

А. В. Игнатьев

МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Учебное пособие

Волгоград, ВолгАСУ, 2014



УДК 004.4(075.8)
ББК 32.97я73
И266

Р е ц е н з е н т ы:

кандидат технических наук *М. В. Филиппов*, заведующий кафедрой информационных систем и технологий Волгоградского института бизнеса; кандидат технических наук *А. А. Платонов*, доцент кафедры прикладной математики и вычислительной техники Волгоградского государственного архитектурно-строительного университета

*Утверждено редакционно-издательским советом университета
в качестве учебного пособия*

Игнатъев, А. В.
И266 Методы и средства проектирования информационных систем и технологий [Электронный ресурс] : учебное пособие / А. В. Игнатъев ; М-во образования и науки Рос. Федерации, Волгогр. гос. архит.-строит. ун-т. — Электронные текстовые и графические данные (0,9 Мбайт). — Волгоград : ВолГАСУ, 2014. — Учебное электронное издание сетевого распространения. — Систем. требования: PC 486 DX-33; Microsoft Windows XP; Internet Explorer 6.0; Adobe Reader 6.0. — Официальный сайт Волгоградского государственного архитектурно-строительного университета. Режим доступа: <http://www.vgasu.ru/publishing/on-line/> — Загл. с титул. экрана.

ISBN 978-5-98276-682-3

Описаны современные методы проектирования информационных систем и технологий. Представлены различные подходы к проектированию информационных систем и дана их характеристика.

Для студентов-бакалавров всех форм обучения направления «Информационные системы и технологии».

Для удобства работы с изданием рекомендуется пользоваться функцией Bookmarks (Закладки) в боковом меню программы Adobe Reader.

УДК 004.4(075.8)
ББК 32.97я73

ISBN 978-5-98276-682-3



© Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Волгоградский государственный архитектурно-строительный университет», 2014

Оглавление

1. Основные положения проектирования информационных систем и технологий	4
1.1. Что такое проектирование?	4
1.2. Что мы проектируем?	5
Контрольные вопросы	7
2. Модели жизненного цикла разработки программного обеспечения информационной системы	8
2.1. Понятие и модели жизненного цикла	8
2.2. Каскадная (водопадная) модель	8
2.3. Итеративная и инкрементальная модель — эволюционный подход	10
2.4. Спиральная модель	11
Контрольные вопросы	13
3. Целеориентированное проектирование и его интеграция в жизненный цикл разработки продукта	14
3.1. Ошибки проектирования	14
3.2. Целеориентированное проектирование	16
3.2.1. Исследования	17
3.2.2. Моделирование	18
3.2.3. Выработка требований	18
3.2.4. Определение инфраструктуры	19
3.2.5. Детализация	20
3.2.6. Сопровождение разработки	20
Контрольные вопросы	21
4. Методы сбора требований	22
4.1. Значение качественных исследований	22
4.2. Виды качественных исследований	23
4.2.1. Интервьюирование заинтересованных лиц	23
4.2.2. Интервьюирование экспертов в предметной области	24
4.2.3. Интервьюирование покупателей	24
4.2.4. Интервьюирование пользователей	25
4.2.5. Анкетирование	26
4.2.6. Наблюдение за пользователями	26
4.2.7. Самостоятельное описание требований	27
4.2.8. Обзор литературы и технической документации	27
4.2.9. Аудит продукта/прототипа и конкурирующих решений	28
4.2.10. Совместные семинары	28
Контрольные вопросы	28
5. Модели бизнес-процессов	29
5.1. Методики моделирования бизнес-процессов	29
5.1.1. Функциональная методика IDEF0	30
5.1.2. Описание бизнес-процессов в Унифицированном языке моделирования	33
5.1.3. ДРАКОН-технология	34
5.2. Описание бизнес-процесса «Планирование закупок и размещение заказов поставщикам»	38
Контрольные вопросы	50
6. Метод персонажей	51
6.1. Кто такие персонажи?	51
6.2. Разработка персонажей	52
Контрольные вопросы	55
Список использованной литературы	56
Список рекомендуемой литературы	56

1. ОСНОВНЫЕ ПОЛОЖЕНИЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

1.1. Что такое проектирование?

Обратившись к Новой философской энциклопедии, мы найдем там следующее определение проектирования:

Проектирование — один из основных (наряду с инженерной деятельностью) способов создания техники и других изделий и сооружений.

Исторически проектирование возникает внутри сферы изготовления (домостроения, кораблестроения, изготовления машин, градостроения и т. д.). Это часть работы, связанная с расчетами и изображением на чертежах внешнего вида, строения и функционирования будущего изделия (дома, корабля, машины). Проектирование становится самостоятельной сферой деятельности, когда происходит разделение труда между архитектором (конструктором, расчетчиком, чертежником) и собственно изготовителем (строителем, машиностроителем).

Исходя из предпосылки, что проектирование является самостоятельной сферой деятельности, уточним определение проектирования.

Проектирование — деятельность человека или организации по созданию проекта, то есть прототипа, прообраза предполагаемого или возможного объекта, состояния; комплекта документации, предназначенной для создания определенного объекта, его эксплуатации, ремонта и ликвидации, а также для проверки или воспроизведения промежуточных и конечных решений, на основе которых был разработан данный объект.

Проектирование программного обеспечения (ПО) является частным случаем проектирования продуктов и процессов.

Проектирование программного обеспечения — процесс создания проекта программного обеспечения, а также дисциплина, изучающая методы проектирования.

Целью проектирования является определение внутренних свойств системы и детализации ее внешних (видимых) свойств на основе выданных заказчиком требований к ПО (исходные условия задачи).

1.2. Что мы проектируем?

Вопрос, вынесенный в заглавие раздела, определяет дальнейшее содержание изучаемого курса. Для ответа на него дадим основные определения и рассмотрим различие между информационной системой предприятия и системой программного обеспечения, разрабатываемого для этой системы.

Под *системой* понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов. Системы значительно отличаются между собой как по составу, так и по главным целям. Добавление к понятию «система» слова «информационная» отражает цель ее создания и функционирования.

Приведем одно из определений информационной системы, получившее наиболее широкое распространение и приведенное, например, в учебнике [1].

Информационная система — взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленных целей.

Исторически первыми видами информационных систем являются архивы и библиотеки. Им присущи все атрибуты информационной системы. Они обеспечивают в какой-либо предметной области сбор данных, их представление и хранение в определенной форме (книго-, архивохранилища, каталоги и т. д.), в них определяется порядок использования информационных фондов (т. е. определены абоненты, режимы и способы выдачи информации — абоненты, читальные залы и т. п.).

Современное понимание информационной системы предполагает использование средств вычислительной техники в качестве основных технических средства переработки информации. Информационная система обеспечивает формирование и управление информацией в интересах людей. Часть этой информации генерируется автоматически компьютерными системами. Другая часть вводится людьми вручную, поэтому техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

Практика создания и использования автоматизированных информационных систем в различных сферах деятельности позволяет дать более широкое и универсальное определение, которое полнее отражает все аспекты их сущности.

Под *информационной системой* в дальнейшем понимается организованная совокупность программно-технических и других вспомогательных средств, технологических процессов и функционально-определенных групп работников, обеспечивающих сбор, представление и накопление информационных ресурсов в определенной предметной области, поиск и выдачу сведений, необходимых для удовлетворения информационных потребностей установленного контингента пользователей — абонентов системы.

Другими словами, информационные системы — социальные системы, которые включают и используют ПО и другие компоненты в интересах предприятия.

Можно выделить следующие компоненты информационной системы:

- люди;
- данные/информация;
- процедуры;
- ПО;
- аппаратное обеспечение;
- линии связи.

Системы ПО, разработке и проектированию которых посвящена основная часть изучаемого курса, являются частью (хотя и фундаментальной) намного большей информационной системы предприятия. Л. Мацяшек [2] приводит диаграмму Венна, демонстрирующую включение системы ПО в информационную систему предприятия, которая, в свою очередь, является компонентом предприятия как целого, а само предприятие является частью бизнес-среды (рис. 1).



Рис. 1. Диаграмма Венна, демонстрирующая включение системы ПО в информационную систему предприятия

Различие между процессом создания и эксплуатации ПО, с одной стороны, и бизнес-процессом, с другой, определяется и связано с продуктом или сервисом, которые ожидается получить от этих процессов. Результат процесса создания и эксплуатации ПО — ПО. Результат бизнес-процесса — бизнес.

Имеются четкие отношения между ПО и бизнесом. ПО — потенциально главный вкладчик в бизнес-успех. ПО — часть бизнеса, но не наоборот. Фактически это отношение старшинства было отображено на рис. 1. Предприятие на рис. 1 — это другой термин для обозначения бизнеса. Цель функционирования предприятия состоит в том, чтобы сформировать цепочку создания ценностей, которая обеспечивает реализацию бизнес-назначения, задач и целей.

Различие между процессом создания, эксплуатации ПО и бизнес-процессом сродни различию между эффективностью процесса и результативностью.

Эффективность (efficiency) означает «делать что-то правильно». *Результативность* (effectiveness) означает «делать правильную вещь». В организационных терминах результативность подразумевает достижение бизнес-назначения, задач и целей. Все они — то, что нужно получить как результат процесса стратегического планирования, проводимого на предприятии. Частью стратегического планирования является бизнес-моделирование. Следовательно, целью бизнес-процесса является обеспечение результативности.

В противоположность этому процесс создания и эксплуатации ПО должен обеспечить эффективность. Следовательно, возможна ситуация, когда процесс создания и эксплуатации ПО даст очень эффективный программный продукт или сервис, который будет нерезультативен для бизнеса. В лучшем случае нерезультативность может означать нейтральный результат с точки зрения бизнеса. В худшем случае это может сделать бизнес уязвимым для конкурентов и даже привести к банкротству.

Поэтому ясно, что процесс создания и эксплуатации ПО является характерной частью бизнес-процесса, жизненно важной для успеха предприятия.

Чтобы обеспечить результативность наряду с эффективностью, процесс создания и эксплуатации ПО должен быть составной частью бизнес-процесса.

В конце концов, решение разрабатывать ли программный продукт или сервис в первую очередь будет результатом стратегического планирования и бизнес-моделирования.

Процесс программной инженерии устанавливает соответствие ПО и бизнес-процессов. С одной стороны, разработка ПО все более и более внедряется в среду бизнес-моделирования.

С другой стороны, разработка ПО предназначена для поставки программных продуктов и сервисов, увеличивая для предприятия стоимость бизнеса. Это имеет отношение к трем уровням управления, которые бизнес-процессы обслуживают: оперативный, тактический и стратегический.

Помещение разработки ПО в среду бизнес-моделирования означает, что процесс создания и эксплуатации ПО получается из более широкой бизнес-модели и старается поддерживать и реализовывать конкретный бизнес-процесс в этой модели. Отсюда следует, что программный продукт/сервис не может быть только информационным сервисом. Он также должен реализовывать бизнес-операции или содействовать им. Проект информационной системы должен или явно определить бизнес-процесс, который он обслуживает, или, что лучше, быть частью системы управления знаниями.

Контрольные вопросы

1. Дайте определение проектирования.
2. Дайте определение проектирования программного обеспечения.
3. Дайте определение системы.
4. Дайте определение информационной системы.
5. Какие компоненты информационной системы можно выделить?
6. В чем различие между процессом создания и эксплуатации ПО и бизнес-процессом?

2. МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Понятие и модели жизненного цикла

В программной инженерии термин «*жизненный цикл*» (на английском языке *lifecycle*, обычно пишется единым словом) применяется к искусственным системам ПО и означает изменения, которые происходят в «жизни» программного продукта.

Различные стадии между «рождением» изделия и его возможной «смертью» известны как стадии жизненного цикла.

Наиболее часто говорят о следующих моделях жизненного цикла:
каскадная (водопадная) или последовательная;
итеративная и инкрементальная-эволюционная (гибридная, смешанная);
спиральная (модель Бозма).

Легко обнаружить, что в разное время и в разных источниках приводится разный список моделей и их интерпретация. Например, ранее, инкрементальная модель понималась как построение системы в виде последовательности сборок (релизов), определенной в соответствии с заранее подготовленным планом и заданными (уже сформулированными) и неизменными требованиями. Сегодня об инкрементальном подходе чаще всего говорят в контексте постепенного наращивания функциональности создаваемого продукта.

