну и ту же информацию о процессе</p>	
<p>Сообщение (message)</p>	<p>Сообщение используется для отображения сущности взаимодействия между двумя участниками бизнес-процесса</p>	
<p>Группа (group)</p>	<p>Группа предназначена для группировки графических элементов, принадлежащих одной и той же категории. Такая группировка не оказывает влияния на поток операций. На диаграмме бизнес-процесса название категории, к которой принадлежат сгруппированные элементы, отображается в качестве названия группы. Такого рода группировка может использоваться в целях составления документации или при проведении анализа. Графически группы отображаются так же, как и категории объектов</p>	

Текстовая аннотация (связана с ассоциацией) (text annotation)	Текстовые аннотации являются механизмом, позволяющим разработчику модели бизнес-процесса вводить дополнительную информацию для тех, кто работает с BPMN диаграммами	
---	---	---

5.4. Действия

Действия изображаются прямоугольниками со скругленными углами. Среди действий различают задания и подпроцессы. Графическое изображение свёрнутого подпроцесса снабжено знаком плюс у нижней границы прямоугольника.

Задание (task) – это единица работы, элементарное действие в процессе.

Множественные экземпляры (multiple instances) действия показывают, что одно действие выполняется многократно, по одному разу для каждого объекта. Например, для каждого объекта в заказе клиента выполняется один экземпляр действия. Экземпляры действия могут выполняться параллельно или последовательно.


Циклическое действие (loop activity) выполняется, пока условие цикла верно. Условие цикла может проверяться до или после выполнения действия.

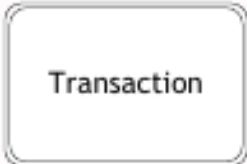


Развёрнутый подпроцесс (expanded subprocess) является сложным действием и содержит внутри себя собственную диаграмму бизнес-процессов.

Свёрнутый подпроцесс (collapsed subprocess) также является составным действием, но скрывает детали реализации процесса.







Ad-hoc подпроцесс (ad-hoc subprocess) содержит задания. Задания выполняются до тех пор, пока не выполнено условие завершения подпроцесса.

Основные действия [23]:








	<p>Задача – единица работы. Если задача помечена символом +, то задача является подпроцессом и может быть детализирована</p>
---	--

	<p>Транзакция – набор логически связанных действий. Для транзакции может быть определен протокол выполнения</p>
	<p>Событийный подпроцесс помещается внутри другого процесса. Он начинает выполняться, если инициируется его начальное событие. Событийный подпроцесс подпроцесс может или выполняться прерывать родительский параллельно с ним</p>
	<p>Вызывающее действие является точкой входа для глобально определенного подпроцесса, который повторно используется в данном процессе</p>

Маркер действия отражает поведение действия во время выполнения [22].

	<p>Маркер подпроцесса</p>
	<p>Маркер цикла</p>
	<p>Маркер параллельных множественных экземпляров (МЭ)</p>
	<p>Маркер последовательных множественных экземпляров (МЭ)</p>
	<p>Маркер ad hoc</p>
	<p>Маркер компенсации</p>

Тип задачи [22] определяет природу действия, которое будет выполнено.

	<p>Задача отправки сообщения</p>
	<p>Задача получения сообщения</p>
	<p>Пользовательская задача</p>
	<p>Неавтоматизированная задача</p>
	<p>Задача-бизнес-правило</p>
	<p>Задача-сервис</p>
	<p>Задача-сценарий</p>

5.5. События

Существует три типа событий, классифицированных по времени воздействия на ход процесса: начальные (start events), промежуточные (intermediate events) и конечные (end events). Начальные и конечные события представляют собой точки начала и окончания процесса и должны обязательно присутствовать на диаграмме.

В нотации BPMN определены следующие типы триггеров:

- сообщение (message) – исходит от некоторого участника или триггера процесса и предшествует началу, продолжению или окончанию некоторого действия процесса;
- таймер (timer) – устанавливает цикл времени течения процесса;
- правило (rule) – тестовая строка, описывающая некоторое правило, применяемое к событию;
- исключительное событие (exception) – при завершении некоторого действия информирует процесс о возникновении ошибки;
- компенсация (compensation) – показывает, как подпроцесс может быть скомпенсирован последовательностью отката;
- отмена (cancel) – указывает на отмену события;
- ссылка (link) – представляет собой механизм, обеспечивающий подключение окончания события одного потока процесса к началу события другого потока процесса;
- составное событие (multiple) – указывает на то, что событие может задействовать несколько путей развития процесса или продолжить процесс в случае наличия промежуточного события;
- условия (conditional events) позволяют интегрировать бизнес-правила в процесс;
- сигналы (signal events) рассылают и принимают сигналы между несколькими процессами;
- остановки (terminate events) приводят к немедленному завершению всего бизнес процесса (во всей диаграмме).

На рис. 36 представлен перечень графических элементов событий [22].








События

	Начальные			Промежуточные			Завершающие	
	Верхнеуровневые	Прерывающие событийный подпроцесс	Не прерывающие событийный подпроцесс	Обрабатывающие	Граничные прерывающие	Граничные непрерывающие		Генерирующие
Простое: нетипизированное событие, обычно показывающее начало или окончание процесса.								
Сообщение: получение и отправка сообщений.								
Таймер: циклические события, моменты времени, временные периоды и таймауты.								
Эскалация: перенос рассмотрения вопроса на более высокий уровень организационной иерархии								
Условное: реакция на изменение бизнес-условий или интеграция бизнес-правил.								
Ссылка: пара соответствующих ссылок эквивалента потоку последовательности.								
Ошибка: генерация и обработка заданного типа ошибок.								
Отмена: обработка отмены транзакции или инициирование отмены.								
Компенсация: обработка или инициирование компенсации.								
Сигнал: передается между процессами и может обрабатываться многими получателями.								
Составное: обработка одного события из множества или генерация всех определенных событий.								
Параллельное составное: обработка всего множества параллельных событий.								
Останов: вызывает немедленное прекращение выполнения процесса.								

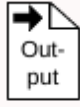

Рисунок 36 – События

5.6. Шлюзы

Шлюзы изображаются ромбами и представляют точки принятия решений в процессе. С помощью логических операторов организуется ветвление и синхронизация потоков управления в модели процесса [22].

	<p>Оператор исключающего ИЛИ, управляемый данными (data-based exclusive gateway). При ветвлении направляет поток лишь по одной из исходящих ветвей. При синхронизации потока оператор ожидает завершения одной входящей ветви и активирует исходящий поток управления</p>
	<p>Оператор исключающего ИЛИ, событийный. (event-based exclusive gateway). Предшествует только событиям обработки или заданиям-обработчикам сообщений. Поток управления направляется по той ветви, где событие произошло раньше</p>
	<p>Оператор И (parallel gateway) При разделении на параллельные потоки все ветви активируются одновременно. При ждет синхронизации завершения параллельных всех входящих ветвей ветвей оператор и затем активирует исходящий поток</p>
	<p>Оператор включающего ИЛИ (inclusive gateway) При ветвлении активируется одна или более ветвей. При слиянии все выполняющиеся входящие ветви должны быть завершены</p>
	<p>Сложный оператор (complex gateway). Моделирует сложные условия ветвления и слияния</p>
	<p>Оператор исключающего ИЛИ, событийный, создает новый экземпляр, (exclusive event-based gateway (instantiate)). Наступление каждого из последующих событий создает экземпляр процесса</p>
	<p>Оператор И, событийный, создает новый экземпляр, (parallel event-based gateway (instantiate)). Наступление всех последующих событий создает экземпляр процесса</p>


5.7. Данные [22]

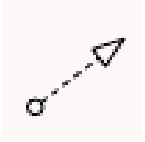
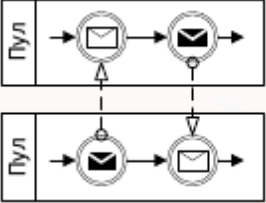
	Входные данные – внешний для вход процесса. Действия могут использовать эти данные
	Выходные данные – результат выполнения процесса
	Объект данных представляет информацию, которая обрабатывается в ходе процесса, например, документ или письмо
	Коллекция объектов данных представляет группу объектов, несущих информацию, например, список заказанных товаров
	Хранилище данных – объект, который процесс может использовать для записи и извлечения данных, например, база данных. Хранилище данных позволяет хранить данные после окончания жизненного цикла экземпляра процесса
	Сообщение позволяет явно показать передачу информации в ходе общения двух участников

5.8. Пулы

Роли – визуальный механизм организации различных действий в категории со сходной функциональностью. Существует два типа ролей:

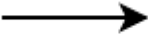


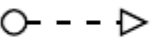
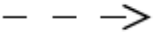
- пулы изображаются прямоугольником, который содержит несколько объектов потока управления, соединяющих объектов и артефактов;
- дорожки представляют собой часть пула, позволяют организовать объекты потока управления, связывающие объекты и артефакты [22].

	Пулы (участники) и дорожки отражают распределение обязанностей. Пул или дорожка обозначает организацию, роль или систему. Дорожки позволяют иерархически делить пулы и другие дорожки
---	---

	<p>Поток сообщений описывает информационный поток между участниками процесса. Поток сообщений может присоединяться к пулам, действиям или событиям-сообщениям</p>
	<p>Порядок обмена сообщениями может быть задан при помощи потока сообщений и потока управления</p>

5.9. Соединяющие элементы

Соединяющие элементы используются для соединения элементов потока процесса [8].

	<p>Последовательный (простой) поток (sequence flow) используется для отображения порядка следования действий процесса. Графически последовательный поток изображается сплошной линией с закрашенной стрелкой</p>
	<p>Условный поток – последовательный поток с условным выражением, измеряемым по времени выполнения, с целью определить, будет ли использоваться поток. Графически условный поток изображается сплошной линией с закрашенной стрелкой и ромбиком на противоположном конце</p>
	<p>Поток по-умолчанию – условный поток, который будет использоваться в случае, если все другие условные потоки не верны при выполнении. Графически поток по-умолчанию изображается сплошной линией с закрашенной стрелкой и косой чертой на противоположном конце</p>
	<p>Поток сообщений (message flow) используется для отображения потока сообщений между двумя отдельными участниками процесса. Графически поток сообщений изображается пунктирной линией с незакрашенной стрелкой и незакрашенным кружочком на противоположном конце</p>
	<p>Ассоциация (association) используется для того, чтобы связать данные, текст и другие артефакты с потоком объектов процесса. Графически ассоциация изображается пунктирной линией с V-образной стрелкой</p>

5.10. Декомпозиция блока IDEF0 в диаграмму BPMN

1. Запустите Modelio.
2. Создайте проект: выберите **File, Create Project**, введите имя **Пельмени**.
3. Создайте диаграмму: выберите **Пельмени, M2**, выберите **Create diagram, Diagram creation wizard**, выберите **Bpmn process collaboration diagram**, в поле **Name** введите **Изготовление пельменей**, нажмите ОК.
4. Создайте шесть задач (task): Подготовка ингредиентов, Приготовление фарша, Приготовление теста, Формование пельменей, Замораживание пельменей, Упаковка и складирование готовой продукции.
5. Создайте события (event): start event, end event.
6. Создайте шлюзы (gateways) по рисунку 37.

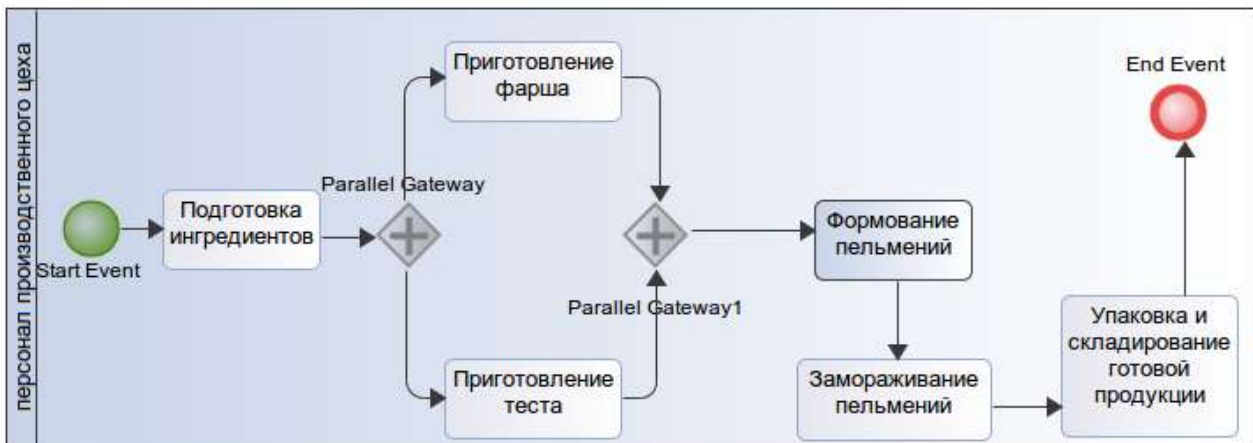


Рисунок 37 – Диаграмма BPMN

7. Соедините элементы диаграммы с помощью sequence flow.
8. Введите свойства: выберите задачу **Приготовление фарша, M2**, в поле **Description** введите данные из таблицы 7, выберите **Close**, выберите **File, Save project**.
9. Введите свойства задач по таблице 7.
10. Результат должен быть, как на рис. 37.

Таблица 7 – Свойства задач

Имя	Objects	Facts	Constrains
Приготовление фарша	Мясо свинины и говядины	Используются агрегаты: мясорубка, фаршемешалка	Фарш должен быть однородным, не слишком сухим, но и не «скользким» из-за избытка жира
Приготовление теста	Пшеничная мука, яйца, растительное масло, естественные красители	Используются агрегаты: мукопросеиватель, тестомес	Тесто должно быть эластичным и не липким
Формованиепельменей	Тесто, фарш	Используетсяпельменный агрегат НПА-1М (производительность 50 кг/ч, мощность 0,756 кВт, масса 72 кг, стоимость 35 000 руб.)	–
Замораживаниепельменей	Изготовленныепельмени	Используетсяхолодильная камера с объемом 10 м ³	Камера обеспечивает охлаждение продукции, произведенной за смену в течение следующих суток
Упаковка и складирование готовой продукции	Замороженныепельмени	Используетсяполуавтомат РТ-УМ-11 («Русская трапеза») и низкотемпературная камера (-18 ⁰ С)	–

6-й день. РЕИНЖИНИРИНГ БИЗНЕС-ПРОЦЕССОВ КОМПАНИИ ПО ПРОИЗВОДСТВУ ПЕЛЬМЕНЕЙ

Им овладело беспокойство,
Охота к перемене мест
(Весьма мучительное свойство,
Немногих добровольных крест).
А.С. Пушкин. Евгений Онегин, 8, XIII

6.1. Расщепление модели

1. Выполните декомпозицию блока **Отгрузка и получение** на три дочерних блока.
2. Введите свойства блоков по таблице 8.

Таблица 8 – Декомпозиция блока «Отгрузка и получение»

Имя	Описание
Получить сырье	Получить сырье и сделать соответствующие записи в ИС
Доставить сырье	Доставить сырье в производственный цех
Отгрузить продукцию и отходы	Отгрузить продукцию клиентам и отходы

3. Свяжите граничные стрелки по диаграмме Г.
4. Введите внутренние и граничные стрелки по таблице 9.

Таблица 9 – Стрелки декомпозиции блока «Отгрузка и получение»

Имя	Описание
Отходы на вывоз	Неутилизированный брак
Сырье	Сырье, доставленное в производственный цех
Оприходованное сырье	Оприходованное сырье для передачи в производственный цех
Сырье от поставщика	

5. Туннелируйте граничные стрелки.
6. Результат должен быть, как на диаграмме Г.
7. Перенаправьте стрелку: выберите диаграмму **A0**, выберите стрелку **Сырье**, нажмите клавишу **Delete**, выберите **Да**, выберите на палитре инструментов **Режим работы со стрелками**, выберите правую грань блока **Отгрузка и получение**, выберите левую грань блока **Изготовление и контроль качества**, стрелка должна соединиться с блоком, выберите в палитре инструментов кнопку **Режим курсора**.
8. Введите метку стрелки **Сырье**.
9. Результат должен быть как на диаграммах Д и Е.

6.2. План реинжиниринга

В результате анализа бизнес-процессов нашей компании принимается решение реорганизовать функцию «Изготовление и контроль качества», а функции «Продажи и маркетинг» и «Отгрузка и получение» оставить без изменений.

Для этого решено создать отдел кулинарии, который должен разрабатывать рецептуру изготовления пельменей и инструкции по изготовлению, по процедуре контроля качества.

6.3. Модификация модели

10. Измените данные контекста модели: выберите блок **Деятельность компании**, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Название**, введите **Предлагаемая модель деятельности компании**, нажмите **ОК**.

11. Измените цель: выберите объект **цель**, нажмите **МП**, выберите **Редактировать активный элемент**, выберите вкладку **Название**, введите **Документировать предлагаемые изменения бизнес-процессов компании**, нажмите **ОК**.

12. Сохраните модель: выберите **Файл, Сохранить как**, в поле **Имя файла** введите **Пельмени_to_be**, в поле **Тип** выберите ***.rsf**, выберите **Сохранить**.

13. Переименуйте блок: выберите блок **Изготовление и контроль качества**, нажмите **МП**, выберите **Редактировать активный**

элемент, выберите вкладку **Название**, введите **Производство продукции**, нажмите **ОК**.

14. Переместите блок **Контроль качества** на диаграмму **A2.1 Изготовление пельменей**.

15. Переименуйте блок **Изготовление пельменей** на диаграмме **A0 в Изготовление продукта**.

16. Удалите блок: выберите блок **Изготовление вареников**, нажмите клавишу **Delete**, выберите **Да**.

17. Переименуйте стрелку **Заказы на пельмени** на **Заказы на изготовление**.

18. Переименуйте блок **Управление изготовлением и контролем качества** на диаграмме **A0 в Планирование производства**.

19. Создайте блок **Разработать рецепт**.

20. Ветвь стрелки **Персонал производственного цеха** свяжите с нижней гранью блока **Разработать рецепт**.

21. Введите метку созданной ветви **Кулинар**.

22. Создайте граничную стрелку выхода **Стандарты на продукцию** блока **Разработать рецепт**.

23. Туннелируйте созданную граничную стрелку **Стандарты на продукцию**.

24. Создайте ветвь стрелки **Стандарты на продукцию** и свяжите с верхней гранью блока **Планирование производства**.

25. Введите метку созданной ветви **Список необходимых ингредиентов**.

26. Удалите стрелку **Правила изготовления и контроля качества**.

27. Создайте ветвь стрелки **Стандарты на продукцию** и свяжите с верхней гранью блока **Изготовление продукта**.

28. Введите метку созданной ветви **Правила изготовления и контроля качества**.

29. Создайте граничную стрелку управления **Прогноз продаж** блока **Планирование производства**.

30. Создайте граничную стрелку управления **Информация от поставщика** блока **Планирование производства**.

31. Создайте граничную стрелку выхода **Заказ поставщику** блока **Планирование производства**.

32. Туннелируйте граничные стрелки.
33. Результат должен быть, как на диаграмме **Ж**.
34. На диаграмме **A0** туннелируйте стрелку **Готовая продукция**.
35. На диаграмме **A2** свяжите стрелку **Готовая продукция** с выходом функции **Изготовление продукта**.
36. На диаграмме **A2** туннелируйте стрелки **Информация от поставщика** и **Заказ поставщику**.
37. На диаграмме **A0** направьте стрелку **Прогноз продаж** с выхода **Продажи и маркетинг** на управление **Производство продукции**.
38. На диаграмме **A0** направьте стрелку **Стандарты на продукцию** с выхода **Производство продукции** на управление **Продажи и маркетинг**.
39. На диаграмме **A0** удалите ветвь стрелки управления **Правила и процедуры** блока **Производство продукции**.
40. Результат должен быть, как на диаграммах **З** и **И**.

6.5. Перенос блока

41. Перенесите блок **Разработать рецепт** с диаграммы **A2: Производство продукции** на диаграмму **A0**.
42. Удалите на диаграмме **A0** стрелку **Стандарты на продукцию**.
43. Создайте ветвь стрелки **Правила и процедуры** и свяжите с верхней гранью блока **Разработать рецепт**.
44. Создайте внутреннюю стрелку **Стандарты на продукцию** связи выход-управление между блоками **Разработать рецепт** и **Производство продукции**.
45. Создайте ветвь стрелки **Стандарты на продукцию** и свяжите с верхней гранью блока **Продажи и маркетинг**.
46. Результат должен быть, как на диаграммах **К** и **Л**.

6.6. Модификация BPMN

1. Запустите Modelio.
2. Откройте проект **Пельмени**.
3. Скопируйте модель **Изготовление проекта** в модель **Изготовление продукта**.
4. Измените модель **Изготовление продукта** по рис 38.

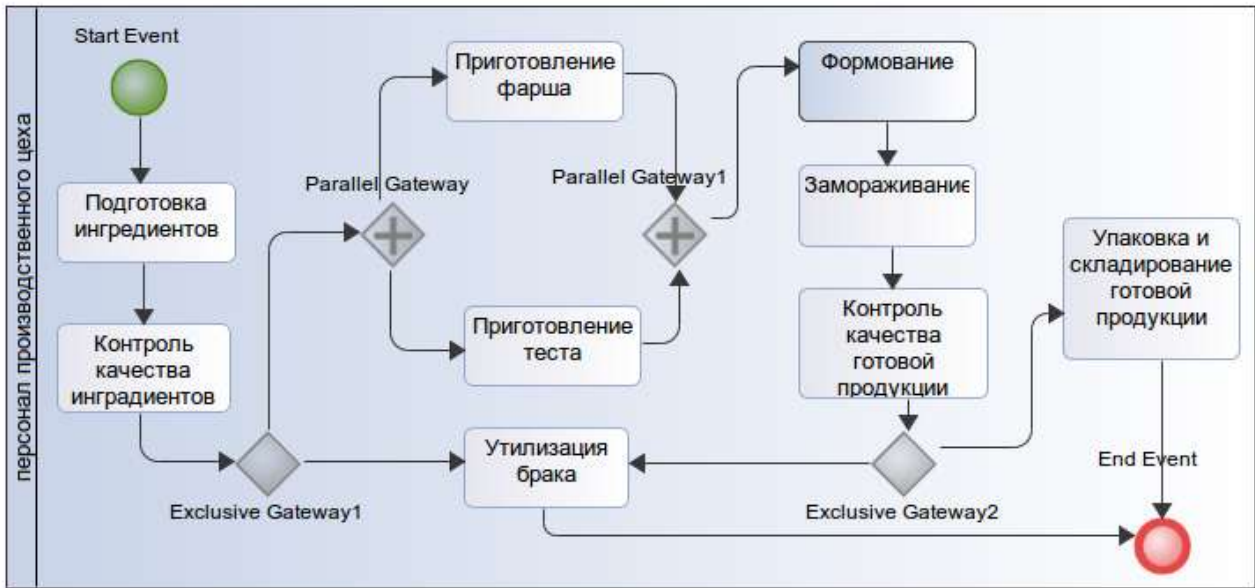


Рисунок 38 – Диаграмма BPMN модернизированная

7-й день. ДОПОЛНЕНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ДИАГРАММОЙ DFD

«...Больше ничего
Не выжмешь из рассказа моего».
А.С. Пушкин. Домик в Коломне, XL

7.1. Назначение

Диаграммы потоков данных (DFD) – это способ представления процессов обработки информации. Подобно IDEF0, DFD представляет систему как сеть процессов, связанных между собой с помощью стрелок.

В отличие от стрелок IDEF0, которые представляют собой жесткие взаимосвязи, стрелки DFD (потоки данных) показывают, как объекты (включая и данные) реально перемещаются от одной функции к другой. Это представление потока данных обеспечивает отражение в модели DFD таких физических характеристик системы, как движение объектов, хранение объектов, распространение объектов.

Диаграммы DFD обеспечивают удобный способ описания передаваемой информации как между частями моделируемой системы, так и между системой и внешним миром. Это качество определяет область применения DFD – они используются для создания моделей информационного обмена организации, например, модели документооборота.

7.2. Основные понятия

7.2.1. Процесс

В DFD процессы представляют собой функции системы, преобразующие входы в выходы. Процессы изображаются прямоугольниками со скругленными углами. Их смысл совпадает с блоком IDEF0. Они имеют входы и выходы, но не поддерживают управление и механизмы.

Каждый процесс должен быть именован глаголом с последующим дополнением. Кроме того, каждый процесс должен иметь уникальный номер.

Физически процесс может быть реализован различными способами: это может быть подразделение организации, выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство.

7.2.2. Поток данных

Потоки данных определяют информацию, передаваемую через некоторое соединение от источника к приемнику. Поток данных изображается линией, заканчивающейся стрелкой, которая показывает направление потока. Каждая стрелка имеет имя, отражающее его содержание.

Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными дискетами, переносимыми с одного компьютера на другой.

Поскольку в DFD каждая сторона блока не имеет четкого назначения, как в IDEF0, то стрелки могут подходить и выходить из любой грани блока. В DFD также применяется двунаправленные стрелки для описания диалогов типа команда-ответ.

7.2.3. Хранилище данных

В отличие от потока данных, описывающих объекты в движении, хранилища данных изображают объекты в покое. Хранилище представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в хранилище и через некоторое время извлечь, причем способы помещения могут быть любыми.

Имя хранилища должно идентифицировать его содержимое и быть существительным.

Хранилище физически может быть реализовано в виде ящика в картотеке, таблицы в оперативной памяти, файла на магнитном диске.

То есть хранилище в общем случае является прообразом будущей базы данных.

7.2.4. Внешняя сущность

Внешняя сущность представляет собой материальный объект или физическое лицо, представляющее собой источник или приемник информации (например, заказчик, персонал, поставщик, клиенты, склад). Ее имя должно содержать существительные. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы.

Внешняя сущность обозначается в виде прямоугольника с тенью и обычно располагается по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной диаграмме. Обычно такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

7.3. Построение диаграммы DFD

Диаграммы DFD могут быть построены с использованием структурного анализа, подобно тому, как строятся диаграммы IDEF0.

Сначала строится физическая модель, отображающая текущее состояние дел. Затем эта модель преобразуется в логическую модель, которая отображает требования к существующей системе. После этого строится модель отображающая требования к будущей системе. И, наконец, строится физическая модель, на основе которой должна быть построена новая система.

Другой подход, популярный при создании программного обеспечения, называется событийным разделением, в котором для моделирования системы строится несколько моделей DFD.

Создается логическая модель, отображающая систему как совокупность процессов и описывающая, что система должна делать. Затем строится модель окружения, которая описывает систему как объект, взаимодействующий с событиями из внешних сущностей. Эта модель содержит одну контекстную диаграмму и список событий.

Контекстная диаграмма содержит один процесс, изображающий систему в целом и внешние сущности, с которыми система взаимодействует. В конце создается модель поведения, показывающая, как система обрабатывает события. Эта модель состоит из одной диаграммы, в которой каждый блок изображает каждое событие из модели окружения. Хранилища могут быть добавлены для моделирования данных, которые необходимо запоминать между событиями. Потоки добавляются для связи с другими элементами.

7.4. Горячие слова

Диаграммы потоков данных (DFD, data flow diagramming) – это способ представления процессов обработки информации; элементы модели: процесс, поток, хранилище – представляющие обработку, передачу и хранение данных (или материальных объектов).

Внешняя сущность (external reference) – изображается прямоугольником с тенью – материальный объект или физическое лицо, представляющее собой источник или приемник информации, который находится за пределами границы системы.

Поток данных (data flow) – изображается стрелкой, описывает движение объектов (включая данные) от одного процесса к другому.

Процесс – изображается прямоугольником со скругленными углами, это функция обработки информации, преобразующая входные потоки в выходные; совпадает со смыслом блоков IDEF0; имеет входы и выходы, не поддерживает управление и механизмы.

Хранилище данных (data store) – описывает объекты (данные), которые необходимо сохранить в памяти прежде, чем использовать в процессах, т. е. изображает объекты в покое.

7.5. Декомпозиция блока

1. Сохраните модель: выберите **Файл, Сохранить как**, в поле **Имя файла** введите **Пельмени_DFD**, в поле **Тип** выберите ***.rsf**, выберите **Сохранить**.

2. Декомпозируйте блок **Продажи и маркетинг** на три блока: **Предоставление информации о ценах, Оформление заказов, Исследование рынка**.

3. Свяжите стрелки по диаграмме М.

7.6. Декомпозиция блока IDEF0 в диаграмму DFD

Согласно нотации DFD диаграмма не должна иметь граничных стрелок – все стрелки должны начинаться и заканчиваться на процессах, хранилищах данных или внешних сущностях.