Может показаться, что индустрия пришла, наконец, к общей «правильной» модели. Однако каскадная модель, многократно «убитая» и теорией, и практикой, продолжает встречаться в реальной жизни. Спиральная модель является ярким представителем эволюционного взгляда, одновременно представляя собой единственную модель, уделяющую явное внимание анализу и предупреждению рисков.

Рассмотрим и охарактеризуем три модели — каскадную, эволюционную и спиральную.

2.2. Каскадная (водопадная) модель

Данная модель предполагает строго последовательное (во времени) и однократное выполнение всех фаз проекта с жестким (детальным) предварительным планированием в контексте предопределенных или однажды и целиком определенных требований к программной системе.

На рис. 2 изображены типичные фазы каскадной модели жизненного цикла и соответствующие активы проекта, являющиеся для одних фаз выходами, а для других — входами.



Рис. 2. Каскадная модель жизненного цикла

При активном использовании эта модель продемонстрировала свою проблемность в подавляющем большинстве ИТ-проектов, за исключением, может быть, отдельных проектов обновления программных систем для критически-важных программно-аппаратных комплексов (например, авионики или медицинского оборудования). Практика показывает, что в реальном мире, особенно в мире бизнес-систем, каскадная модель не должна применяться. Специфика таких систем (если можно говорить о «специфике» для подавляющего большинства создаваемых систем) в том, что требования характеризуются высокой динамикой корректировки и уточнения, невозможно четко и однозначно определить требования до начала работ по реализации (особенно, для новых систем) и быстрой изменчивостью требований в процессе эксплуатации системы.

Фредерик Брукс во втором издании своего классического труда «Мифический человеко-месяц» так описывает главную беду каскадной модели: «Основное заблуждение каскадной модели состоит в предположениях, что проект проходит через весь процесс один раз, архитектура хороша и проста в использовании, проект осуществления разумен, а ошибки в реализации устраняются по мере тестирования. Иными словами, каскадная модель исходит из того, что все ошибки будут сосредоточены в реализации, а потому их устранение происходит равномерно во время тестирования компонентов и системы» [3].

В каскадной модели переход от одной фазы проекта к другой предполагает полную корректность результата (выхода) предыдущей фазы. Однако, например, неточность какого-либо требования или некорректная его интерпретация приводит к тому, что приходится «откатываться» к ранней фазе проекта, а требуемая переработка не просто выбивает проектную команду из графика, но приводит часто к качественному росту затрат и, не исключено, к прекращению проекта в той форме, в которой он изначально задумывался. Кроме того, эта модель не способна гарантировать необходимую скорость

отклика и внесение соответствующих изменений в ответ на быстро меняющиеся потребности пользователей, для которых программная система является одним из инструментов исполнения бизнес-функций. И таких примеров проблем, порожаемых самой природой модели, можно привести достаточно много для отказа от каскадной модели жизненного цикла.

2.3. Итеративная и инкрементальная модель — эволюционный подход

Итеративная модель предполагает *разбиение жизненного цикла проекта на последовательность итераций*, каждая из которых напоминает «мини-проект», включая все фазы жизненного цикла в применении к созданию меньших фрагментов функциональности (по сравнению с проектом в целом).

Цель каждой итерации — получение работающей версии программной системы, включающей функциональность, определенную интегрированным содержанием всех предыдущих и текущей итерации. Результата финальной итерации содержит всю требуемую функциональность продукта. Таким образом, с завершением каждой итерации, *продукт развивается инкрементально*.

С точки зрения структуры жизненного цикла такую модель называют *итеративной (iterative)*. С точки зрения развития продукта — *инкрементальной (incremental)*. Опыт индустрии показывает, что невозможно рассматривать каждый из этих взглядов изолировано. Чаще всего такую *смешанную эволюционную модель* называют просто итеративной (говоря о процессе) и/или инкрементальной (говоря о наращивании функциональности продукта).

Эволюционная модель подразумевает не только сборку работающей (с точки зрения результатов тестирования) версии системы, но и ее развертывание в реальных операционных условиях с анализом откликов пользователей для определения содержания и планирования следующей итерации. «Чистая» инкрементальная модель не предполагает развертывания промежуточных сборок (релизов) системы и все итерации проводятся по заранее определенному плану наращивания функциональности, а пользователи (заказчик) получают только результат финальной итерации как полную версию системы.

Таким образом, значимость эволюционного подхода на основе организации итераций особо проявляется в снижении неопределенности с завершением каждой итерации. В свою очередь, снижение неопределенности позволяет уменьшить риски. Рис. 3 иллюстрирует некоторые идеи эволюционного подхода, предполагая, что итеративному разбиению может быть подвержен не только жизненный цикл в целом, включающий перекрывающиеся фазы — формирование требований, проектирование, конструирование и т. п., но и каждая фаза может, в свою очередь, разбиваться на уточняющие итерации, связанные, например, с детализацией структуры декомпозиции проекта — например, архитектуры модулей системы.

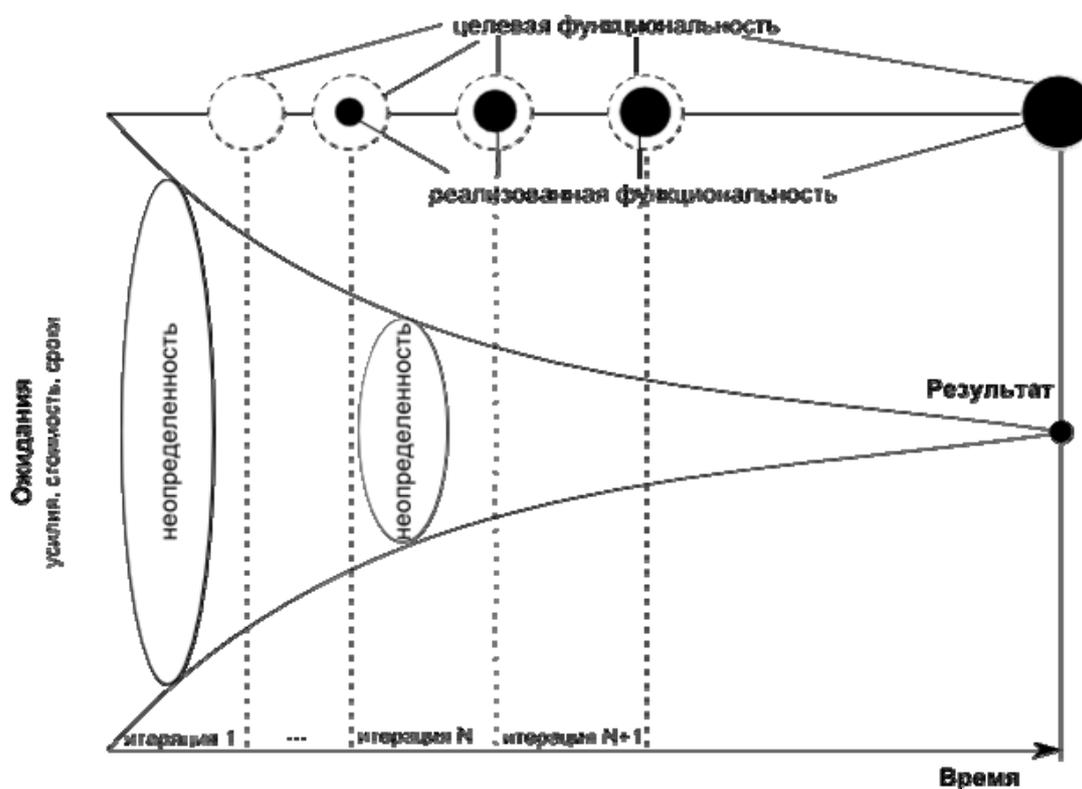


Рис. 3. Снижение неопределенности и инкрементальное расширение функциональности при итеративной организации жизненного цикла

2.4. Спиральная модель

Наиболее известным и распространенным вариантом эволюционной модели является *спиральная модель*, ставшая уже фактически самостоятельной моделью, имеющей различные сценарии развития и детализации.

Спиральная модель (рис. 4) была впервые сформулирована Барри Боэмом в 1988 г. Отличительной особенностью этой модели является специальное внимание *рискам*, влияющим на организацию жизненного цикла.

Главное достижение спиральной модели состоит в том, что она предлагает спектр возможностей адаптации удачных аспектов существующих моделей процессов жизненного цикла.

Спиральная модель обладает рядом преимуществ:

1. Уделяет специальное внимание раннему анализу возможностей повторного использования. Это обеспечивается, в первую очередь, в процессе идентификации и оценки альтернатив.

2. Предполагает возможность эволюции жизненного цикла, развитие и изменение программного продукта. Главные источники изменений заключены в целях, для достижения которых создается продукт. Подход, предусматривающий скрытие информации о деталях на определенном уровне дизайна, позволяет рассматривать различные архитектурные альтернативы так, как если бы мы говорили о единственном проектном решении, что уменьшает риск невозможности согласования функционала продукта и изменяющихся целей (требований).

3. *Предоставляет механизмы достижения необходимых параметров качества как составную часть процесса разработки программного продукта.* Эти механизмы строятся на основе идентификации всех типов целей (требований) и ограничений на всех циклах спирали разработки. Например, ограничения по безопасности могут рассматриваться как риски на этапе специфицирования требований.

4. *Уделяет специальное внимание предотвращению ошибок и отбрасыванию ненужных, необоснованных или неудовлетворительных альтернатив на ранних этапах проекта.* Это достигается явно определенными работами по анализу рисков, проверкой различных характеристик создаваемого продукта (включая архитектуру, соответствие требованиям и т. п.) и подтверждением возможности двигаться дальше на каждом цикле процесса разработки.

5. *Позволяет контролировать источники проектных работ и соответствующих затрат.* По сути, речь идет об ответе на вопрос — как много усилий необходимо затратить на анализ требований, планирование, конфигурационное управление, обеспечение качества, тестирование, формальную верификацию и т. д.? Модель, ориентированная на риски, позволяет в контексте конкретного проекта решить задачу приложения адекватного уровня усилий, определяемого уровнем рисков, связанных с недостаточным выполнением тех или иных работ.

6. *Не проводит различий между разработкой нового продукта и расширением (или сопровождением) существующего.* Этот аспект позволяет избежать часто встречающегося отношения к поддержке и сопровождению как ко второсортной деятельности. Такой подход предупреждает большое количество проблем, возникающих в результате одинакового уделения внимания как обычному сопровождению, так и принципиально важным вопросам расширения функциональности продукта, всегда связанным с повышенными рисками.

7. *Позволяет решать интегрированные задачи системной разработки, охватывающие и программную, и аппаратную составляющие создаваемого продукта.* Подход, основанный на управлении рисками и возможности своевременного отбрасывания непривлекательных альтернатив (на ранних стадиях проекта), сокращает расходы, и одинаково применим и к аппаратной части, и к программному обеспечению.

Так как взглядов на детализацию описания жизненного цикла может быть много, то существуют различные методики, среди которых наибольшее распространение получили:

Rational Unified Process (RUP);

Enterprise Unified Process (EUP);

Microsoft Solutions Framework (MSF) в обоих представлениях: MSF for Agile и MSF for CMMI (анонсированная изначально как MSF Formal);

Agile-практики (eXtreme Programming (XP), Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), SCRUM и др.).

3. ЦЕЛЕОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ И ЕГО ИНТЕГРАЦИЯ В ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОДУКТА

3.1. Ошибки проектирования

Во втором издании своей книги «Алан Купер об интерфейсе. Основы проектирования взаимодействия» Алан Купер со товарищи так описывают главную проблему, возникающую при создании систем программного обеспечения: «В большинстве своем цифровые продукты возникают в процессе разработки — как монстр из булькающей протоплазмы в резервуаре. Разработчики, вместо того чтобы планировать и действовать исходя из нужд тех пользователей, которые приобретут и будут использовать созданные продукты, сплошь и рядом сосредотачиваются на технологии и в результате порождают решения, слабоуправляемые и неудобные в применении. Словно безумные ученые они терпят поражение, потому что не наделяют свои создания человечностью...

Так в чем же реальная проблема? Почему индустрия технологий оказывается в целом недееспособной, когда требуется продумать интерактивную составляющую цифровых продуктов? Тому есть три основных причины: отсутствие представления о пользователях, конфликт между потребностями людей и приоритетами разработки, а также отсутствие процесса, позволяющего понимать потребности человека и помогающего в разработке удобной формы и качественного поведения продукта» [4].

Сложившаяся практика проектирования такова, что мы получаем четкое представление о бизнес-процессах предприятия, для которого создается программное обеспечение. Соответственно, мы имеем смутное представление о работе, которую делают будущие пользователи системы, и некоторых основных часто выполняемых задачах, но практически не получаем сведений о том, как они в действительности будут применять продукт, который мы создаем, почему они занимаются той деятельностью, в которой им может помочь наш продукт, почему они могут отдать предпочтение именно нашему продукту, а не продукту конкурента, и как нам добиться этого.

Продукты, при проектировании и создании которых преследовались только цели бизнеса, ожидает провал, так как личные цели пользователей требуют своей доли внимания. Учет личных целей пользователей при проектировании продукта позволяет гораздо более эффективно достигать целей бизнеса. Цели — не то же самое, что задачи или деятельность. Цель — это предвосхищение конечного состояния, тогда как задачи и деятельность являются лишь промежуточными этапами (на различных уровнях организации), необходимыми для достижения целей.

Программы, дающие пользователям возможность решать стоящие перед ними задачи, но не обращающие внимания на цели, редко позволяют выполнять работу действительно эффективно. Если перед пользователем поставлена задача ввести в базу данных 5000 имен и адресов, самое идеальное приложение для ввода данных не даст ему ничего близко похожего на то удовлетворение, которое принесет инструмент, автоматически извлекающий имена из системы хранения счетов.

Строя проектирование исключительно на основе анализа деятельности и задач, мы рискуем попасть в ловушку устаревших технологий или применить модель, соответствующую целям корпорации, но не отвечающую целям пользователей. Взгляд через призму целей позволяет пользоваться преимуществами современной технологии для исключения лишних задач и радикального упорядочения структуры деятельности. Понимание целей пользователя помогает проектировщикам избавляться от деятельности и задач, которые технология способна выполнять за человека.

Необходимо использовать такие методы проектирования, которые сочетают:

- понимание желаний, потребностей, мотивации пользователей и контекста, в котором эти пользователи находятся;

- понимание возможностей, требований и ограничений бизнеса, технологии и предметной области;

- использование этих знаний в качестве основы всех планов по созданию продуктов, форма, содержание и поведение которых делают их полезными, удобными и желанными, а также экономически жизнеспособными и технически осуществимыми.

Если проектирование осуществляется подходящими методами, оно способно восстановить отсутствующую связь человека с технологическими продуктами. Очевидно, что большинство существующих подходов к проектированию цифровых продуктов не дают обещанного эффекта.

Создание качественных продуктов требует исследований пользовательской аудитории, в то время как суть этих исследований для многих организаций по-прежнему остается загадкой. Проблема возникает непосредственно после анализа результатов: большинство традиционных подходов ничего не говорят о том, как преобразовать результаты исследований в проектные решения. Сотню страниц с результатами анкетирования пользователей не так-то легко превратить в набор требований к продукту, а извлечь из них пони-

мание того, как эти требования должны быть выражены в терминах ясной и осмысленной инфраструктуры интерфейса, еще сложнее. Проектирование остается «черным ящиком»: «А вот здесь происходит чудо». Эта пропасть между результатами исследований и конечными проектными решениями есть порождение процесса, неспособного протянуть связи от запросов пользователя к тому, чем продукт становится в финале. Эта проблема преодолевается при помощи целеориентированного подхода.

3.2. Целеориентированное проектирование

Одна из проблем существующего процесса разработки состоит в том, что роли участников чрезмерно узки: аналитики исследуют, а проектировщики проектируют. В этой модели не хватает системных средств для перевода и синтеза результатов исследований в интерфейсные решения. Один из способов решения проблемы для проектировщиков — научиться быть исследователями.

Прямые и обширные контакты с пользователями, без которых не обходится ни одно серьезное исследование, погружают проектировщиков в мир пользователей и заставляют думать о них задолго до того, как речь пойдет о выработке решений. Одна из наиболее опасных практик при создании продукта — изоляция проектировщиков от пользователей. Кроме того, обычному исследователю часто сложно понять, какая информация о пользователях существенна с точки зрения проектирования. Вовлечение проектировщиков в исследование снимает оба этих вопроса.

Весьма немногие из распространенных методов проектирования включают в себя средства эффективного и систематического преобразования знаний, собранных в ходе исследований, в детальную спецификацию интерфейса.

Другая же причина состоит в том, что очень немногие подходы фиксируют поведение пользователей в форме, пригодной для формирования определения продукта. Вместо информации о целях пользователей, большинство методов описывают задачи. Информация такого типа хорошо подходит для создания компоновочных схем, моделирования рабочего процесса и преобразования функций в элементы пользовательского интерфейса, но далеко не столь полезна для определения общей инфраструктуры, отражающей то, чем продукт является, что он делает и как это соответствует различным потребностям пользователей.

Для преодоления разрыва нам требуется строгий систематический процесс создания моделей пользователей, определения требований к пользовательскому интерфейсу и преобразования их в общую концепцию взаимодействия. Задача целеориентированного проектирования — устранить существующий в процессе разработки цифровых продуктов разрыв между исследованиями пользовательской аудитории и проектированием, эффективно сочетая новые и уже известные подходы.

Целеориентированное проектирование позволяет создавать решения, соответствующие потребностям и целям пользователей с одной стороны, а также бизнес-требованиям и технологическим ограничениям — с другой. Процесс можно грубо разделить на шесть стадий: исследования, моделирование, выработка требований, определение общей инфраструктуры, детализация и сопровождение (рис. 5).

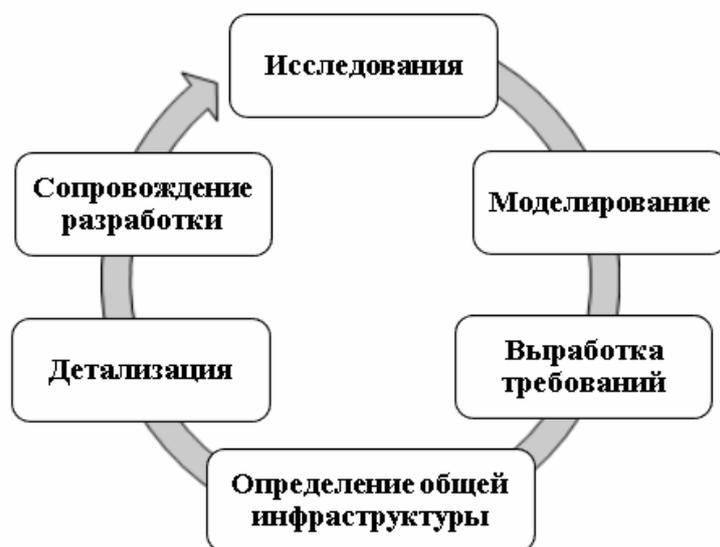


Рис. 5. Процесс целеориентированного проектирования

Рассмотрим содержание каждого из этапов.

3.2.1. Исследования

На стадии исследований для сбора качественных данных о существующих и/или потенциальных пользователях продукта применяются такие методы, как наблюдения за потенциальными пользователями системы и их интервьюирование. Помимо этого, если того требует предметная область, может проводиться конкурентный анализ, обзор маркетинговых исследований, обзор статей о технологиях, а также индивидуальное интервьюирование лиц, принимающих решения, разработчиков, специалистов в предметной области и экспертов по технологии.

Одним из основных результатов исследований и интервью с пользователями является набор поведенческих шаблонов — характерных поведенческих моделей, помогающих классифицировать варианты использования будущего или существующего продукта. Анализ этих поведенческих шаблонов позволяет определять цели и мотивы (частные и общие желаемые результаты применения продукта).

В деловой и технической областях такие поведенческие шаблоны часто совпадают с бизнес-ролями пользователей; в случае потребительской продукции — соответствуют стилю жизни пользователей. Поведенческие модели и связанные с ними цели пользователей являются основой персонажей, которые создаются на стадии моделирования.

Маркетинговые исследования помогают находить и проводить отбор персонажей, укладывающихся в бизнес-модели. Интервью с лицами, принимающими решения, обзоры литературы и аудит пользовательского интерфейса продуктов дают проектировщикам возможность глубже вникнуть в предметную область и проливают свет на цели бизнеса, атрибуты бренда и технические ограничения, которые следует учесть при проектировании. В главе четвертой приводится более подробная информация о методах сбора требований.

3.2.2. Моделирование

На стадии моделирования поведенческие шаблоны и шаблоны рабочих процессов, выявленные путем анализа результатов полевых исследований и интервью, собираются вместе в виде моделей предметной области и моделей пользователей. Модели предметной области могут включать информационные потоки и диаграммы рабочих процессов. Пользовательские модели, или персонажи, — это подробные и структурированные архетипы пользователей, которые представляют собой различные устойчивые комбинации поведенческих моделей, склонностей, взглядов, целей, мотивов, обнаруженных на стадии исследований.

Персонажи становятся главными действующими лицами описательной методики проектирования, основанной на сценариях. На стадии определения инфраструктуры персонажи способствуют генерации концепций взаимодействия, на стадии детализации — обеспечивают обратную связь, позволяющую улучшить внутреннюю согласованность дизайна и его соответствие целям, а также служат мощным инструментом коммуникации, помогающим разработчикам и руководителям видеть, на чем основывается дизайн, и ранжировать функции продукта, исходя из потребностей пользователей. На стадии моделирования проектировщики применяют разнообразные методологические инструменты для синтеза, дифференциации и ранжирования персонажей.

Проектировщики выявляют различные типы целей и связывают типы возможных моделей поведения с персонажами таким образом, чтобы не оставалось белых пятен и не возникало повторений.

Конкретное направление проектирования выбирается путем сопоставления целей персонажей и создания иерархии приоритетов, основанной на том, насколько широко цели того или иного персонажа покрывают цели других персонажей. Процесс присвоения персонажам типов определяет, насколько серьезное влияние каждый персонаж окажет на окончательную форму и поведение продукта.

Подробнее моделирование персонажей описано в шестой главе.

3.2.3. Выработка требований

Методы проектирования, применяемые проектными командами на стадии выработки требований, обеспечивают столь нужную связь между пользовательскими и всеми прочими моделями и инфраструктурой проекта. Здесь используются сценарные методы проектирования с одним важным но-

вовведением: сценарии концентрируются не на абстрактных задачах пользователей, но прежде всего на достижении целей и удовлетворении потребностей конкретных персонажей. Персонажи дают понимание того, какие задачи действительно важны и почему, что приводит к созданию интерфейса, минимизирующего число задач (усилий), но при этом увеличивающего отдачу от них. Персонажи становятся главными участниками этих сценариев, и проектировщики изучают пространство возможных решений посредством своего рода ролевой игры.

Для каждого интерфейса/ключевого персонажа процесс проектирования на данном этапе включает в себя анализ данных, связанных с персонажем, и анализ функциональных потребностей (выраженный в терминах объектов, действий и контекстов), сформированных и ранжированных с помощью целей персонажей, их моделей поведения, а также особенностей взаимодействия с другими персонажами в различных контекстах.

Такой анализ выполняется посредством последовательного уточнения контекстного сценария. Отправной точкой служит описание «одного дня жизни» персонажа, применяющего продукт, которое намечает высокоуровневые точки соприкосновения с продуктом, после чего происходит пошаговая детализация. Помимо требований, определяемых сценарием, проектировщики рассматривают навыки персонажа и его физические возможности, равно как и вопросы, связанные со средой применения продукта. Происходит также учет и балансирование целей бизнеса, желаемых атрибутов бренда и технических ограничений с целями и потребностями персонажа. На выходе этого процесса возникает сбалансированный перечень требований, включающий в себя пользовательские требования, требования бизнеса и технические ограничения, которым продукт должен удовлетворять.

3.2.4. Определение инфраструктуры

На стадии определения инфраструктуры команда проектировщиков создает общую концепцию продукта, определяя концепцию поведения, графического оформления и, если требуется, физической формы. Проектировщики взаимодействия синтезируют инфраструктуру взаимодействия при помощи контекстных сценариев в сочетании с еще двумя важнейшими методологическими инструментами.

Первый инструмент — набор общих принципов проектирования взаимодействия, которые способствуют определению уместного в контексте различных ситуаций поведения системы.