4. Создайте диаграмму DFD: выберите диаграмму **A2: Продажи и маркетинг**, выберите блок **Оформление заказов**, выберите на палитре инструментов кнопку **Перейти к дочерней диаграмме**, в счетчике выберите 2 новых блока, выберите нотацию декомпозиции **IDEF0**, выберите шаблон **простой**, нажмите **ОК**.

5. Введите имена процессов: **Проверка и внесение клиента, Внесение заказа**.

6. Создайте классификатор: выберите **Окно, Показать окно, Классификаторы**, в окне **Классификаторы** нажмите кнопку **Создать элемент**, нажмите **F2**, введите **Список клиентов**.

7. Создайте классификаторы: **Список продуктов, Список заказов, Звонки клиентов**.

8. Создайте хранилище данных: выберите на палитре инструментов **Режим добавления хранилищ данных**, мышью выберите место размещения объекта, нажмите **МП**, выберите **Редактировать активный элемент**, выберите **Задать DFD-объект**, выберите **Список клиентов**, нажмите **ОК**, выберите **Цвет фона**, выберите цвет, нажмите **ОК**, выберите в палитре инструментов кнопку **Режим курсора**.

9. Создайте хранилища данных: **Список продуктов, Список заказов**.

10. Удалите все граничные стрелки с диаграммы DFD.

11. Создайте внешнюю сущность: выберите на палитре инструментов **Режим добавления внешних ссылок**, мышью выберите место размещения объекта, нажмите **МП**, выберите **Редактировать активный элемент**, выберите **Задать DFD-объект**, выберите **Звонки клиентов**, нажмите **ОК**, выберите **Цвет фона**, выберите цвет, нажмите **ОК**, выберите в палитре инструментов кнопку **Режим курсора**.

12. Создайте внутренние стрелки по диаграмме **H**.

13. На родительской диаграмме **A2** туннелируйте стрелки: **Звонки клиентов, Стандарты на продукцию, Правила и процедуры, Бухгалтерская система, Заказы клиентов** блока **Оформление заказов**.

14. Создайте диаграмму DFD: выберите диаграмму **A2: Продажи и маркетинг**, выберите блок **Исследование рынка**, выберите на палитре инструментов кнопку **Перейти к дочерней диаграмме**, в счетчике выберите 3 новых блока, выберите нотацию декомпозиции **IDEF0**, выберите шаблон **простой**, нажмите **ОК**.

15. Введите имена процессов: **Разработка прогнозов продаж, Разработка маркетинговых материалов, Привлечение новых клиентов**.

16. Создайте хранилища данных **Список клиентов, Список продуктов, Список заказов**.

17. Удалите все граничные стрелки с диаграммы DFD.

18. Добавьте внешние сущности **Маркетинговые материалы и Прогноз продаж**.

19. Создайте внутренние стрелки по диаграмме **О**.

20. На родительской диаграмме A2 туннелируйте стрелки.

ЗАКЛЮЧЕНИЕ

Рассмотренные в пособии методы использования и приемы работы с программными продуктами Ramus Education и Modelio являются базовыми для построения в соответствии с методологиями IDEF0, DFD, BPMN описаний моделируемых бизнес-процессов.

Построенная бизнес-модель может быть непосредственно подвергнута анализу, а моделируемый процесс в результате усовершенствован. Кроме своей основной цели – анализа и совершенствования деятельности организации – бизнес-моделирование несет дополнительные ценности (рис. 39). Эти ценности подробно описаны в главе «Гимн бизнес-моделированию» бестселлера [10].



Рисунок 39 – Значимость бизнес-моделирования

Благодаря созданию моделей бизнес-процессов в компании появляется:

- набор дополнительных регламентирующих документов, например, «Положение о финансовой структуре»;
- база знаний о накопленном опыте и наработанные технологии деятельности, которую должны использовать сотрудники, и особенно новые, как для обучения, так и для практического применения в повседневной деятельности;
- стандарт общения, облегчающий и ускоряющий процесс управления организацией;
- основа для проведения функционально-стоимостного анализа (АВС);

- основа для проведения динамического моделирования;
- основа для проведения бенчмаркинга (benchmarking);
- возможность составления функциональных требований и внедрения корпоративной информационной системы;
- основа для внедрения системы качества;
- возможность поставить процесс изменений деятельности компании на технологическую основу.

Пособие формирует следующие компетенции для дисциплины «Моделирование бизнес-процессов»:

– по направлению 09.03.03: ОПК-1 – способность использовать нормативно-правовые документы, международные и отечественные стандарты в области ИС и технологий; ПК-1 – способность проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к ИС; ПК-6 – способность собирать детальную информацию для формализации требований пользователей заказчика; ПК-7 – способность проводить описание прикладных процессов и информационного обеспечения решения прикладных задач

– по направлению 38.03.05: ОК-5 – способность использовать нормативные правовые документы в своей деятельности; ПК-1 – проводить анализ архитектуры предприятия; ПК-5 – проводить обследование деятельности и ИТ-инфраструктуры предприятий; ПК-8 – использовать современные стандарты и методики, разрабатывать регламенты деятельности предприятия; ПК-22 – консультировать заказчиков по совершенствованию бизнес-процессов и ИТ-инфраструктуры предприятия.

ЛИТЕРАТУРА

1. Грабауров, В.А. Информационные технологии для менеджеров / В.А. Грабауров. – М.: Финансы и статистика, 2001. – 368 с.
2. Использование конструкций BPMN 2.0. [Электронный ресурс] Berliner BPM-Offensive. – Режим доступа – URL: http://www.bpmb.de/images/BPMN2_0_Poster_RU.pdf (дата обращения: 21.02.16)
3. Краткий словарь терминов современного менеджмента. [Электронный ресурс] БИГ-Петербург. – Режим доступа – URL: http://big.spb.ru/publications/glossary/part3_model_and_reorg_proc.shtml (дата обращения: 21.02.16)
4. Кулябов, Д.С. Введение в формальные методы описания бизнес-процессов: учеб. пособие / Д.С. Кулябов, А.В. Королькова. – М.: РУДН, 2008. – 173 с.
5. Марка, Дэвид А. Методология структурного анализа и проектирования SADT / Дэвид А. Марк, Клемент Мак-Гоуэн. – М.: Метатехнология, 1993.
6. Миндалёв, И.В. Моделирование бизнес-процессов с помощью CASE-средства VPwin за 7 дней: учеб. пособие / И.В. Миндалёв; Краснояр. гос. аграр. ун-т. – Красноярск, 2002. – 120 с.
7. Нотация BPMN 2.0 на русском языке. [Электронный ресурс] ELMA. – Режим доступа – URL: <http://www.elma-bpm.ru/bpmn2/> (дата обращения: 21.02.16)
8. Регионы России. Социально-экономические показатели – 2014. [Электронный ресурс] Федеральная служба государственной статистики. – Режим доступа – URL: http://www.gks.ru/bgd/regl/b14_14p/isswww.exe/stg/d03/19-07.htm (дата обращения: 21.02.16)
9. Р50.1.028-2001. Методология функционального моделирования. М.: Госстандарт России, 2000. [Электронный ресурс] НИЦ «Прикладная Логистика». – Режим доступа – URL: http://cals.ru/sites/default/files/downloads/ndocs/r_50.1.028-2001.pdf (дата обращения: 21.02.16)
10. 7 нот менеджмента. Настольная книга руководителя / под ред. В.В. Кондратьева. – 7-е изд., перераб. и доп. – М.: Эксмо, 2008. – 976 с.
11. Черемных, С.В. Структурный анализ систем: IDEF-технологии / С.В. Черемных, И.О. Семенов, В.С. Ручкин. – М.: Финансы и статистика, 2001. – 208 с.

12. Documents Associated with Business Process Model and Notation (BPMN). [Электронный ресурс] The Object Management Group (OMG). – Режим доступа – URL: <http://www.omg.org/spec/BPMN/2.0/PDF> (дата обращения: 21.02.16)

13. Downloads. [Электронный ресурс] Modelio. – Режим доступа – URL: <https://www.modelio.org/downloads/download-modelio.html> (дата обращения: 21.02.16)

14. Howto install proprietary Java from Oracle. [Электронный ресурс] Rosa wiki. – Режим доступа – URL: http://wiki.rosalab.ru/en/index.php/Howto_install_proprietary_Java_from_Oracle#JRE_7u6_64bit (дата обращения: 21.02.16)

15. Install Oracle Java JDK/JRE 8u45 on Fedora 22/21, CentOS/RHEL 7.1/6.6/5.11. [Электронный ресурс] Most Popular Linux Install Guides. – Режим доступа – URL: <http://www.if-not-true-then-false.com/2014/install-oracle-java-8-on-fedora-centos-rhel/> (дата обращения: 21.02.16)

16. Integration Definition For Function Modeling (IDEF0). Draft Federal Information Processing Standards Publication 183, 1993, December 2. [Электронный ресурс] IDEF Family of Methods. – Режим доступа – URL: <http://idef.com/pdf/idef0.pdf> (дата обращения: 21.02.16)

17. Java SE 6 Downloads. [Электронный ресурс] Oracle Technology Network. – Режим доступа – URL: <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR> (дата обращения: 21.02.16)

18. Java SE Development Kit 8 Downloads. [Электронный ресурс] Oracle Technology Network. – Режим доступа – URL: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> (дата обращения: 21.02.16)

19. JDK Self-Extracting Binary Installation for Linux (32-bit). [Электронный ресурс] Oracle Technology Network. – Режим доступа – URL: <http://java.sun.com/javase/6/webnotes/install/jdk/install-linux-self-extracting.html> (дата обращения: 21.02.16)

20. List of Unified Modeling Language tool. [Электронный ресурс] Wikipedia. – Режим доступа – URL: https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tool (дата обращения: 21.02.16)

21. Ramus Educational [Электронный ресурс] Официальный русскоязычный сайт проекта Ramus! – Режим доступа – URL: http://ramussoftware.com/index.php?option=com_content&view=article&id=10&Itemid=16 (дата обращения: 21.02.16)

22. Ramus Educational 1.1.1. [Электронный ресурс] Ramus. – Режим доступа – URL: http://ramussoftware.com/index.php?option=com_docman&task=cat_view&gid=15&Itemid=10 (дата обращения: 21.02.16).

23. Ramus Educational Version: 1.1.1. Ramus Help.

24. Requirements. [Электронный ресурс] Modelio. – Режим доступа – URL: <https://www.modelio.org/documentation/installation/12-installation.html> (дата обращения: 21.02.16)

ПРИЛОЖЕНИЯ

Приложение 1

Перечень типовых бизнес-процессов предприятия (Структура классификации процессов*)

Структура классификации процессов (Process Classification Framework) разработана Американским центром производительности и качества (American Productivity & Quality Center).

* – Copyright American Productivity & Quality Center

1. Анализировать рынок и потребности потребителей

1.1. Определять потребности и пожелания потребителей

1.1.1. Выполнять качественные оценки

1.1.1.1. Проводить интервьюирование потребителей

1.1.1.2. Проводить анализ фокус-групп

1.1.2. Выполнять количественные оценки

1.1.2.1. Подготовка и проведение инспекций

1.1.3. Прогнозировать покупательский спрос потребителей

1.2. Измерять удовлетворение потребителей

1.2.1. Осуществлять мониторинг удовлетворенности продуктами и услугами

1.2.2. Осуществлять мониторинг удовлетворения потребителей при разрешении жалоб

1.2.3. Осуществлять мониторинг удовлетворенности потребителей от общения

1.3. Осуществлять мониторинг изменений на рынке или в ожиданиях потребителей

1.3.1. Определять слабые стороны в предложении продуктов/услуг

1.3.2. Идентифицировать новые инновации, которые обеспечивают потребности потребителей

1.3.3. Определять реакцию потребителей на конкурирующие предложения

2. Разрабатывать видение и стратегию

2.1. Осуществлять мониторинг внешней среды

2.1.1. Анализировать и выявлять причины конкуренции

2.1.2. Определять экономические тренды

- 2.1.3. Идентифицировать политические и правовые вопросы
- 2.1.4. Оценивать новые технологические инновации
- 2.1.5. Анализировать демографию
- 2.1.6. Идентифицировать социальные и культурные изменения
- 2.1.7. Анализировать экологические проблемы
- 2.2. Определять концепцию бизнеса и стратегию организации
 - 2.2.1. Выбирать релевантные рынки
 - 2.2.2. Определять долгосрочное видение
 - 2.2.3. Формулировать стратегию бизнес-единиц
 - 2.2.4. Разрабатывать всеобщую формулировку миссии
- 2.3. Разрабатывать организационную структуру и систему взаимоотношений между организационными единицами
- 2.4. Разрабатывать и ранжировать цели организации
- 3. Разрабатывать продукты или услуги**
 - 3.1. Разрабатывать концепцию и план продукта/услуги
 - 3.1.1. Переводить потребности и желания потребителя в требования к продукту/услуге
 - 3.1.2. Планировать и детализировать цели по качеству
 - 3.1.3. Планировать и детализировать цели по стоимости
 - 3.1.4. Разрабатывать жизненный цикл продукта и определять цели по времени
 - 3.1.5. Разрабатывать и интегрировать лидирующие технологии в концепцию продукта/услуги
 - 3.2. Разрабатывать, создавать и оценивать прототипы продуктов и услуг
 - 3.2.1. Разрабатывать спецификации продуктов/услуг
 - 3.2.2. Осуществлять параллельное проектирование
 - 3.2.3. Осуществлять расчет стоимости
 - 3.2.4. Документировать спецификацию конструкции
 - 3.2.5. Разрабатывать прототипы
 - 3.2.6. Получать патенты
 - 3.3. Совершенствовать существующие продукты/услуги
 - 3.3.1. Разрабатывать расширения (улучшения потребительских качеств) продукта/услуги
 - 3.3.2. Устранять проблемы качества и надежности
 - 3.3.3. Устранять устаревшие продукты/услуги
 - 3.4. Тестировать эффективность новых или измененных продуктов или услуг
 - 3.4.1. Осуществлять подготовку к производству

3.4.2. Разрабатывать и тестировать процесс производства прототипа

3.4.3. Разрабатывать и обеспечивать необходимыми материалами и оборудованием

3.4.4. Внедрять и проверять процесс или методологию

3.5. Управлять процессом разработки продукта/услуги

4. Продавать продукты/услуги

4.1. Позиционировать продукты и услуги на сегментах потребительского рынка

4.1.1. Разрабатывать ценовую стратегию

4.1.2. Разрабатывать рекламную стратегию

4.1.3. Разрабатывать маркетинговые слоганы

4.1.4. Оценивать возможность рекламы и требования по ее финансированию

4.1.5. Идентифицировать выделенных (особенных) целевых потребителей и их потребности