Второй важнейший методологический инструмент — это набор шаблонов проектирования взаимодействия, являющихся решением (вариации здесь зависят от контекста) для соответствующих типов когда-то проанализированных проблем. Шаблоны проектирования взаимодействия выстроены в иерархию и эволюционируют с появлением новых контекстов. Их функция — не загнать творчество проектировщика в рамки, а дать ему точку опоры для решения сложных задач, снабдив проверенными знаниями о проектировании.

Когда информационные и функциональные потребности описаны на таком высоком уровне, они преобразуются в элементы дизайна в соответствии с принципами взаимодействия, а затем структурируются при помощи шаблонов и принципов в эскизы дизайна и описание поведения. Результатом этого процесса является определение инфраструктуры взаимодействия — устоявшейся концепции проекта, задающей логическую и примерную формальную структуру для последующей детализации. Эта детализация выполняется на следующей стадии при помощи последовательных итераций более узко сфокусированных сценариев. Такой подход часто представляет собой баланс проектирования «сверху вниз» (опирающегося на шаблоны) и проектирования «снизу вверх» (опирающегося на принципы).

3.2.5. Детализация

Стадия детализации схожа со стадией определения инфраструктуры, но в большей степени сосредоточена на подробностях реализации.

Проектировщики взаимодействия фокусируются на согласованности задач, используя ключевые (пошаговые) маршруты, а также проверочные сценарии, дающие максимально подробные пути прохождения по пользовательскому интерфейсу.

3.2.6. Сопровождение разработки

Даже очень продуманное и проверенное проектное решение не позволяет предусмотреть все препятствия и технические сложности на пути разработчиков. Мы на своем опыте знаем, насколько важно оставаться в контакте с разработчиками, чтобы отвечать на их вопросы, возникающие в процессе создания продукта. Часто требуется корректировать проектные решения, упрощать их по мере того, как команда разработчиков назначает приоритетными отдельные фрагменты своей работы и урезает проект, чтобы уложиться в сроки. Если в тот момент, когда возникла нужда в таких решениях, команда проектировщиков недоступна, разработчикам приходится самостоятельно искать выход в условиях дефицита времени, что в конечном итоге может не лучшим образом сказаться на целостности продукта.

Целеориентированное проектирование — мощный инструмент, отвечающий на самые важные вопросы, которые возникают при описании и проектировании цифрового продукта:

- Кем являются пользователи проектируемой системы ПО?
- Чего пытаются достичь пользователи проектируемой системы ПО?
- Что пользователи думают о своих целях сами?
- Какого рода опыт будет для пользователей привлекательным и полезным?
- Как должен себя вести разрабатываемый продукт?
- Как должен выглядеть разрабатываемый продукт?
- Как пользователи будут взаимодействовать с разрабатываемым продуктом?
- Как наиболее эффективно реализовать функции разрабатываемого продукта?
- Как начинающие пользователи будут знакомиться с разрабатываемым продуктом?

Каким образом разрабатываемый продукт сможет придать технологии привлекательный облик и сделать ее понятной и управляемой?

Как разрабатываемый продукт может решить проблемы пользователей?

Как разрабатываемый продукт поможет в достижении целей тем пользователям, которые редко работают с продуктом или имеют мало опыта?

Каким образом разрабатываемый продукт сможет удовлетворить запросы опытных пользователей, которым нужна функциональная мощь и глубина проработки?

Контрольные вопросы

1. Назовите три основных причины возникновения ошибок в проектировании.
2. Дайте характеристику целеориентированного проектирования.
3. Какие компоненты должны сочетать методы проектирования?
4. Назовите основные этапы целеориентированного проектирования.
5. Сформулируйте основные вопросы, которые возникают при описании и проектировании цифрового продукта.

4. МЕТОДЫ СБОРА ТРЕБОВАНИЙ

4.1. Значение качественных исследований

Оценивать результат усилий по проектированию следует, исходя из того, насколько успешно он отвечает требованиям как пользователей, так и компании — инициатора разработки.

Если у проектировщика нет ясного и детального представления о пользователях, для которых выполняется проектирование, если у него отсутствует понимание имеющихся ограничений, организационных задач и бизнес-целей, которые являются движущей силой разработки, то шансов на успех остается очень мало — неважно, насколько при этом хороши навыки и творческие способности проектировщика.

Глубокие знания о потенциальных пользователях можно получить лишь посредством качественных исследований.

Качественные исследования позволяют изучить:

поведение, взгляды, склонности потенциальных пользователей продукта; предметную область — технический и деловой контексты разрабатываемого продукта;

используемый лексикон и прочие социальные аспекты предметной области; способы применения существующих продуктов.

Качественные исследования способствуют ходу проектирования, поскольку:

обеспечивают доверие и уважение к команде проектировщиков (так как источник проектных решений можно проследить вплоть до результатов исследований);

объединяют команду разрабатываемого продукта общим для всех пониманием особенностей предметной области и проблем пользователей;

дают руководителям возможность принимать решения по тем или иным вопросам проектирования продукта на основе данных — вместо догадок и личных предпочтений.

Качественные методы в сравнении с количественными, как правило, оказываются более быстрыми, менее дорогостоящими и с большей вероятностью дают верные ответы на важные вопросы, без которых невозможно достичь превосходных результатов.

Вот эти вопросы:

Как продукт вписывается в более широкий контекст жизни пользователей?

Каковы основные цели при работе с продуктом, и какие базовые задачи позволяют достигать этих целей?

Какой опыт пользователи находят привлекательным? Как этот опыт соотносится с проектируемым продуктом?

С какими проблемами сталкиваются пользователи при использовании имеющихся технологий для решения своих задач?

Значение качественных исследований не ограничивается поддержкой процесса проектирования. Время, потраченное на то, чтобы увидеть за пользовательской аудиторией человеческие существа, способно принести плоды в виде важных для бизнеса прозрений, к которым невозможно прийти средствами традиционных исследований рынка.

Далее мы рассмотрим конкретные методы проведения качественных исследований, которые служат фундаментом целеориентированного проектирования.

4.2. Виды качественных исследований

Сосредоточим внимание на тех методиках, которые зарекомендовали себя в практической деятельности в последние годы. При этом постараемся представить эти техники с прагматической точки зрения, не углубляясь в теорию.

Вот перечень методик качественных исследований:

- 1) интервьюирование заинтересованных лиц;
- 2) интервьюирование экспертов в предметной области (ЭПО);
- 3) интервьюирование пользователей и покупателей;
- 4) анкетирование;
- 5) наблюдение за пользователями;
- 6) самостоятельное описание требований;
- 7) обзор литературы и технической документации;
- 8) аудит продукта/прототипа и конкурирующих решений;
- 9) совместные семинары.

Рассмотрим подробнее каждую из методик.

4.2.1. Интервьюирование заинтересованных лиц

Исследование, предваряющее проектирование любого нового продукта, должно начинаться с получения представления о техническом окружении и бизнес-контексте продукта. Практически всегда продукт проектируется (или перепроектируется) для достижения одной или нескольких конкретных бизнес-целей (как правило, речь идет об извлечении прибыли). Обязанность проектировщиков — создавать решения, не теряя из виду эти бизнес-цели, поэтому крайне важно, чтобы команда проектировщиков начинала работу с изучения возможностей и ограничений, стоящих за краткой спецификацией проекта.

В общем случае заинтересованное лицо — это любой человек, обладающий полномочиями в отношении проектируемого продукта и/или несущий ответственность за какой-либо его аспект. Говоря более конкретно, заинтересованные лица — это ключевые члены организации, иницилирующей работы по проекту.

Интервьюирование заинтересованных лиц должно проводиться до начала любых исследований пользовательской аудитории, поскольку возникающие обсуждения нередко задают способы проведения пользовательских исследований.

От заинтересованных лиц важно получить информацию по следующим вопросам:

- предварительное видение продукта;
- бюджет и график проекта;
- технические возможности и ограничения;
- потребности бизнеса;
- представления заинтересованных лиц о пользователях.

Понимание этих аспектов и того, как они влияют на проектные решения, способствует разработке удачных продуктов. Каким бы притягательным ни был спроектированный продукт для пользователей и клиентов, без учета осуществимости и жизнеспособности предложенного решения нет никаких шансов, что продукт преуспеет.

4.2.2. Интервьюирование экспертов в предметной области

На ранних стадиях проектирования неоценимый вклад часто дает выявление и интервьюирование нескольких экспертов в предметной области (ЭПО) — людей, сведущих в предметной области, на которую ориентирован проектируемый продукт. Многие ЭПО когда-то сами были пользователями текущей версии продукта или его прежних версий, а теперь могут быть преподавателями, менеджерами или консультантами. Часто это не сами заинтересованные лица, а нанятые ими эксперты.

Как и заинтересованные лица, ЭПО способны представить продукт и его пользователей под интересным углом зрения, однако проектировщикам следует проявлять осторожность и понимать, что точка зрения ЭПО в определенном смысле искажена.

Некоторые из вещей, которые следует знать о работе с ЭПО:

- ЭПО — это зачастую пользователи-эксперты;
- ЭПО хорошо осведомлены, но они не проектировщики;
- ЭПО необходимы в сложных или специализированных предметных областях, таких как медицина, наука или финансовые службы;
- с ЭПО понадобится общаться в течение всего процесса проектирования.

4.2.3. Интервьюирование покупателей

Очень легко спутать пользователей с покупателями. В случае с потребительскими продуктами покупатель и пользователь часто представлены в одном лице, однако в корпоративных и технических областях слова «пользова-

тели» и «покупатели» редко относятся к одной и той же группе людей. Хотя интервьюировать следует обе группы, у каждой из них будет собственная точка зрения на продукт, которую следует совершенно по-разному учитывать при принятии окончательных проектных решений.

Покупатели продукта — это люди, принимающие решение о его приобретении.

Если речь идет о потребительских продуктах, покупатели зачастую являются пользователями продукта. В случае с большинством технических, медицинских и производственных продуктов покупателем оказывается вовсе не пользователь — часто это кто-либо из руководства компании либо менеджер по информационным технологиям, имеющий свои цели и потребности, отличные от пользовательских.

Чтобы сделать продукт жизнеспособным, важно понимать покупателей и их цели. Не менее важно осознавать, что покупатели редко сами пользуются продуктом, а когда все же делают это, то совсем не так, как пользователи.

В интервью с покупателями необходимо понять:
каковы их цели в контексте приобретения продукта;
что их не устраивает в существующих решениях;
каков процесс принятия решения при покупке продуктов наподобие того, который вы проектируете;
их роль в установке, обслуживании и управлении продуктом;
проблемы предметной области и особенности используемой терминологии.

Подобно ЭПО, покупатели могут предлагать множество идей о том, как улучшить продукт.

4.2.4. Интервьюирование пользователей

Пользователи продукта в процессе проектирования должны находиться в центре внимания. Именно эти люди (а не их руководители или команда поддержки) лично пытаются добиться каких-то результатов с помощью продукта.

Если перепроектируется или улучшается существующий продукт, важно общаться не только с нынешними, но и с потенциальными пользователями, то есть с людьми, которые сегодня не пользуются продуктом, однако являются хорошими кандидатами на его использование в будущем, поскольку имеют потребности, удовлетворяемые продуктом, и входят в его целевую аудиторию.

Интервьюирование обеих категорий позволяет выявить то влияние, которое оказывает на поведение и образ мысли пользователя опыт работы с существующей версией продукта.

От пользователей необходимо получить следующую информацию:
контекст интеграции продукта (или аналогичной системы, если продукт еще не создан) в жизнь или рабочий процесс пользователей — когда, почему и каким образом применяется (или будет применяться) продукт;

осведомленность в предметной области с точки зрения пользователя — что необходимо знать пользователям, чтобы делать свою работу;

существующие задачи и виды деятельности — как те, которые выполняются при помощи данного продукта, так и те, которые не поддерживаются им; цели и мотивы использования продукта;

ментальная модель — как пользователи думают о своей работе и деятельности, а также чего они ожидают от продукта;

проблемы и сложности при работе с продуктом (или с аналогичной системой, если продукт еще не создан).

4.2.5. Анкетирование

Анкетирование — самый малозатратный для аналитика способ извлечения информации, он же — и наименее эффективный. Обычно применяется как дополнение к другим стратегиям выявления требований.

Недостатки анкетирования очевидны:

респонденты часто бывают неспособны либо слабо мотивированы в том, чтобы хорошо и информативно заполнить анкету;

велика вероятность получить неполную или вовсе ложную информацию.

Преимущество в том, что подготовка и анализ анкет требуют небольшого ресурса.

Рекомендуется формулировать в анкетах вопросы с замкнутым циклом ответов в одной из следующих трех форм:

многоальтернативные вопросы (могут расширяться комментариями респондента в свободной форме);

рейтинговые вопросы (представляют predetermined набор ответов на сформулированные вопросы с использованием таких значений, как «абсолютно согласен», «согласен», «отношусь нейтрально», «не согласен», «абсолютно не согласен», «не знаю»);

вопросы с ранжированием (предусматривает ранжирование (упорядочивание) ответов путем присваивания им порядковых номеров, процентных значений и т. п.).

4.2.6. Наблюдение за пользователями

Большинство людей не способны точно описать собственное поведение, особенно когда находятся вне контекста своей деятельности.

Из этого следует, что интервью, проводимое вне контекста ситуации, которую стремится понять проектировщик, даст менее полные и точные данные.

Можно обсудить с пользователями их представления о собственном поведении, а можно непосредственно наблюдать это поведение. Второй вариант дает лучшие результаты.

Виды наблюдения:

пассивное;

активное (при активном наблюдении аналитик работает как участник команды, что позволяет улучшить понимание процессов).

Через наблюдение, а возможно, и участие аналитики получают информацию о происходящих день за днем операциях из первых рук. Во время на-

блюдения за работой системы часто возникают вопросы, которые никогда бы не появились, если бы аналитик только читал документы или разговаривал с экспертами.

Недостаток этой стратегии в том, что наблюдатель, как и всякий «измерительный прибор», вносит помехи в результаты измерений: сотрудники организации, находясь «под колпаком», могут начать вести себя принципиально по-другому, чем обычно.

Вероятно, наиболее эффективным методом сбора качественных данных о пользователях является сочетание интервьюирования и наблюдения.

4.2.7. Самостоятельное описание требований

Документы — хороший источник информации, потому что они чаще всего доступны и их можно «опрашивать» в удобном для себя темпе. Чтение документов — прекрасный способ получить первоначальное представление о системе и сформулировать вопросы к экспертам.

Недостаток этой стратегии в опасности пропуска знаний, специфичных для объекта исследования (в случае самоопроса), либо неформализованных знаний, эмпирических правил и процедур, широко используемых на практике, но не вошедших в документы.

4.2.8. Обзор литературы и технической документации

Параллельно с интервьюированием заинтересованных лиц команде проектировщиков следует изучить какую-либо литературу, касающуюся продукта или его предметной области.

Команда проектировщиков должна собрать эту литературу и использовать ее как основу для формирования списка вопросов к заинтересованным лицам и ЭПО, а также в качестве источника дополнительных данных о предметной области и терминологии, а также для сравнения с уже собранными данными о пользователях.

Литература, касающаяся продукта или его предметной области должна включать:

- маркетинговые планы;
- стратегию бренда;
- исследования рынка;
- опросы пользователей;
- технические спецификации и информационные материалы;
- статьи в деловых и технических журналах, связанных с предметной областью;
- сравнительный анализ конкурентных решений;
- результаты поиска в Интернете похожих продуктов и новостей о них;
- результаты и метрики юзабилити-исследований;
- данные службы поддержки, такие как статистика обращений пользователей за поддержкой.

4.2.9. Аудит продукта/прототипа и конкурирующих решений

Часто оказывается полезным параллельно с интервьюированием заинтересованных лиц и ЭПО изучить любые существующие версии или прототипы продукта, а также его основных конкурентов. Тем самым команда проектировщиков получает хорошее представление о состоянии дел в области и почву для подготовки вопросов к интервью.

В идеале команде проектировщиков следует провести неформальную эвристическую или экспертную оценку интерфейсов как своего продукта, так и конкурирующих продуктов, проверяя их на соответствие принципам качественного взаимодействия и визуального дизайна. Данная процедура позволяет команде ознакомиться с сильными и слабыми сторонами доступных пользователю продуктов и дает общее представление о текущем объеме функциональности продукта.

4.2.10. Совместные семинары

Помимо классического интервью тет-а-тет, существует значительное количество методик, предполагающих широкое участие представителей заказчика и исполнителя.

Правила мозгового штурма предполагают полную раскрепощенность и свободу мнений, даже самых вычурных и на первый взгляд, бредовых. Первое правило мозгового штурма — полный запрет на любую критику. Всякое высказанное мнение представляет ценность, а полное отсутствие запретов позволяет полноценно подключить творческую фантазию.

Затем, на втором этапе, все высказанные мнения тщательным образом обсуждаются, заведомо неприемлемые варианты отсеиваются, формируются коллективные предложения.

Совместные семинары, сохраняя все преимущества режима интервью, приносят дополнительные бонусы: работа в группе более продуктивна, группы быстрее обучаются, более склонны к квалифицированным заключениям, позволяют исключить многие ошибки

Эта стратегия — одна из самых затратных, однако она окупается за счет меньшего количества ошибок и отказа от формализации в пользу живого общения, выработке общего языка и пр.

Контрольные вопросы

1. В чем преимущество качественных исследований?
2. Какие основные вопросы позволяют раскрыть качественные исследования?
3. Назовите основные виды качественных исследований.
4. Какими недостатками обладает анкетирование?
5. Назовите преимущества и недостатки совместных семинаров.

5. МОДЕЛИ БИЗНЕС-ПРОЦЕССОВ

Модели широко применяются в проектировании, разработке и науке. Это мощные инструменты для представления сложных структур и связей с целью их лучшего понимания, обсуждения и визуализации.

В отсутствие моделей мы были бы вынуждены осмысливать неструктурированные, сырые данные, не имея подспорья в виде каких-либо организующих принципов. Хорошие модели подчеркивают характерные особенности структур и связей и уводят в тень менее важные детали.

Моделирование бизнес-процессов при проектировании информационных систем позволяет отразить следующие аспекты:

- цель или желаемый результат процесса;
- частоту и важность процесса и каждого действия;
- событие, инициирующее процесс в целом и каждое действие в отдельности;
- зависимости — что требуется, чтобы выполнить процесс в целом и каждое действие в отдельности, а также какие события и действия зависят от завершения процесса в целом и каждого из действий;
- участников процессов, их роли и ответственности;
- конкретные действия;
- принимаемые решения;
- информацию, используемую при принятии решений;
- что может пойти не так — ошибки и исключительные ситуации;
- исправления ошибок и обработку исключений.

5.1. Методики моделирования бизнес-процессов

Рассмотрим коротко некоторые из методик моделирования бизнес-процессов.

Для сравнения удобства моделирования рассмотрим пример моделирования бизнес-процесса «Планирование закупок и размещение заказов поставщикам» с использованием каждой из вышеприведенных методик.

5.1.1. Функциональная методика IDEF0

Методику IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique). Исторически IDEF0 как стандарт был разработан в 1981 г. в рамках обширной программы автоматизации промышленных предприятий, которая носила обозначение ICAM (Integrated Computer Aided Manufacturing). Семейство стандартов IDEF унаследовало свое обозначение от названия этой программы (IDEF=Icam DEFinition), и последняя его редакция была выпущена в декабре 1993 г. Национальным институтом по стандартам и технологиям США (NIST).

Целью методики является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы.

В основе методики лежат четыре основных понятия: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий.

Функциональный блок (Activity Box) представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольной форме (например, «производить услуги»). На диаграмме функциональный блок изображается прямоугольником (рис. 6). Каждая из четырех сторон функционального блока имеет свое определенное значение (роль), при этом:

- верхняя сторона имеет значение «Управление» (Control);
- левая сторона имеет значение «Вход» (Input);
- правая сторона имеет значение «Выход» (Output);
- нижняя сторона имеет значение «Механизм» (Mechanism).



Рис. 6. Функциональный блок

Интерфейсная дуга (Arrow) отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют потоками или стрелками.

С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Таковыми объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т. д.) или потоки данных и информации (документы, данные, инструкции и т. д.).

В зависимости от того, к какой из сторон функционального блока подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или «управляющей».

Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую, так как каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла.

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram).

Декомпозиция (Decomposition) является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбивке сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Последним из понятий IDEF0 является глоссарий (Glossary). Для каждого из элементов IDEF0 — диаграмм, функциональных блоков, интерфейсных дуг — существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т. д., характеризующих объект, отображенный данным элементом. Этот набор называется глоссарием и является описанием сущности данного элемента. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Модель IDEF0 всегда начинается с представления системы как единого целого — одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется контекстной диаграммой.

В пояснительном тексте к контекстной диаграмме должна быть указана цель (Purpose) построения диаграммы в виде краткого описания и зафиксирована точка зрения (Viewpoint).

Определение и формализация цели разработки IDEF0-модели является крайне важным моментом. Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь.

Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдель-

ных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему. Правильный выбор точки зрения существенно сокращает временные затраты на построение конечной модели.

Выделение подпроцессов. В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней (Child Diagram) по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме, соответственно называется дочерним блоком — Child Box). В свою очередь, функциональный блок-предок называется родительским блоком по отношению к дочерней диаграмме (Parent Box), а диаграмма, к которой он принадлежит, — родительской диаграммой (Parent Diagram). Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока. В каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели.

Иногда отдельные интерфейсные дуги высшего уровня не имеет смысла продолжать рассматривать на диаграммах нижнего уровня, или наоборот — отдельные дуги нижнего отражать на диаграммах более высоких уровней, так как это будет только перегружать диаграммы и делать их сложными для восприятия. Для решения подобных задач в стандарте IDEF0 предусмотрено понятие «туннелирование». «Туннель» (Arrow Tunnel) в виде двух круглых скобок вокруг начала интерфейсной дуги обозначает, что эта дуга не была унаследована от функционального родительского блока и появилась (из «туннеля») только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца (стрелки) интерфейсной дуги в непосредственной близости от блока-приемника устанавливает тот факт, что в дочерней по отношению к этому блоку диаграмме эта дуга отображаться и рассматриваться не будет. Чаще всего бывает, что отдельные объекты и соответствующие им интерфейсные дуги не рассматриваются на некоторых промежуточных уровнях иерархии. В таком случае они сначала «погружаются в туннель», а затем при необходимости «возвращаются из туннеля».

Обычно IDEF0-модели несут в себе сложную и концентрированную информацию, и для того чтобы ограничить их перегруженность и сделать удобочитаемыми, в стандарте приняты соответствующие ограничения сложности.

Рекомендуется представлять на диаграмме от трех до шести функциональных блоков, при этом количество подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг предполагается не более четырех.

Стандарт IDEF0 содержит набор процедур, позволяющих разрабатывать и согласовывать модель большой группой людей, принадлежащих к разным областям деятельности моделируемой системы.

Обычно процесс разработки является итеративным и состоит из следующих условных этапов:

1. Создание модели группой специалистов, относящихся к различным сферам деятельности предприятия. Эта группа в терминах IDEF0 называется авторами (Authors). Построение первоначальной модели является динамическим процессом, в течение которого авторы опрашивают компетентных лиц о структуре различных процессов, создавая модели деятельности подразделений. При этом их интересуют ответы на следующие вопросы:

Что поступает в подразделение на входе?

Какие функции и в какой последовательности выполняются в рамках подразделения?

Кто является ответственным за выполнение каждой из функций?

Чем руководствуется исполнитель при выполнении каждой из функций?

Что является результатом работы подразделения (на выходе)?

На основе имеющихся положений, документов и результатов опросов создается черновик (Model Draft) модели.

2. Распространение черновика для рассмотрения, согласований и комментариев. На этой стадии происходит обсуждение черновика модели с широким кругом компетентных лиц (в терминах IDEF0 — читателей) на предприятии. При этом каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передается автору. Автор, в свою очередь, также письменно соглашается с критикой или отвергает ее с изложением логики принятия решения и вновь возвращает откорректированный черновик для дальнейшего рассмотрения. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.

3. Официальное утверждение модели. Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности. Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.

Наглядность графического языка IDEF0 делает модель вполне читаемой и для лиц, которые не принимали участия в проекте ее создания, а также эффективной для проведения показов и презентаций. В дальнейшем на базе построенной модели могут быть организованы новые проекты, нацеленные на производство изменений в модели.