4.1.6. Разрабатывать прогноз продаж

4.1.7. Продавать продукты и услуги

4.1.8. Вести переговоры об условиях поставки

4.2. Обрабатывать заказы потребителей

4.2.1. Получить заказы от потребителей

4.2.2. Включать заказы в процессы производства и доставки

5. Производить и обеспечивать производство

5.1. Планировать и получать необходимые ресурсы

5.1.1. Выбирать и сертифицировать поставщиков

5.1.2. Приобретать основные средства

5.1.3. Приобретать материалы и комплектующие

5.1.4. Приобретать подходящие технологии

5.2. Преобразовывать ресурсы или входы в продукты

5.2.1. Разрабатывать и настраивать процесс производства (для существующего процесса)

5.2.2. Разрабатывать график производства

5.2.3. Перемещать материалы или ресурсы

5.2.4. Изготавливать продукт

5.2.5. Упаковывать продукт

5.2.6. Складевать или хранить продукт

5.2.7. Подготавливать продукт к поставке

5.3. Поставлять продукт

5.3.1. Планировать поставку продукта

5.3.2. Поставлять продукт потребителю

5.3.3. Устанавливать продукт

5.3.4. Подтверждать специальные требования по обслуживанию потребителя

5.3.5. Идентифицировать и планировать ресурсы для удовлетворения требований по обслуживанию

5.3.6. Обеспечивать обслуживание специальных клиентов

5.4. Управлять процессом производства и поставки

5.4.1. Документировать и осуществлять мониторинг статуса заказов

5.4.2. Управлять запасами

5.4.3. Обеспечивать качество продукта

5.4.4. Планировать и выполнять текущий ремонт

5.4.5. Осуществлять мониторинг внешних ограничений

6. Производство и поставка для организаций, ориентированных на сервис

6.1. Планировать и получать необходимые ресурсы

6.1.1. Выбирать и сертифицировать поставщиков

6.1.2. Приобретать материалы и комплектующие

6.1.3. Приобретать подходящие технологии

6.2. Разрабатывать требования к квалификации персонала

6.2.1. Определять требования по квалификации персонала

6.2.2. Идентифицировать и проводить тренинги

6.2.3. Осуществлять мониторинг и управление повышением квалификации

6.3. Оказывать услугу потребителю

6.3.1. Подтверждать специальные требования по обслуживанию конкретного потребителя

6.3.2. Идентифицировать и планировать ресурсы для удовлетворения требований по обслуживанию

6.3.3. Обеспечивать обслуживание специальных клиентов

6.4. Обеспечивать качество обслуживания

7. Выставлять счет и обслуживать потребителей

7.1. Выставлять счета потребителям

7.1.1. Разрабатывать, выставять и поддерживать деятельность по выставлению счетов

7.1.2. Выставлять счета потребителю

7.1.3. Откликаться на запросы по выставлению счетов

7.2. Оказывать послепродажное обслуживание

- 7.2.1. Оказывать послепродажное обслуживание
- 7.2.2. Осуществлять гарантийное обслуживание и претензионную работу
- 7.3. Откликаться на запросы потребителя
 - 7.3.1. Откликаться на информационные запросы
 - 7.3.2. Работать с жалобами потребителей
- 8. Управлять человеческими ресурсами**
 - 8.1. Разрабатывать и управлять стратегиями в области человеческих ресурсов
 - 8.1.1. Идентифицировать требования организации в стратегическом плане
 - 8.1.2. Определять затраты на человеческие ресурсы
 - 8.1.3. Определять требования к человеческим ресурсам
 - 8.1.4. Определять роль организационной структуры
 - 8.2. Детализировать стратегию до уровня функций
 - 8.2.1. Анализировать, разрабатывать и реорганизовывать функции
 - 8.2.2. Определять и систематизировать выходы функций и метрики (показатели)
 - 8.2.3. Определять сферы ответственности за выполнение функций
 - 8.3. Управлять приемом персонала
 - 8.3.1. Планировать и прогнозировать потребность в рабочей силе
 - 8.3.2. Разрабатывать планы продвижения и карьеры
 - 8.3.3. Осуществлять поиск, подбор и прием персонала
 - 8.3.4. Формировать и развертывать команды
 - 8.3.5. Перемещать служащих
 - 8.3.6. Реорганизовывать и сокращать персонал
 - 8.3.7. Управлять увольнением персонала
 - 8.3.8. Обеспечивать трудоустройство увольняемого персонала
 - 8.4. Развивать и обучать персонал
 - 8.4.1. Приводить в соответствие квалификацию персонала и требования по развитию организации
 - 8.4.2. Разрабатывать и управлять программами обучения
 - 8.4.3. Разрабатывать и управлять программами профориентации персонала
 - 8.4.4. Разрабатывать сферы ответственности за выполнение функций/процессов
 - 8.4.5. Разрабатывать сферы ответственности управляющих
 - 8.4.6. Разрабатывать сферы ответственности команд

8.5. Управлять производительностью, осуществлять материальное и моральное стимулирование

8.5.1. Определять показатели производительности

8.5.2. Разрабатывать подходы к управлению производительностью и обратной связью

8.5.3. Управлять производительностью команд

8.5.4. Оценивать ценность функции с точки зрения создания добавленной стоимости внутренней стоимости

8.5.5. Разрабатывать и управлять постоянной и переменной частью зарплаты

8.5.6. Управлять программами материального и морального стимулирования

8.6. Обеспечивать здоровье и удовлетворенность персонала

8.6.1. Управлять удовлетворенностью персонала

8.6.2. Разрабатывать системы поддержки работы и семьи

8.6.3. Управлять и администрировать

9. Управлять информационными ресурсами

9.1. Планировать управление информационными ресурсами

9.1.1. Определять требования на основе стратегий бизнеса

9.1.2. Определять архитектуру систем предприятия

9.1.3. Планировать и прогнозировать информационные технологии и методологии

9.1.4. Устанавливать стандарты данных предприятия

9.1.5. Устанавливать стандарты качества и контроля

9.2. Разрабатывать и развертывать системы поддержки предприятия

9.2.1. Проводить оценку специфических потребностей

9.2.2. Выбирать информационные технологии

9.2.3. Определять жизненные циклы данных

9.2.4. Разрабатывать системы поддержки предприятия

9.2.5. Тестировать, оценивать и развертывать системную безопасность и контроль

9.3. Внедрять системную безопасность и контроль

9.3.1. Устанавливать стратегии системной безопасности и уровни безопасности

9.3.2. Тестировать, оценивать и развертывать системную безопасность и контроль

9.4. Управлять хранением и поиском данных

9.4.1. Устанавливать базы данных

9.4.2. Собирать и упорядочивать информацию

- 9.4.3. Хранить информацию
- 9.4.4. Изменять и обновлять информацию
- 9.4.5. Осуществлять возможность поиска информации
- 9.4.6. Уничтожать информацию
- 9.5. Управлять оборудованием и сетевыми операциями
 - 9.5.1. Управлять централизованным оборудованием
 - 9.5.2. Управлять распределенным оборудованием
 - 9.5.3. Управлять сетевыми операциями
- 9.6. Управлять информационными услугами
 - 9.6.1. Управлять библиотеками и центрами информации
 - 9.6.2. Управлять документированием и фиксированием данных

бизнеса

- 9.7. Обеспечивать распределенный доступ к информации и

коммуникациям

- 9.7.1. Управлять внешними коммуникационными системами
- 9.7.2. Управлять внутренними коммуникационными системами
- 9.7.3. Подготавливать и распространять публикации
- 9.8. Оценивать и проводить аудит качества информации

10. Управлять финансовыми и материальными ресурсами

- 10.1. Управлять финансовыми ресурсами
 - 10.1.1. Разрабатывать бюджеты
 - 10.1.2. Управлять распределением ресурсов
 - 10.1.3. Определять структуру капитала
 - 10.1.4. Управлять потоками денежных средств
 - 10.1.5. Управлять финансовыми рисками
- 10.2. Осуществлять финансовые и учетные операции (транзакции)
 - 10.2.1. Работать с дебиторской задолженностью
 - 10.2.2. Выполнять оплату труда персонала
 - 10.2.3. Работать с кредиторской задолженностью, кредитами и

инкассо

- 10.2.4. Вести бухгалтерский учет
- 10.2.5. Выплачивать премии и пособия
- 10.2.6. Управлять общехозяйственными и представительскими

расходами

- 10.3. Формировать отчеты
 - 10.3.1. Обеспечивать внешней финансовой информацией
 - 10.3.2. Обеспечивать внутренней финансовой информацией
- 10.4. Проводить внутренний аудит
- 10.5. Управлять налогами

- 10.5.1. Обеспечивать соответствие законодательству
- 10.5.2. Планировать налоговую стратегию
- 10.5.3. Выбирать эффективные технологии
- 10.5.4. Управлять налоговыми спорами
- 10.5.5. Информировать менеджмент компании о налогах
- 10.5.6. Управлять администрированием налогов
- 10.6. Управлять материальными ресурсами
- 10.6.1. Управлять планированием капитала
- 10.6.2. Приобретать и продавать основные средства
- 10.6.3. Управлять оборудованием
- 10.6.4. Управлять материальными рисками

11. Выполнять управление программой работы с окружающей средой

- 11.1.1. Разрабатывать стратегию управления окружающей средой
- 11.1.2. Обеспечивать соответствие законодательству
- 11.1.3. Обучать персонал и проводить тренинги
- 11.1.4. Внедрять программы по предупреждению загрязнения окружающей среды
- 11.1.5. Управлять восстановительными работами
- 11.1.6. Внедрять программ реагирования на угрозы
- 11.1.7. Управлять связями с государственными агентствами и PR
- 11.1.8. Разрабатывать и управлять информационной системой окружающей среды
- 11.1.9. Осуществлять мониторинг программы управления окружающей средой

12. Управлять внешними связями

- 12.1. Обеспечивать обмен информацией с владельцами предприятия
- 12.2. Управлять отношениями с правительством
- 12.3. Строить взаимоотношения с кредиторами
- 12.4. Разрабатывать программу PR
- 12.5. Взаимодействовать с советом директоров
- 12.6. Разрабатывать взаимоотношения с обществом
- 12.7. Управлять правовыми и этическими вопросами

13. Управлять улучшениями и изменениями

- 13.1. Измерять показатели деятельности организации
- 13.1.1. Создавать систему измерения показателей
- 13.1.2. Измерять качество продуктов и услуг
- 13.1.3. Измерять затраты на обеспечение качества
- 13.1.4. Измерять затраты

- 13.1.5. Измерять длительность циклов
- 13.1.6. Измерять производительность (продуктивность)
- 13.2. Осуществлять оценки качества
 - 13.2.1. Осуществлять оценки качества на основе внешних критериев
 - 13.2.2. Осуществлять оценки качества на основе внутренних критериев
- 13.3. Осуществлять сравнительный анализ деятельности
 - 13.3.1. Определять возможности проведения сравнительного анализа
 - 13.3.2. Осуществлять сравнительный анализ бизнес-процессов
 - 13.3.3. Осуществлять сравнительный анализ конкурентных преимуществ
- 13.4. Улучшать процессы и системы
 - 13.4.1. Определять направления улучшений
 - 13.4.2. Внедрять непрерывное улучшение бизнес-процессов
 - 13.4.3. Проводить реорганизацию бизнес-процессов и систем
 - 13.4.4. Управлять улучшениями
- 13.5. Внедрять TQM
 - 13.5.1. Определять направления TQM
 - 13.5.2. Разрабатывать и внедрять системы TQM
 - 13.5.3. Управлять жизненным циклом TQM

© В.В. Репин, 2001 г.