5.1.2. Описание бизнес-процессов в Унифицированном языке моделирования

В конце 90-х гг. XX в. был разработан Унифицированный язык моделирования (Unified Modeling Language, UML), который является объектно-ориентированным графическим языком для визуализации, спецификации, конструирования и документирования систем, в которых большая роль принадлежит программному обеспечению.

Процесс создания программного обеспечения по методике Rational Unified Process компании Rational Software Corporation включает следующие этапы:

- моделирование предметной области (Business Modeling);
- определение требований к системе (Requirements);
- анализ и проектирование (Analysis&Design);
- разработка (Implementation);
- тестирование (Test);
- внедрение (Deployment).

Моделирование бизнес-процессов, для автоматизации которых разрабатывается ПО, производится на этапе моделирования предметной области с использованием диаграмм вариантов использования (Usecasediagram) и деятельности (Activitydiagram).

Применение необходимо ограничить, в основном, первыми уровнями декомпозиции бизнес-процессов. Излишняя детализация усложняет чтение и сопровождение модели.

В UML выделяют девять типов диаграмм:

- 1) диаграммы классов (Classdiagram);
- 2) диаграммы объектов (Objectdiagram);
- 3) диаграммы вариантов использования (Usecasediagram);
- 4) диаграммы последовательностей (Sequencediagram);
- 5) диаграммы кооперации (Collaborationdiagram);
- 6) диаграммы состояний (Statediagram);
- 7) диаграммы деятельности (Activitydiagram);
- 8) диаграммы компонентов (Componentdiagram);
- 9) диаграммы развертывания (Deploymentdiagram).

Для описания бизнес-процессов в UML применяются диаграммы вариантов использования и деятельности.

Диаграммы деятельности — это частный случай диаграммы состояний, на ней представлены переходы потока управления от одной деятельности к другой внутри системы. Диаграммы деятельности относятся к динамическому виду системы; они наиболее важны при моделировании ее функционирования и отражают поток управления между объектами.

Диаграмма деятельности может использоваться также для определения подразделений и функций участников бизнес-процесса, описания документооборота.

При описании бизнес-процессов диаграммы деятельности используются для описания последовательности различных действий субъектов и объектов.

5.1.3. ДРАКОН-технология

ДРАКОН: Дружелюбный Русский Алгоритмический язык, Который Обеспечивает Наглядность (DRAKON)

Визуальный алгоритмический язык программирования и моделирования ДРАКОН был разработан в рамках космической программы «Буран». Разра-

ботка языка велась с 1986 г. при участии Федерального космического агентства (Научно-производственный центр автоматики и приборостроения им. акад. Н. А. Пилюгина, Москва) и Российской академии наук (Институт прикладной математики им. М. В. Келдыша) [5]. Основной задачей разработчиков было создание единого универсального языка программирования и моделирования, который своей доступностью и мощностью способен заменить специализированные языки, используемые в этих организациях для разработки и моделирования. Язык построен путем формализации, эргономизации и неклассической структуризации блок-схем алгоритмов, описанных в ГОСТ 19.701—90 и ISO 5807—85.

Как в любом графическом языке, в ДРАКОНе используются два типа элементов:

графические фигуры (икон);

текстовые надписи, расположенные внутри или снаружи икон.

Поэтому язык ДРАКОН имеет не один, а два синтаксиса: графический и текстовый. Графический (визуальный) синтаксис охватывает алфавит икон, правила их размещения в поле чертежа и правила связи икон с помощью соединительных линий. Текстовый синтаксис задает алфавит символов, правила их комбинирования и привязку к иконам (которая необходима, так как внутри разных икон используются разные типы выражений)

Основой графического синтаксиса языка ДРАКОН является графический алфавит. Алфавит состоит из графических элементов (графических фигур), которые называются иконами. В языке ДРАКОН имеется 27 икон (рис. 7).

Для каждой иконы задана ориентация, однозначно показано направление соединительных линий, входов и выходов. Благодаря жестко заданной ориентации икон и соединительных линий в большинстве случаев отпадает необходимость использовать стрелки.

ДРАКОН имеет не только иконы, но и макроиконы. Макроиконы — это графические слова языка ДРАКОН. Подобно тому, как слова слагаются из букв, макроиконы (графические слова) состоят из икон (графических букв). В языке ДРАКОН имеется 21 макроикона (рис. 8). Иконы и макроиконы — это строительные блоки, из которых создаются ДРАКОН-схемы.

Важной частью макроикон служат валентные точки (на рисунке они показаны как маленькие черные кружки). В эти точки последовательно вводятся иконы и макроиконы, которые в совокупности образуют графический узор и (после заполнения икон текстом) превращаются в ДРАКОН-схему.

Подробное описание разработки ДРАКОН-схем приведено в работах В. Д. Паронджанова [5, 6].

В целом ДРАКОН позволяет описать алгоритм работы программы, системы реального времени, технологического процесса, формализовать алгоритм работы экспертов и описывать бизнес-процессы.

Особенности использования языка ДРАКОН при описании бизнес-процессов будут рассмотрены ниже.

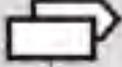
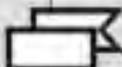
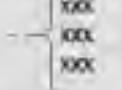
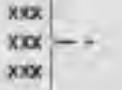
	Икона	Название иконы		Икона	Название иконы
И1		Заголовок	И14		Вывод
И2		Конец	И15		Ввод
И3		Действие	И16		Пауза
И4		Вопрос	И17		Период
И5		Выбор	И18		Пуск таймера
И6		Вариант	И19		Синхронизатор (по таймеру)
И7		Имя ветки	И20		Параллельный процесс
И8		Адрес	И21		Комментарий
И9		Вставка	И22		Правый комментарий
И10		Полка	И23		Левый комментарий
И11		Формальные параметры	И24		Петля цикла
И12		Начало цикла для	И25		Петля силуэта
И13		Конец цикла для			

Рис. 7. Иконы языка ДРАКОН

	Макроикона	Название макроикона
1		Заголовок с параметрами
2		Развилка
3		Переключатель (число вариантов $N \geq 2$)
4		Обычный цикл
5		Переключающий цикл
6		Цикл ДЛЯ
7		Цикл ЖДАТЬ
8		Действие по таймеру
9		Полка по таймеру
10		Развилка по таймеру
11		Переключатель по таймеру
12		Обычный цикл по таймеру
13		Переключающий цикл по таймеру
14		Цикл ДЛЯ по таймеру
15		Цикл ЖДАТЬ по таймеру
16		Вставка по таймеру
17		Вывод по таймеру
18		Ввод по таймеру
18		Запуск таймера по таймеру
20		Параллельный процесс по таймеру

Рис. 8. Макроикона языка ДРАКОН

5.2. Описание бизнес-процесса «Планирование закупок и размещение заказов поставщикам»

Опишем бизнес-процесс «Планирование закупок и размещение заказов поставщикам» с использованием вышеописанных методик.

Предприятие планирует закупки медикаментов. Планирование закупок осуществляется в Департаменте маркетинга, в группе маркетинга и планирования. Планирование закупок осуществляется следующим образом:

1. Менеджер группы планирования и маркетинга ежедневно получает от контрагентов данные внешней и внутренней статистики продаж медикаментов в виде отчетов продаж.

2. Для планирования закупок медикаментов менеджер группы планирования и маркетинга еженедельно на основании статистики продаж осуществляется расчет потребности в товаре. В результате расчета формируется *таблица потребностей в товаре*.

3. Определив количество и номенклатуру заказываемых товаров, менеджер отдела закупок приступает к анализу предложений поставщиков. Данный процесс осуществляется ежемесячно или по мере необходимости. Выбираются наиболее выгодные условия поставки. Для этого сравниваются цены поставщиков. Данные сведения берутся из прайс-листа для закупок. При выборе поставщика важно учесть предоставляемую отсрочку платежа. Данные сведения берутся из контрактов, отмеченных как приоритетные (действующие). В результате формируется *список поставщиков*, каждой позиции присваивается признак основного и запасных поставщиков в порядке убывания приоритета.

4. Менеджер отдела закупок ежемесячно на основании таблицы потребностей в товаре и списка выбранных поставщиков формирует *графики поставок* с указанием сроков и периодичности, но без количества поставки.

5. Ежемесячно после определения потребности в товаре менеджер группы логистики рассчитывает *необходимое количество закупок*. Необходимое количество закупок рассчитывается на основании фактических запасов на складе, необходимого минимального и максимального уровня запасов. Нормы минимального и максимального количества запасов устанавливаются в днях. При расчете необходимого количества закупки учитывается также время товара в пути. Таким образом, данный расчет должен обеспечить возможность бесперебойного наличия товара на складе. По результату расчетов формируется *план заявок на месяц*.

6. Затем в группе логистики ежедневно по плану заявок, графику поставок, прайс-листам поставщиков формируются *заказы поставщикам*.

7. Если предстоит сделать заказ импортному поставщику, то менеджер группы логистики рассчитывает *затраты на сертификацию*, создается отчет о затратах на сертификацию. Затраты на сертификацию проверяются на соответствие внутрифирменным нормам. Данная операция производится по мере необходимости.

8. Если затраты на сертификацию превышают внутрифирменные нормы, то менеджер группы логистики повторяет процесс формирования заказов поставщикам. Формируются новые заказы.

9. Ежедневно подготовленный заказ поставщику акцептуется, заказ должен подписать менеджер по логистике и директор Департамента маркетинга и управления товарными запасами.

10. Ежедневно менеджер группы логистики направляет заказ в отдел закупок. Менеджер отдела закупок направляет заказ поставщику.

Ниже проведены результаты моделирования этого бизнес-процесса с использованием вышеприведенных методологий.

На основании приведенного описания предметной области, используя методологию IDEF0, можно построить функциональную модель, описывающую основные бизнес-процессы.

Построение модели IDEF0 всегда начинается с построения контекстной диаграммой, представленной на рис. 9.

Далее произведено выделение подпроцессов. Получившаяся диаграмма второго уровня, содержащая функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы представлена на рис. 10.

Далее были детализированы подфункции дочерней диаграммы. Результаты детализации представлены на рис. 11, 12, 13.

Для выполнения моделирования бизнес-процесса с использованием языка UML необходимо выполнить следующие действия:

- выявить действующие лица и варианты использования разрабатываемой системы;

- построить диаграмму вариантов использования;

- написать сценарии реализации вариантов использования и/или построить диаграмму деятельности.

Анализ бизнес-процесса «Планирование закупок и размещение заказов поставщикам» позволяет выделить следующие действующие лица, взаимодействующие с системой:

- менеджер группы планирования;

- менеджер отдела закупок;

- менеджер группы логистики.

Далее определяем варианты использования, т. е. множества возможных последовательностей действий (событий), приводящих к значимому для действующего лица результату:

- определение потребности в товаре;

- анализ поставщиков;

- формирование заказа поставщику;

- направление заказа поставщику;

Установив ассоциации между действующими лицами и вариантами использования, получим диаграмму вариантов использования представленную на рис. 14.

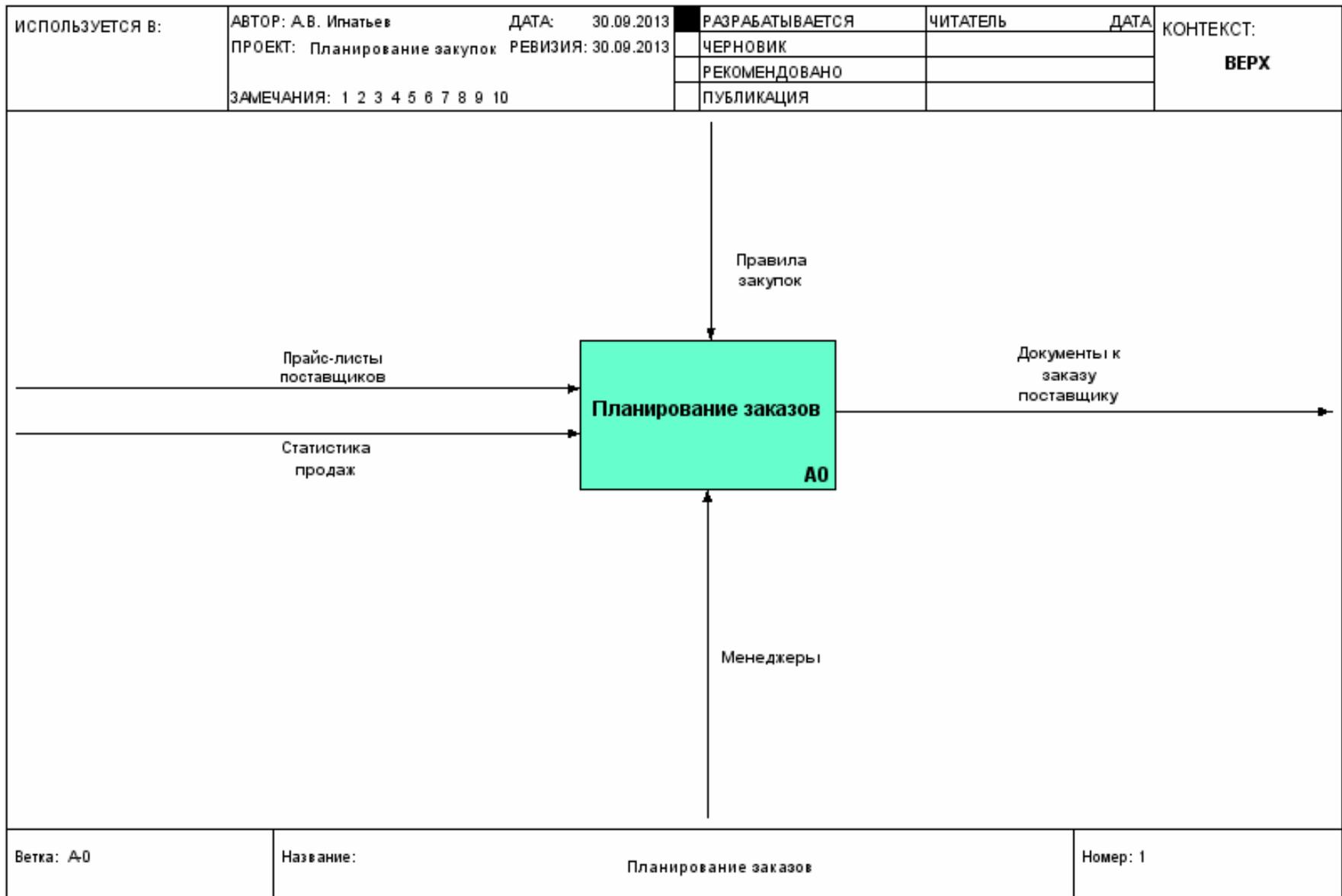


Рис. 9. Контекстная диаграмма бизнес-процесса «Планирование закупок и размещение заказов поставщикам»

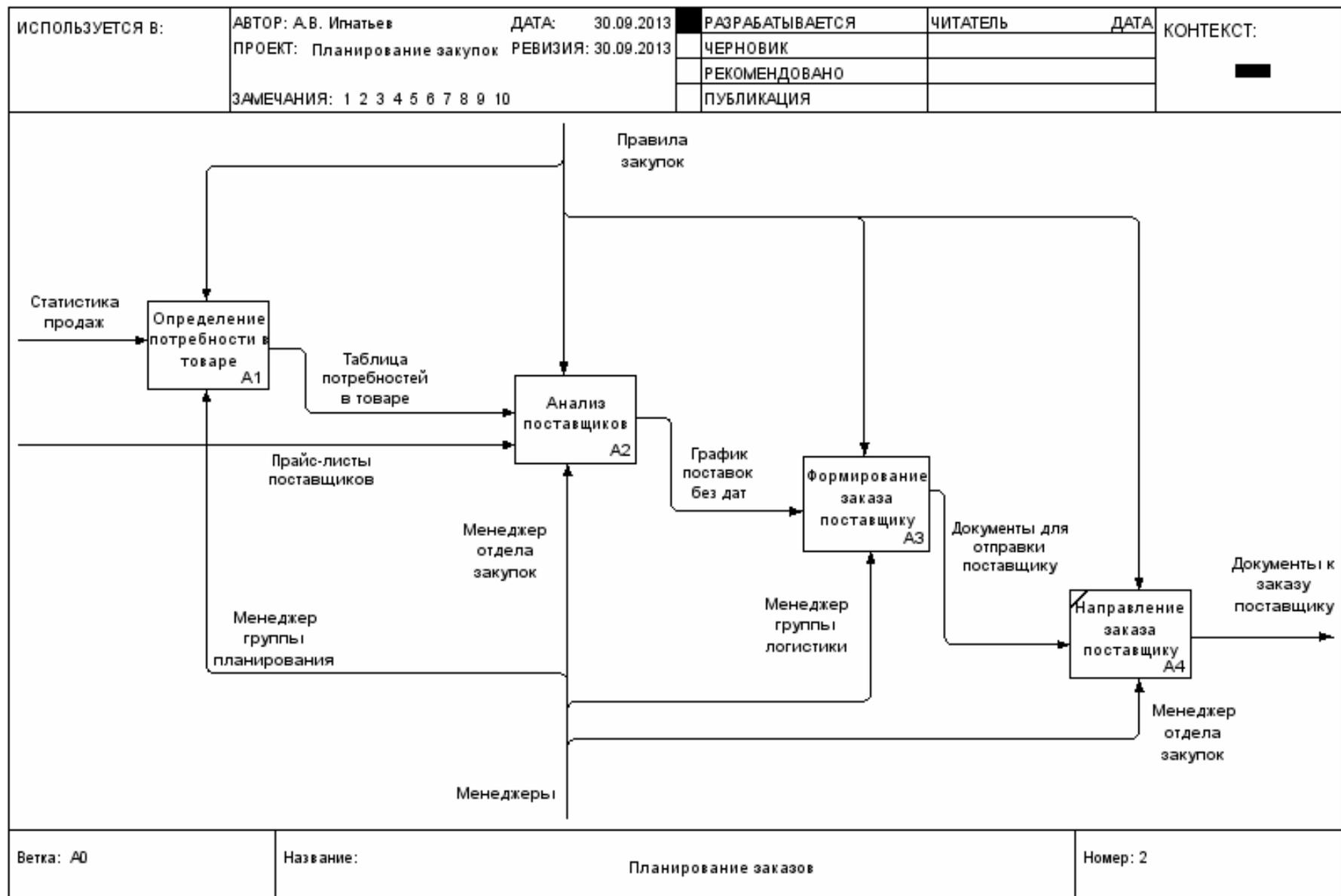


Рис. 10. Диаграмма декомпозиции бизнес-процесса «Планирование закупок и размещение заказов поставщикам»

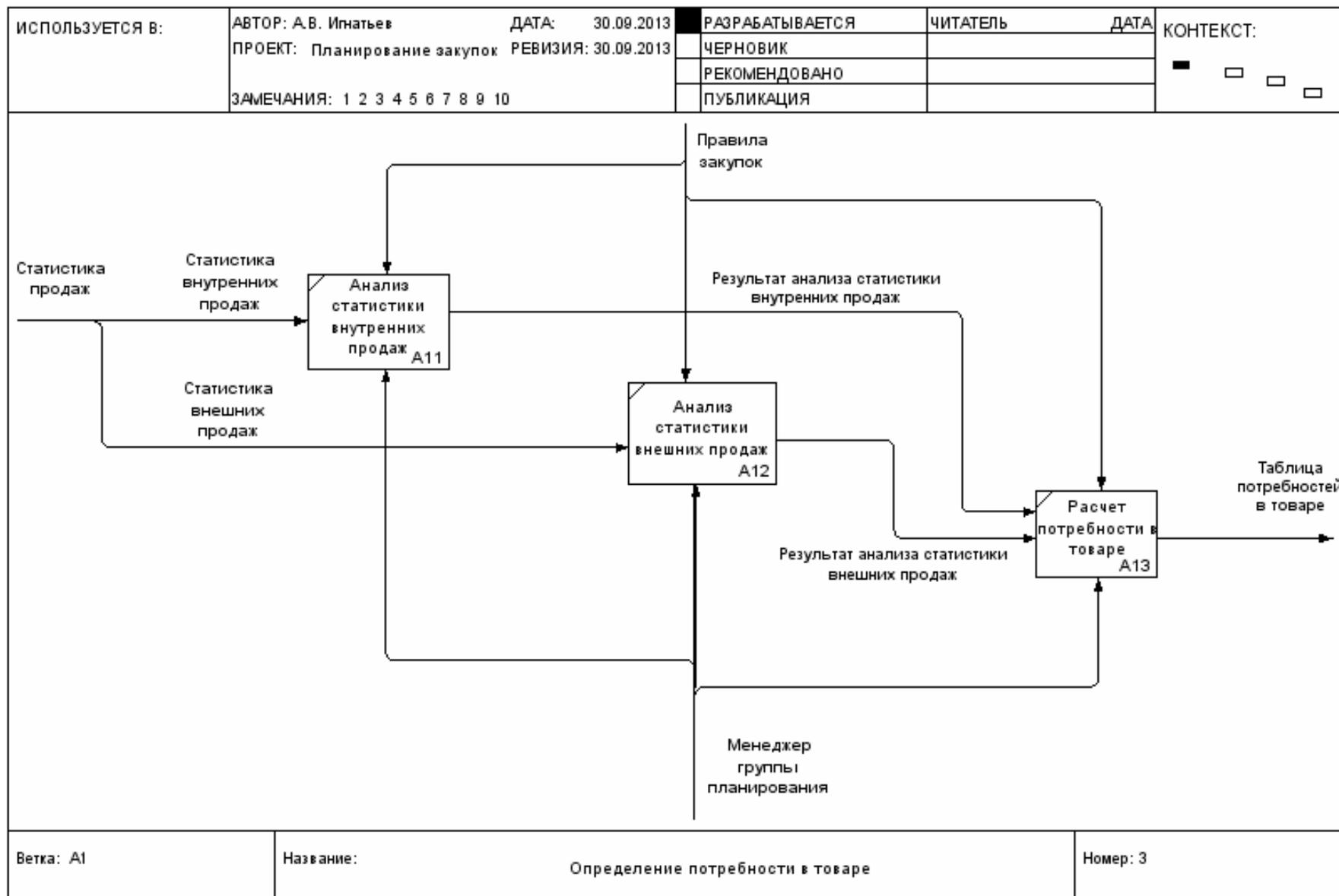


Рис. 11. Диаграмма декомпозиции бизнес-процесса «Определение потребности в товаре»

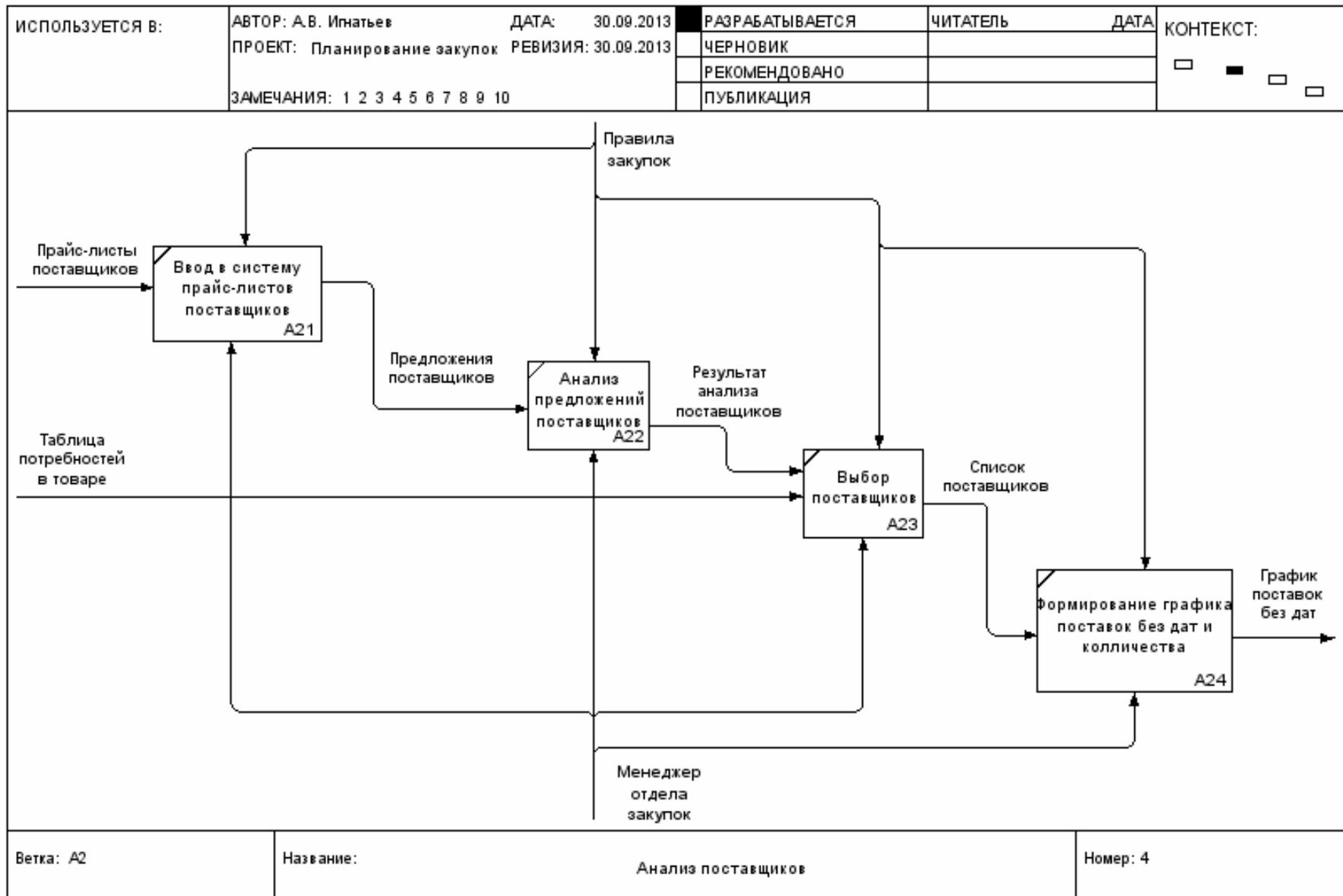


Рис.12. Диаграмма декомпозиции бизнес-процесса «Анализ поставщиков»