Приложение 2

Диаграммы функциональной модели предприятия по производству пельменей

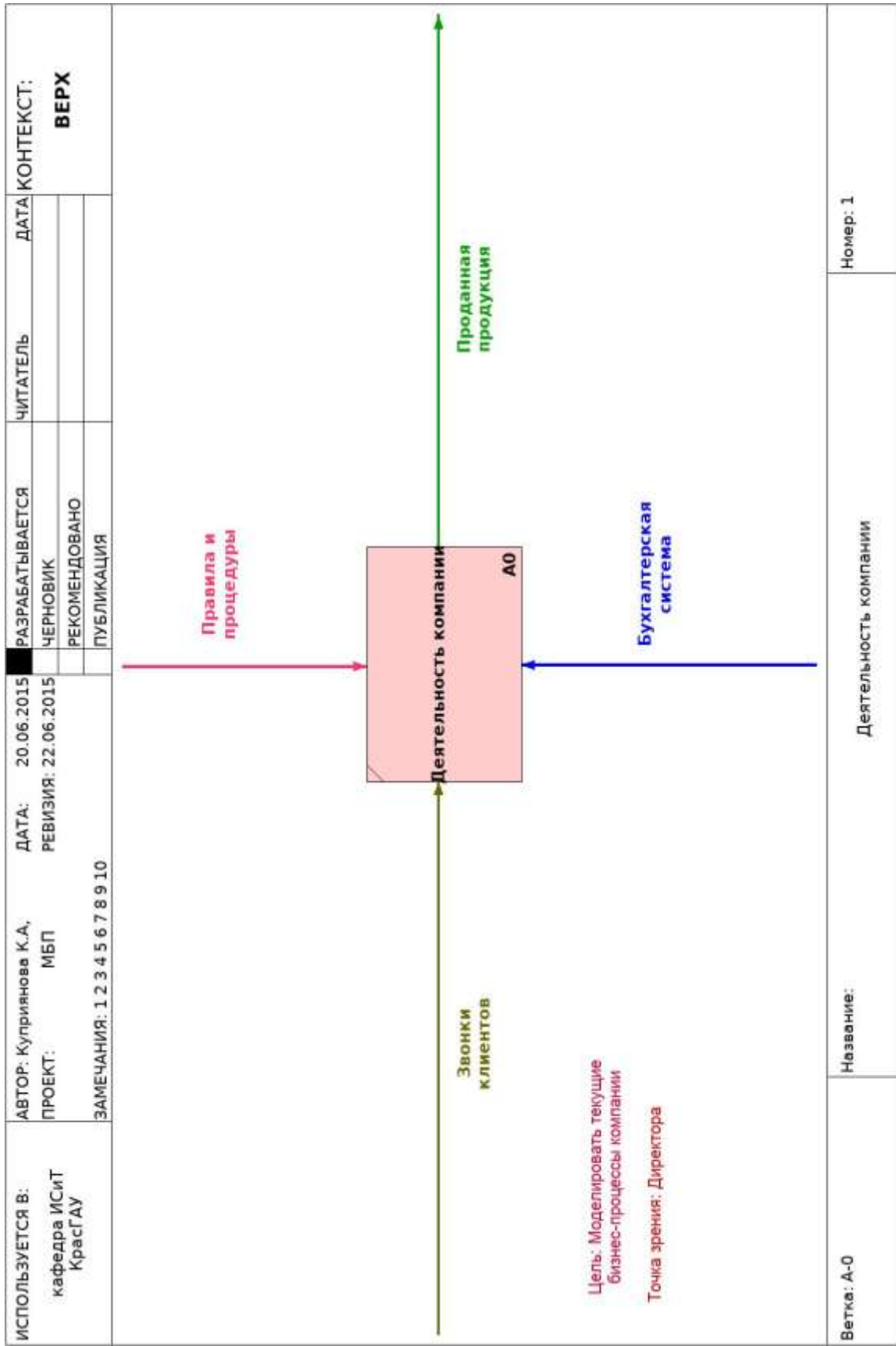


Диаграмма А

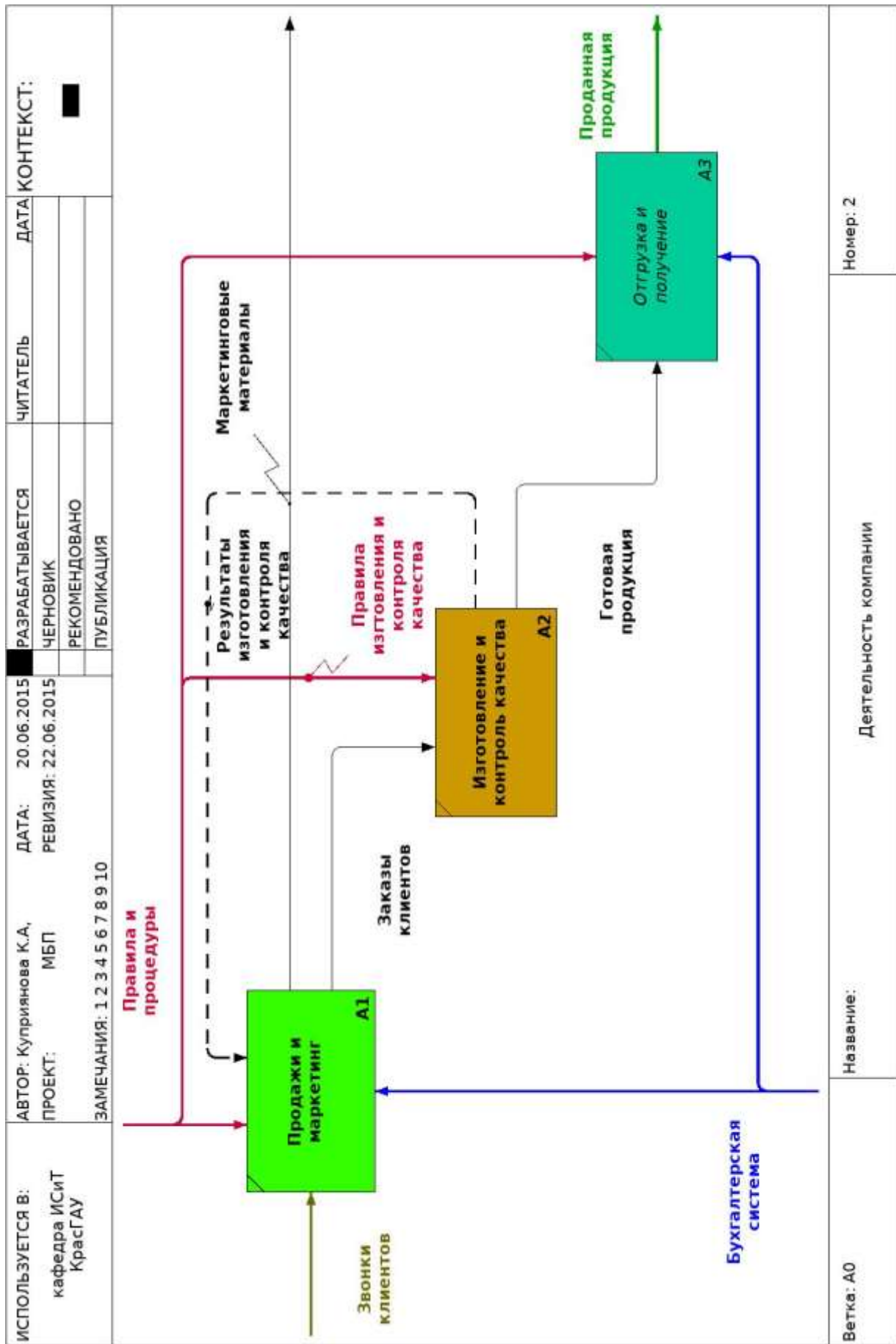
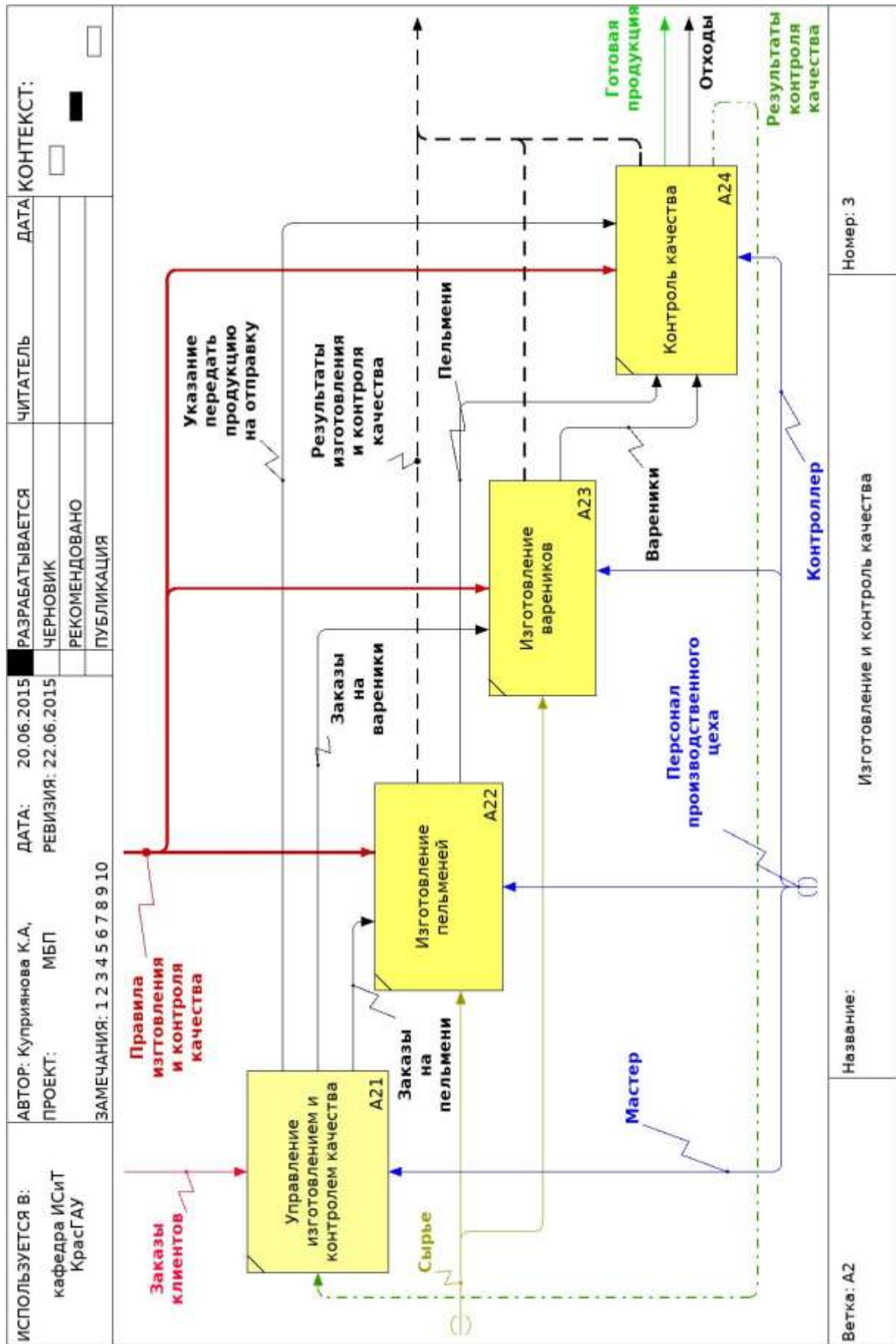


Диаграмма Б



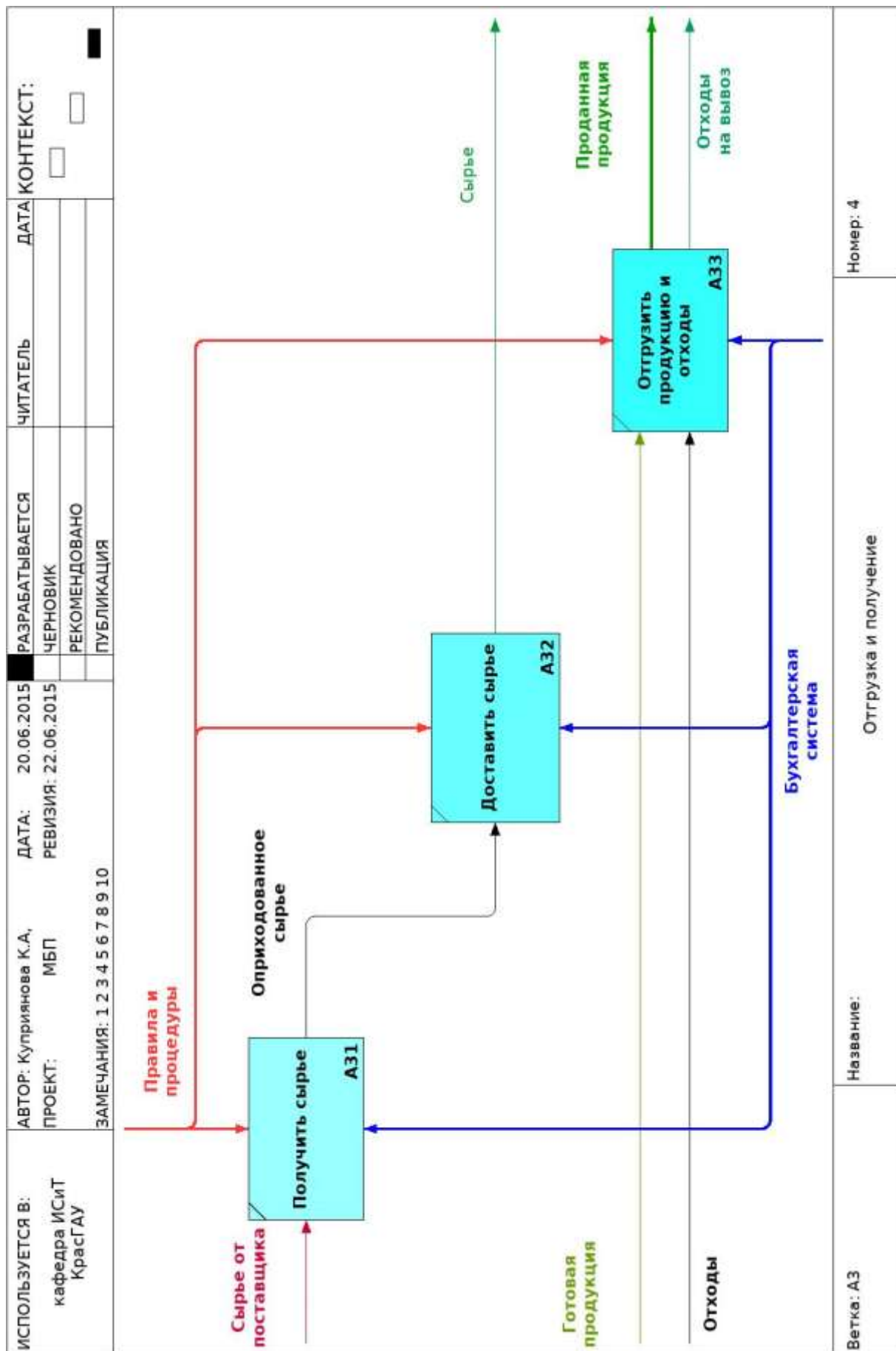
Ветка: A2

Название:

Изготовление и контроль качества

Номер: 3

Диаграмма В



Ветка: А3

Название:

Отгрузка и получение

Номер: 4

Диаграмма Г

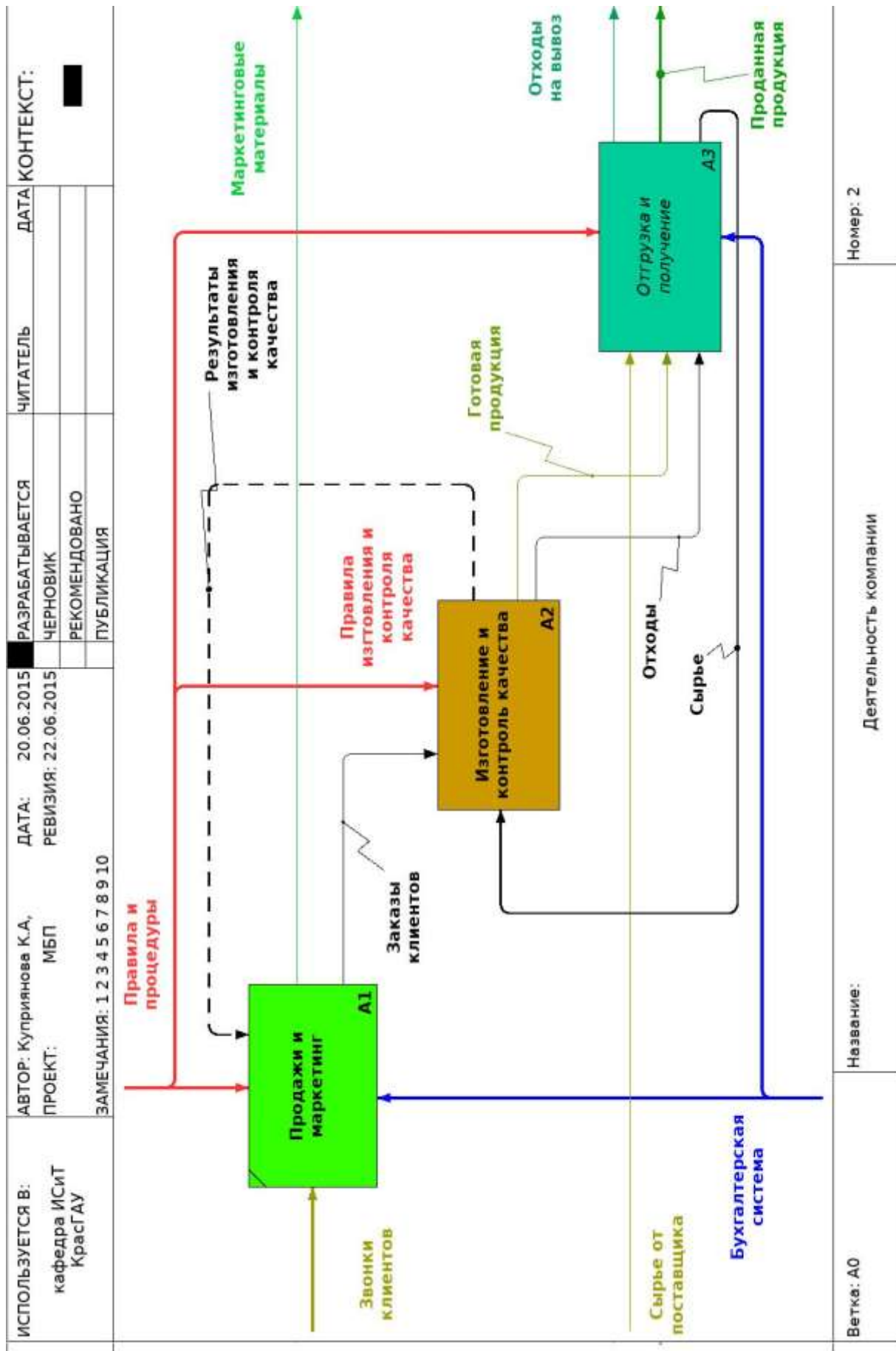


Диаграмма Д

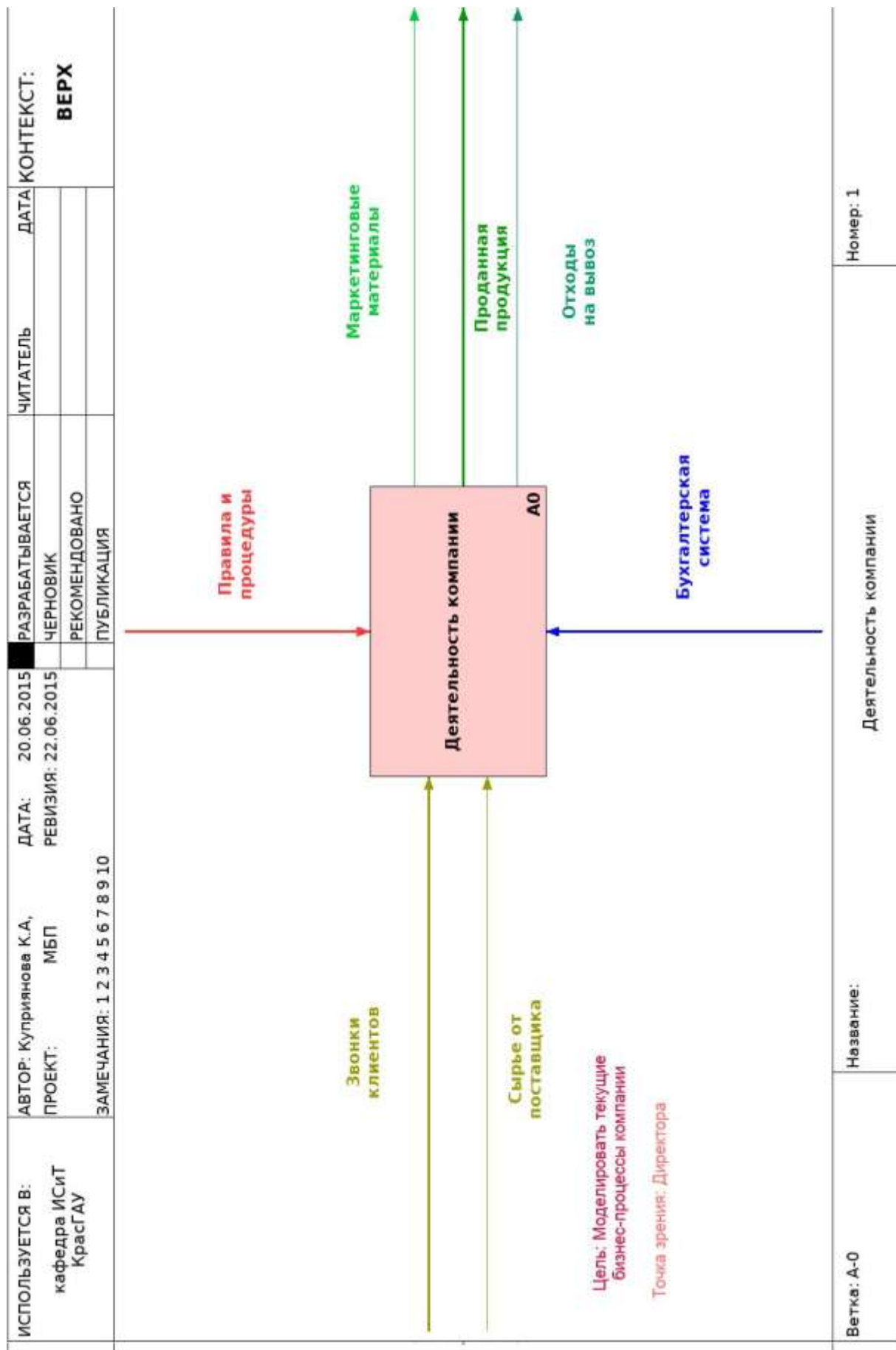
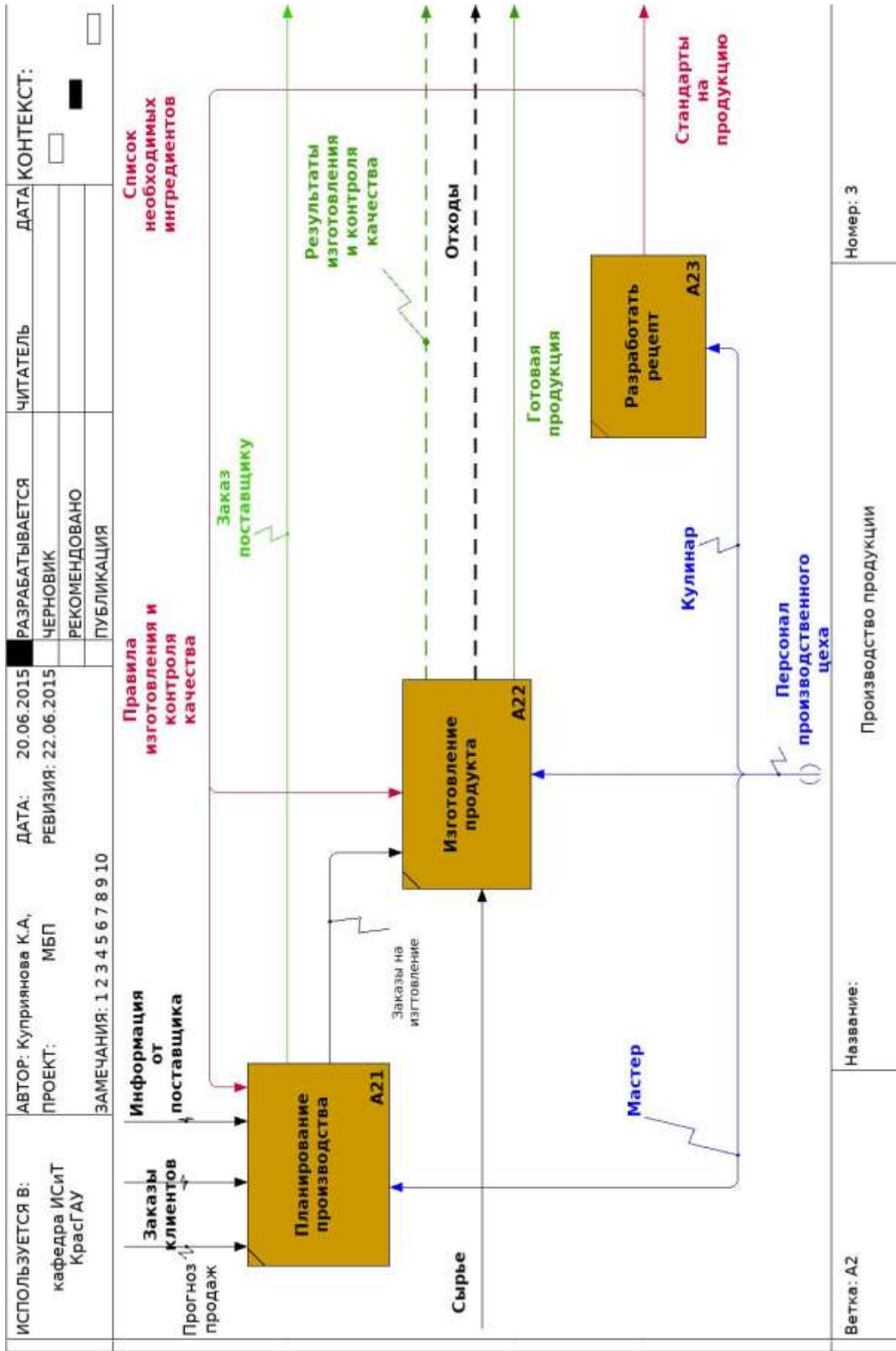


Диаграмма Е



Номер: 3

Производство продукции

Название:

Ветка: A2

Диаграмма Ж

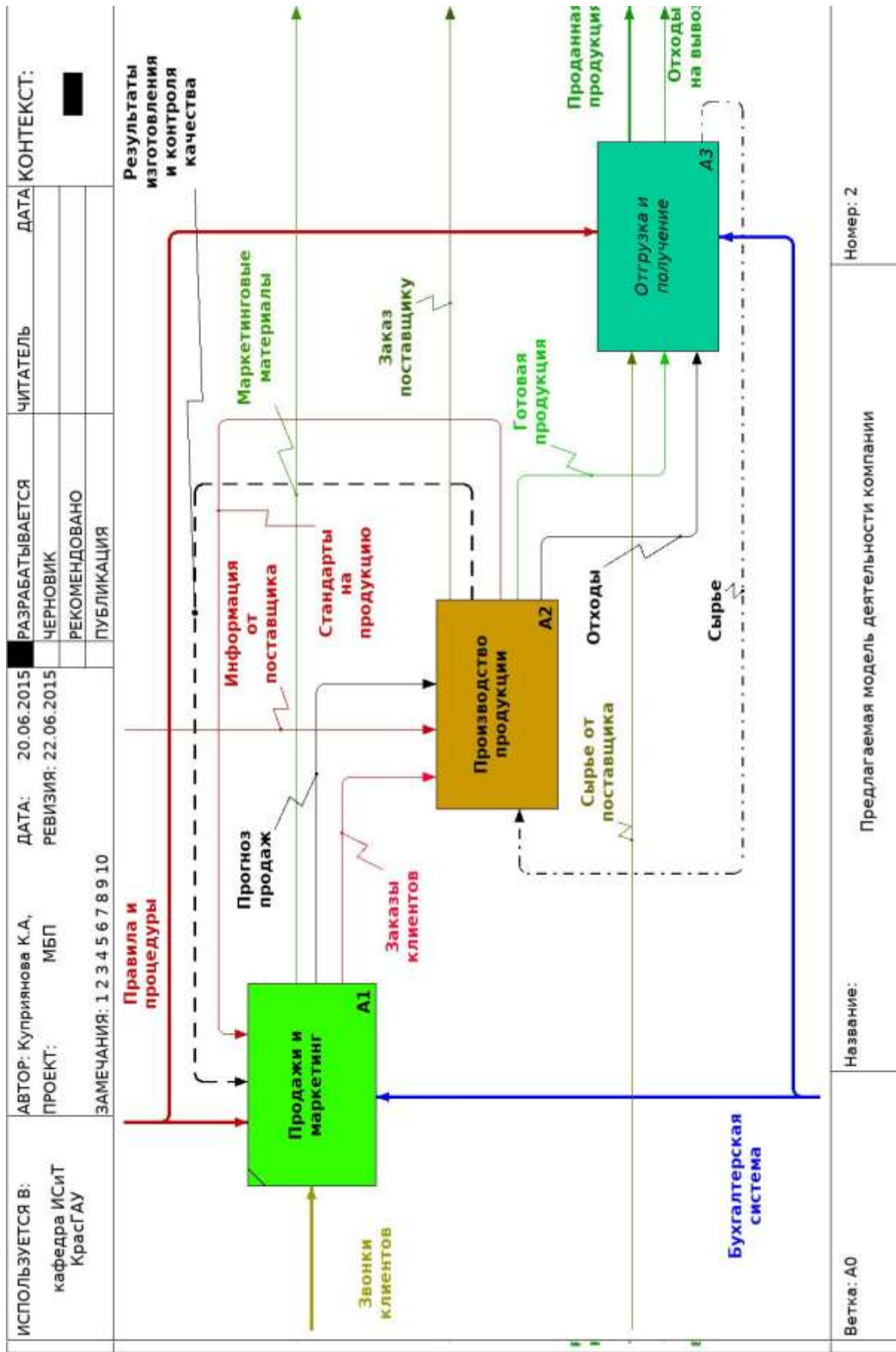


Диаграмма 3

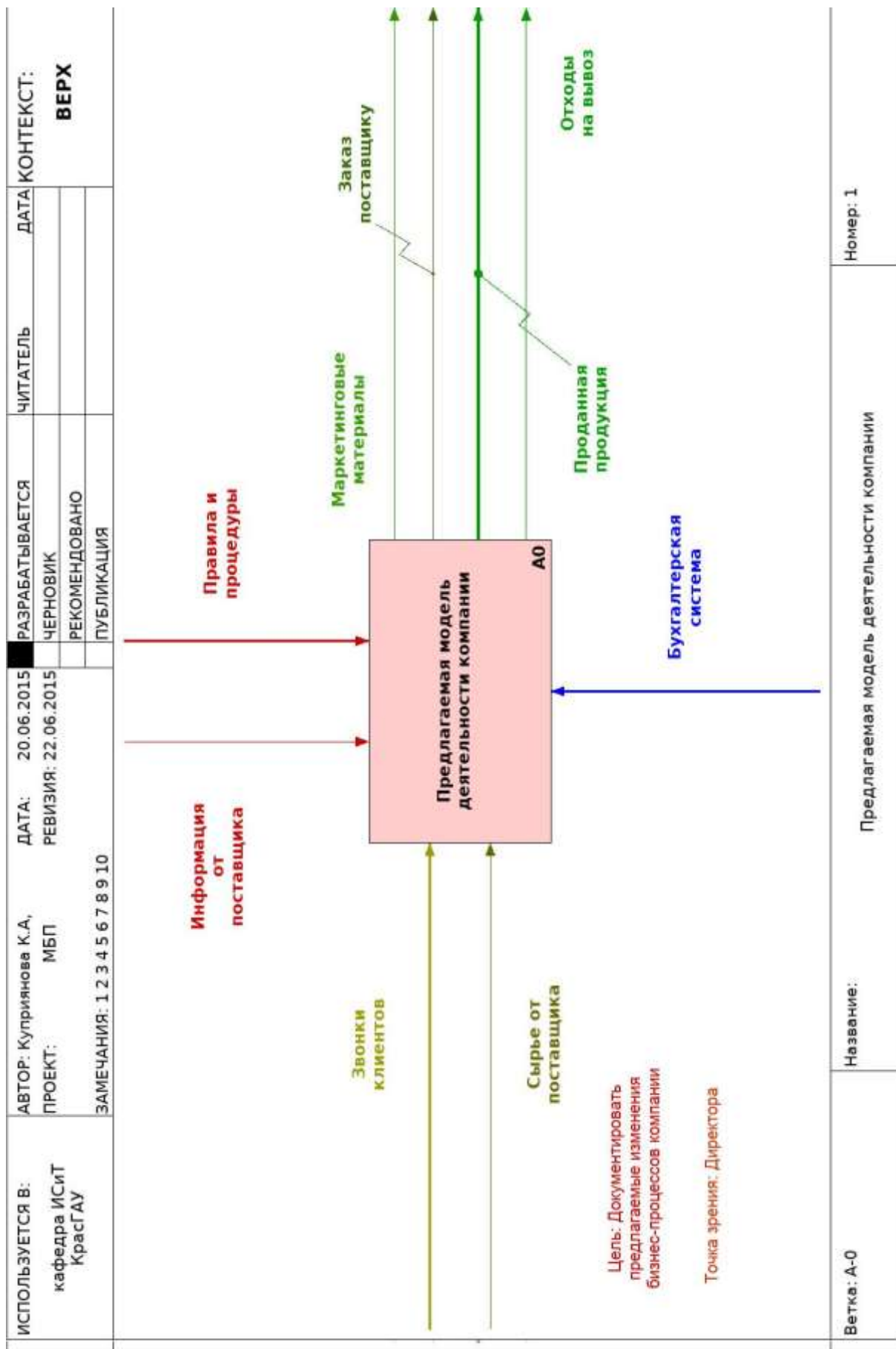


Диаграмма И

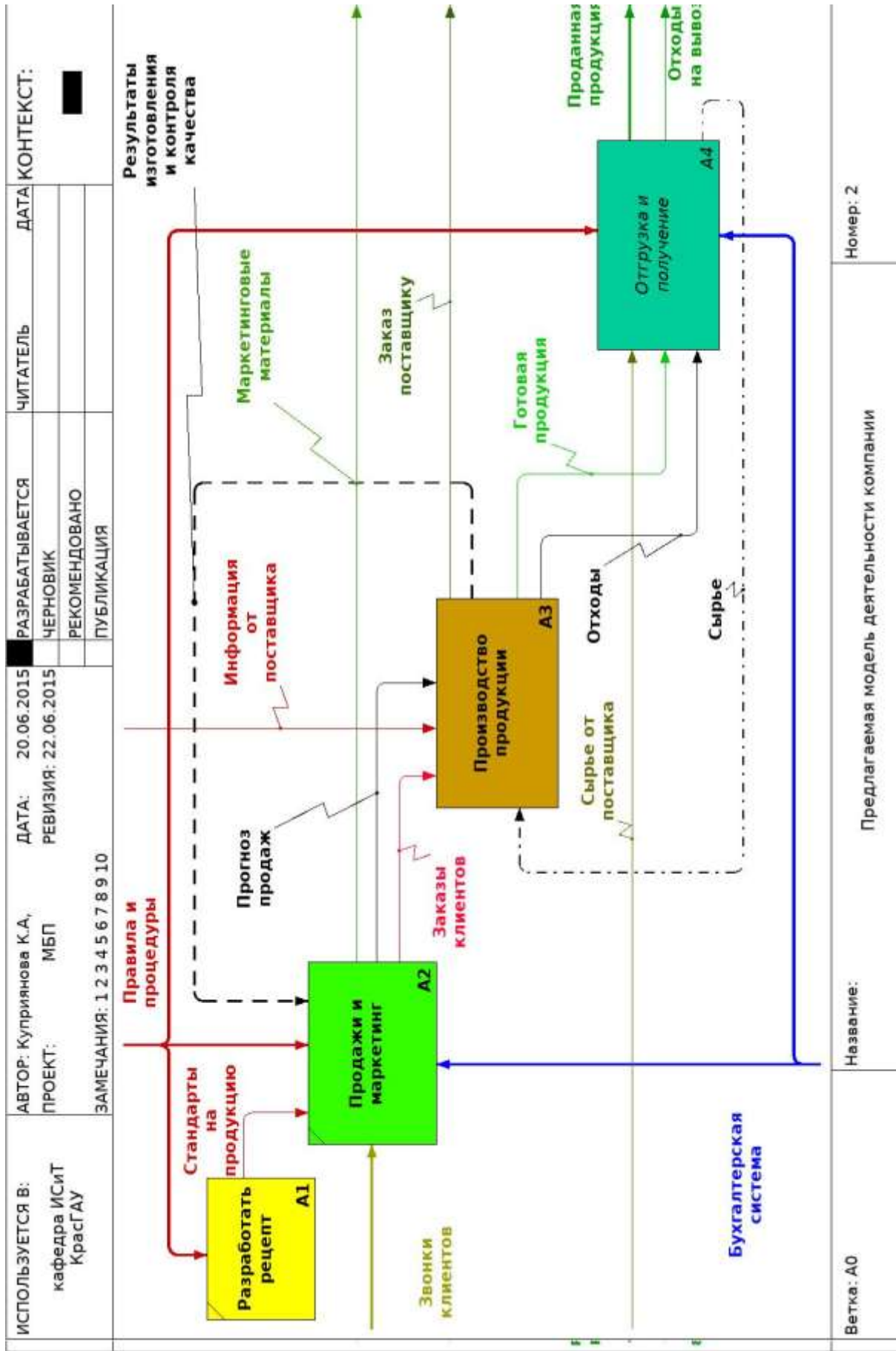


Диаграмма К

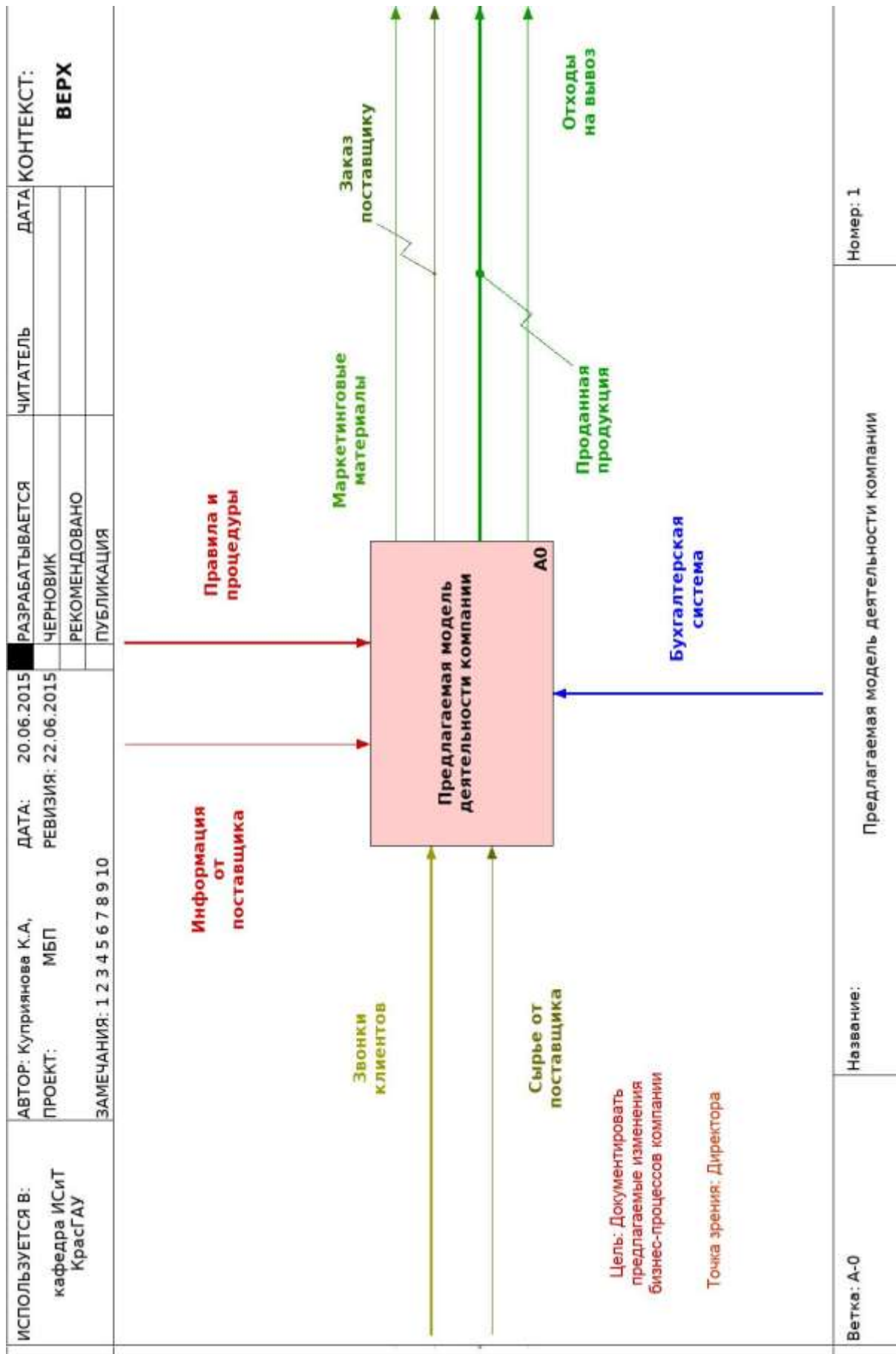
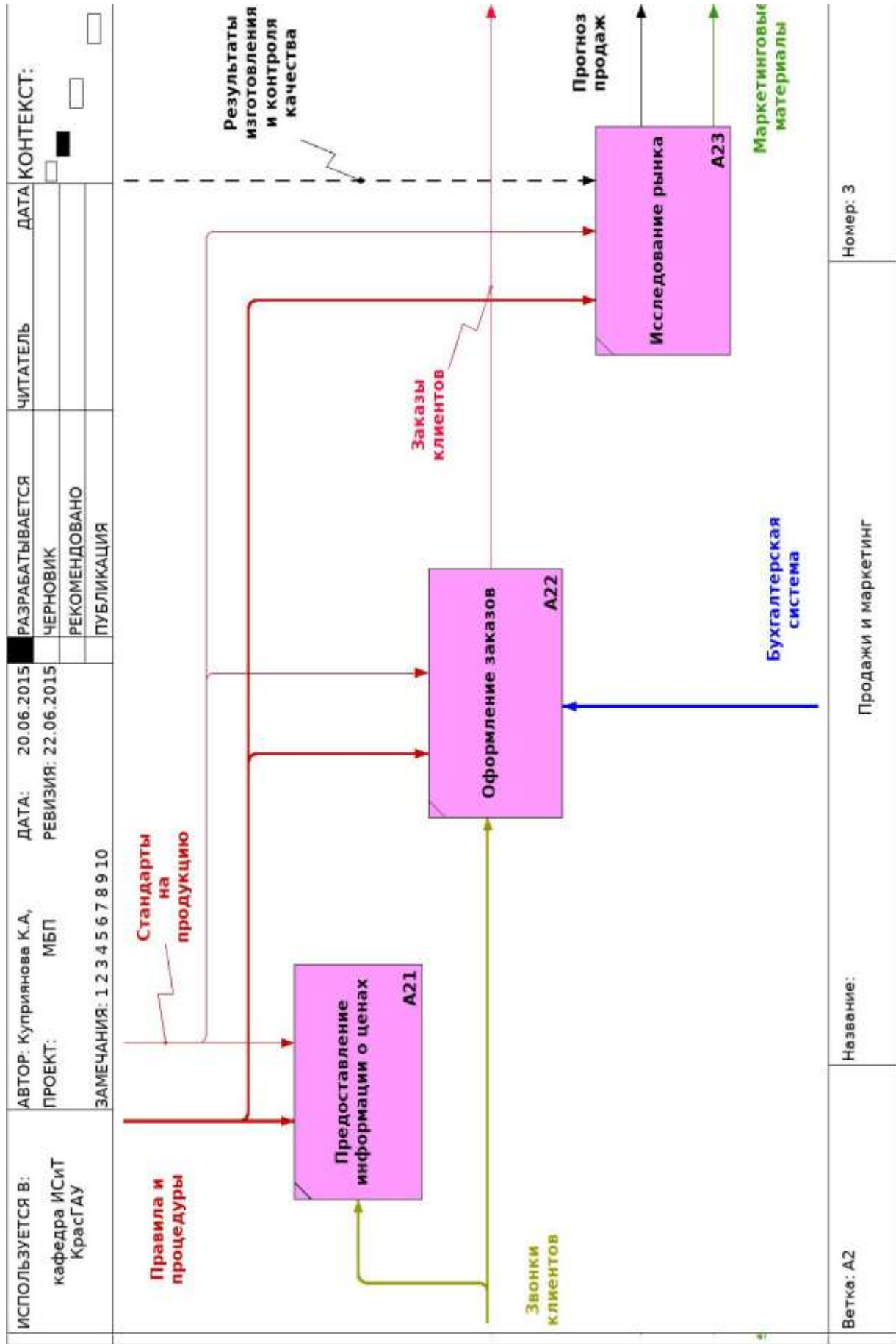


Диаграмма Л



Номер: 3

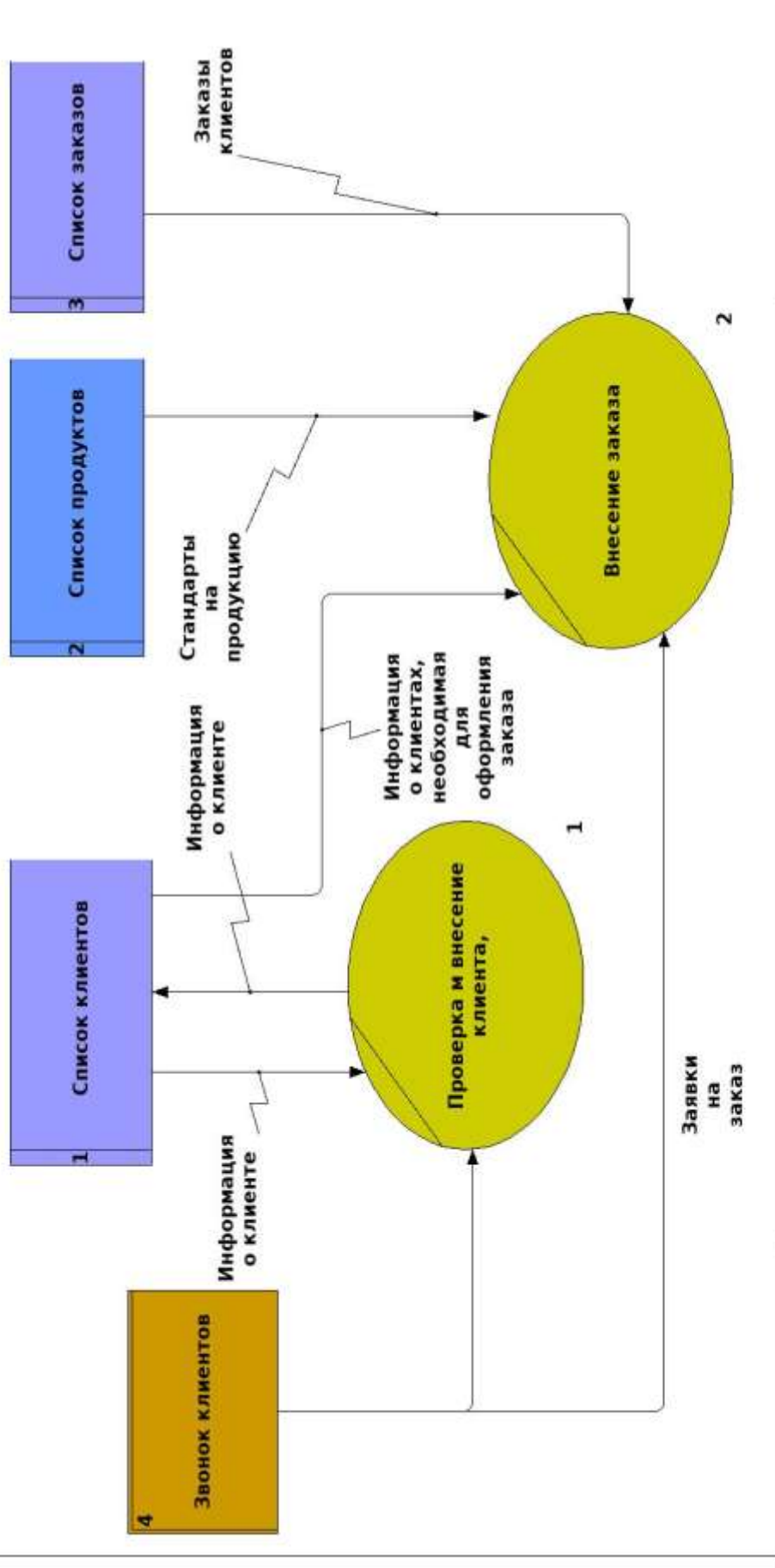
Продажи и маркетинг

Название:

Ветка: A2

Диаграмма М

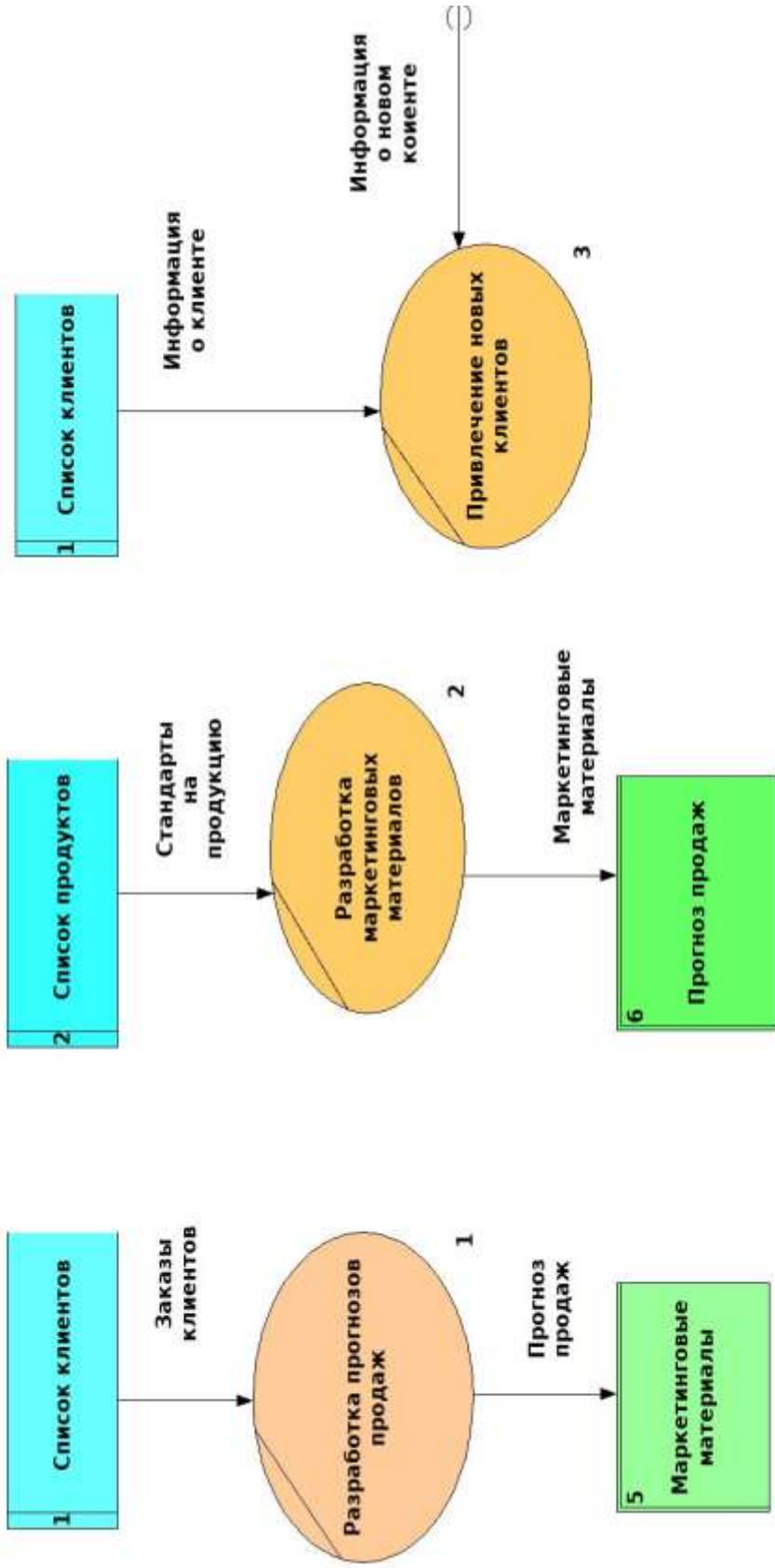
ИСПОЛЬЗУЕТСЯ В: кафедра ИСиТ КрасГАУ	АВТОР: Куприянова К.А, ПРОЕКТ: МБП	ДАТА: 20.06.2015 РЕВИЗИЯ: 22.06.2015	РАЗРАБАТЫВАЕТСЯ ЧЕРНИК РЕКОМЕНДОВАНО ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ:
ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10						



Ветка: A22	Название: Оформление заказов	Номер: 4
------------	------------------------------	----------

Диаграмма Н

ИСПОЛЬЗУЕТСЯ В: кафедра ИСиТ КрасГАУ	АВТОР: Куприянова К.А.	ДАТА: 20.06.2015	РАЗРАБАТЫВАЕТСЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ:	
	ПРОЕКТ: МБП	РЕВИЗИЯ: 22.06.2015	ЧЕРНОВИК				
	ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10		РЕКОМЕНДОВАНО				
			ПУБЛИКАЦИЯ				



Ветка: A23	Название: Исследование рынка	Номер: 5
------------	------------------------------	----------

Диаграмма О

ОГЛАВЛЕНИЕ

Введение	3
1-й день. МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ.....	5
1.1. Задачи менеджера.....	5
1.2. Преобразование предприятия	6
1.3. Способы разработки моделей	8
1.4. Технология Ramus Education.....	8
1.5. Горячие слова.....	10
2-й день. ОСНОВНЫЕ ПОНЯТИЯ IDEF0	13
2.1. Назначение	13
2.2. Концепция	14
2.3. Компоненты языка	15
2.4. Имена и метки.....	16
2.5. Блоки и стрелки	16
2.6. Диаграммы	18
2.6.1. Типы диаграмм	18
2.6.2. Контекстная диаграмма	18
2.6.3. Дочерняя и родительская диаграммы	19
2.6.4. Текст и глоссарий	20
2.6.5. Диаграммы иллюстраций и дерева узлов	21
2.7. Свойства диаграмм.....	21
2.7.1. Стрелки-условия.....	21
2.7.2. Параллельное функционирование.....	21
2.7.3. Связи блоков на диаграмме	22
2.8. Правила построения диаграмм	24
2.9. Классификация функций	26
2.10. Управление	27
2.11. Типизация функциональных моделей.....	28
2.12. Горячие слова.....	29
3-й день. УСТАНОВКА JDK, Ramus Education, Modelio	33
3.1. Установка JDK	33
3.2. Установка JDK (Linux).....	38
3.3. Установка Ramus Education	38
3.4. Установка Ramus Education (Linux).....	42
3.5. Установка JDK 8	42
3.6. Установка переменных среды JAVA_HOME и PATH.....	42
3.7. Hello World!	43

3.8. Установка JDK 8 (Linux).....	45
3.9. Установка Modelio.....	47
4-й день. СОЗДАНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ДЕЯТЕЛЬНОСТИ КОМПАНИИ ПО ПРОИЗВОДСТВУ ПЕЛЬМЕНЕЙ.....	48
4.1. Пользовательский интерфейс	48
4.1.1. Панель инструментов.....	48
4.1.2. Контекстное меню блока	50
4.1.3. Контекстное меню стрелки	51
4.1.4. Меню диаграммы	53
4.2. Построение моделей IDEF0	54
4.3. Модели «как есть» и «как будет».....	56
4.4. Описание деятельности компании	57
4.5. Создание контекстной диаграммы	57
4.5.1. Контекст	58
4.5.2. Блок.....	59
4.5.3. Граничные стрелки.....	59
4.6. Создание диаграммы декомпозиции	60
4.6.1. Декомпозиция	60
4.6.2. Связывание.....	61
4.6.3. Разветвление	61
4.6.4. Внутренние стрелки.....	62
4.6.5. Словарь стрелок	63
4.6.6. Туннель.....	63
4.7. Создание диаграммы декомпозиции A2	64
4.8. Синтаксический контроль	67
5-й день. ДОПОЛНЕНИЕ МОДЕЛИ ДИАГРАММОЙ BPMN.....	68
5.1. Назначение	68
5.2. Основные понятия.....	68
5.3. Основные графические элементы моделирования.....	70
5.4. Действия.....	72
5.5. События.....	74
5.6. Шлюзы.....	76
5.7. Данные.....	77
5.8. Пулы.....	77
5.9. Соединяющие элементы.....	78
5.10. Декомпозиция блока IDEF0 в диаграмму BPMN	79
6-й день. РЕИНЖИНИРИНГ БИЗНЕС-ПРОЦЕССОВ КОМПАНИИ ПО ПРОИЗВОДСТВУ ПЕЛЬМЕНЕЙ	81

6.1. Расщепление модели.....	81
6.2. План реинжиниринга.....	82
6.3. Модификация модели.....	82
6.5. Перенос блока.....	84
6.6. Модификация BPMN.....	84
7-й день. ДОПОЛНЕНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ДИАГРАММОЙ DFD.....	86
7.1. Назначение.....	86
7.2. Основные понятия.....	86
7.2.1. Процесс.....	86
7.2.2. Поток данных.....	87
7.2.3. Хранилище данных.....	87
7.2.4. Внешняя сущность.....	88
7.3. Построение диаграммы DFD.....	88
7.4. Горячие слова.....	89
7.5. Декомпозиция блока.....	89
7.6. Декомпозиция блока IDEF0 в диаграмму DFD.....	90
ЗАКЛЮЧЕНИЕ.....	92
ЛИТЕРАТУРА.....	94
ПРИЛОЖЕНИЯ.....	97

Моделирование бизнес-процессов с помощью IDEF0, DFD, BPMN за 7 дней

Учебное пособие

Миндалёв Игорь Викторович

Редактор М.М. Ионина

Санитарно-эпидемиологическое заключение № 24.49.04.953.П. 000381.09.03 от 25.09.2003 г.

Подписано в печать 26. 05. 2016. Формат 60х90/16. Бумага тип. № 1.

Печать – ризограф. Усл. печ. л. 8,0. Тираж 56 экз. Заказ № 165

Редакционно-издательский центр Красноярского государственного аграрного университета
660017, Красноярск, ул. Ленина, 117