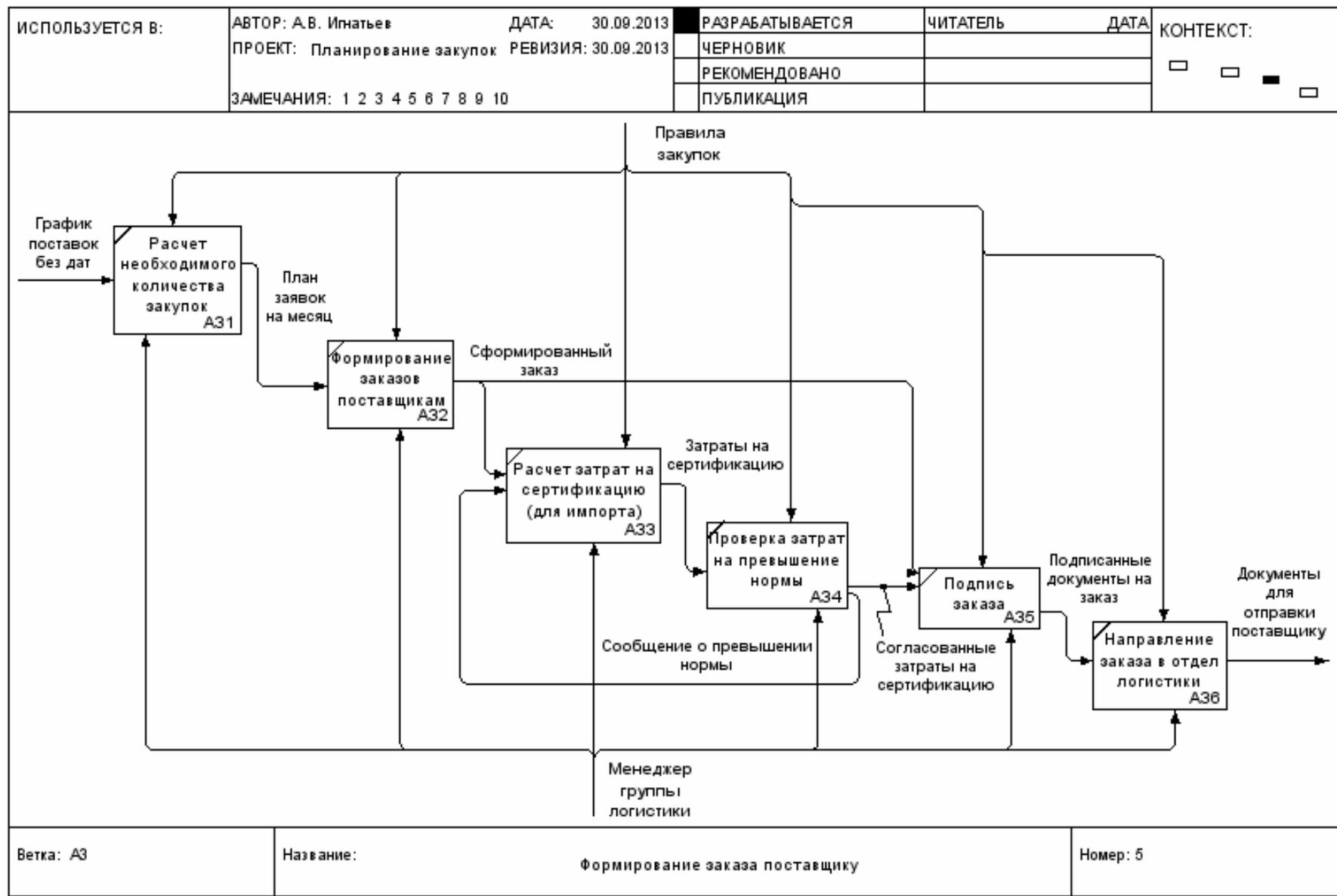


Рис. 13. Диаграмма декомпозиции бизнес-процесса «Формирование заказа поставщику»

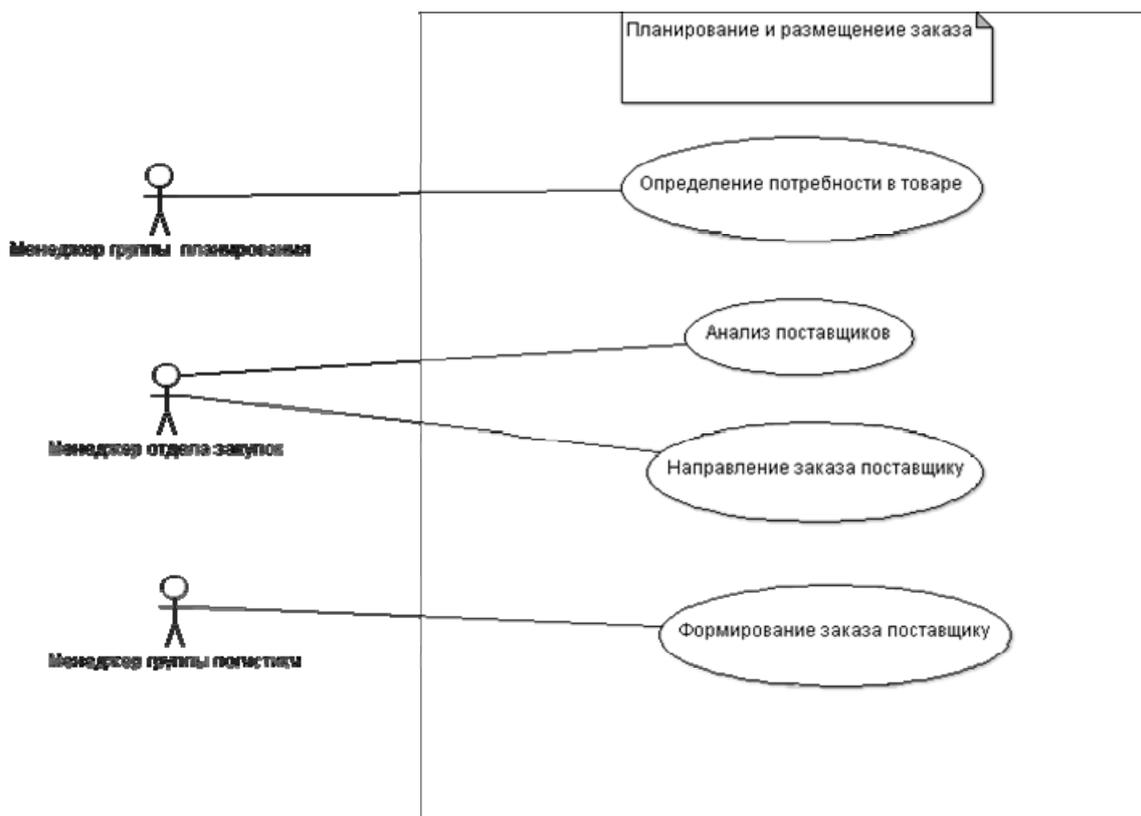


Рис. 14. Диаграмма вариантов использования в бизнес-процессе «Планирование закупок и размещение заказов поставщикам»

Поскольку в нашем учебном примере описание бизнес-процесса уже разбито на шаги, то написание сценариев для каждого из вариантов использования не представляет труда. Достаточно разбить общее описание процесса на этапы, соответствующие вариантам использования. В реальных проектах процесс написания сценариев для вариантов использования является более сложным трудоемким.

Для бизнес-процесса «Планирование закупок и размещение заказов поставщикам» могут быть написаны следующие сценарии:

Сценарий варианта использования: «Определение потребности в товаре»:

1. Менеджер группы планирования и маркетинга ежедневно получает от контрагентов данные внешней и внутренней статистики продаж медикаментов в виде отчетов продаж.

2. Для планирования закупок медикаментов менеджер группы планирования и маркетинга еженедельно на основании статистики продаж осуществляется расчет потребности в товаре. В результате расчета формируется *таблица потребностей в товаре*.

Сценарий варианта использования: «Анализ поставщиков»:

1. Определив количество и номенклатуру заказываемых товаров, менеджер отдела закупок приступает к анализу предложений поставщиков. Данный процесс осуществляется ежемесячно или по мере необходимости. Выбираются наиболее выгодные условия поставки. Для этого сравниваются це-

ны поставщиков. Данные сведения берутся из прайс-листа для закупок. При выборе поставщика важно учесть предоставляемую отсрочку платежа. Данные сведения берутся из контрактов, отмеченных как приоритетные (действующие). В результате, формируется список поставщиков, каждой позиции присваивается признак основного и запасных поставщиков в порядке убывания приоритета.

2. Менеджер отдела закупок ежемесячно на основании таблицы потребностей в товаре и списка выбранных поставщиков формирует графики поставок с указанием сроков и периодичности, но без количества поставки.

Сценарий варианта использования: «Формирование заказа поставщику»:

1. Ежемесячно после определения потребности в товаре менеджер группы логистики рассчитывает необходимое количество закупок. Необходимое количество закупок рассчитывается на основании фактических запасов на складе, необходимого минимального и максимального уровня запасов. Нормы минимального и максимального количества запасов устанавливаются в днях. При расчете необходимого количества закупки учитывается также время товара в пути. Таким образом, данный расчет должен обеспечить возможность бесперебойного наличия товара на складе. По результату расчетов формируется план заявок на месяц.

2. Затем в группе логистики ежедневно по плану заявок, графику поставок, прайс-листам поставщиков формируются заказы поставщикам.

3. Если предстоит сделать заказ импортному поставщику, то менеджер группы логистики рассчитывает затраты на сертификацию, создается отчет о затратах на сертификацию. Затраты на сертификацию проверяются на соответствие внутрифирменным нормам. Данная операция производится по мере необходимости.

4. Если затраты на сертификацию превышают внутрифирменные нормы, то менеджер группы логистики повторяет процесс формирования заказов поставщикам. Формируются новые заказы.

5. Ежедневно подготовленный заказ поставщику акцептуется, заказ должен подписать менеджер по логистике и директор Департамента маркетинга и управления товарными запасами.

6. Ежедневно менеджер группы логистики направляет заказ в отдел закупок.

Сценарий варианта использования: «Направление заказа поставщику»:

Менеджер отдела закупок направляет заказ поставщику.

Диаграмма деятельности, описывающая полный сценарий работы, представлена на рис. 15.

Рассмотрим теперь описание того же бизнес-процесса с использованием языка ДРАКОН.

Для описания входов, выходов и участников бизнес-процесса будем дополнительно использовать язык ГНОМ (Графический Наглядный Очень Маленький язык). Подробное его описание приведено в [6].

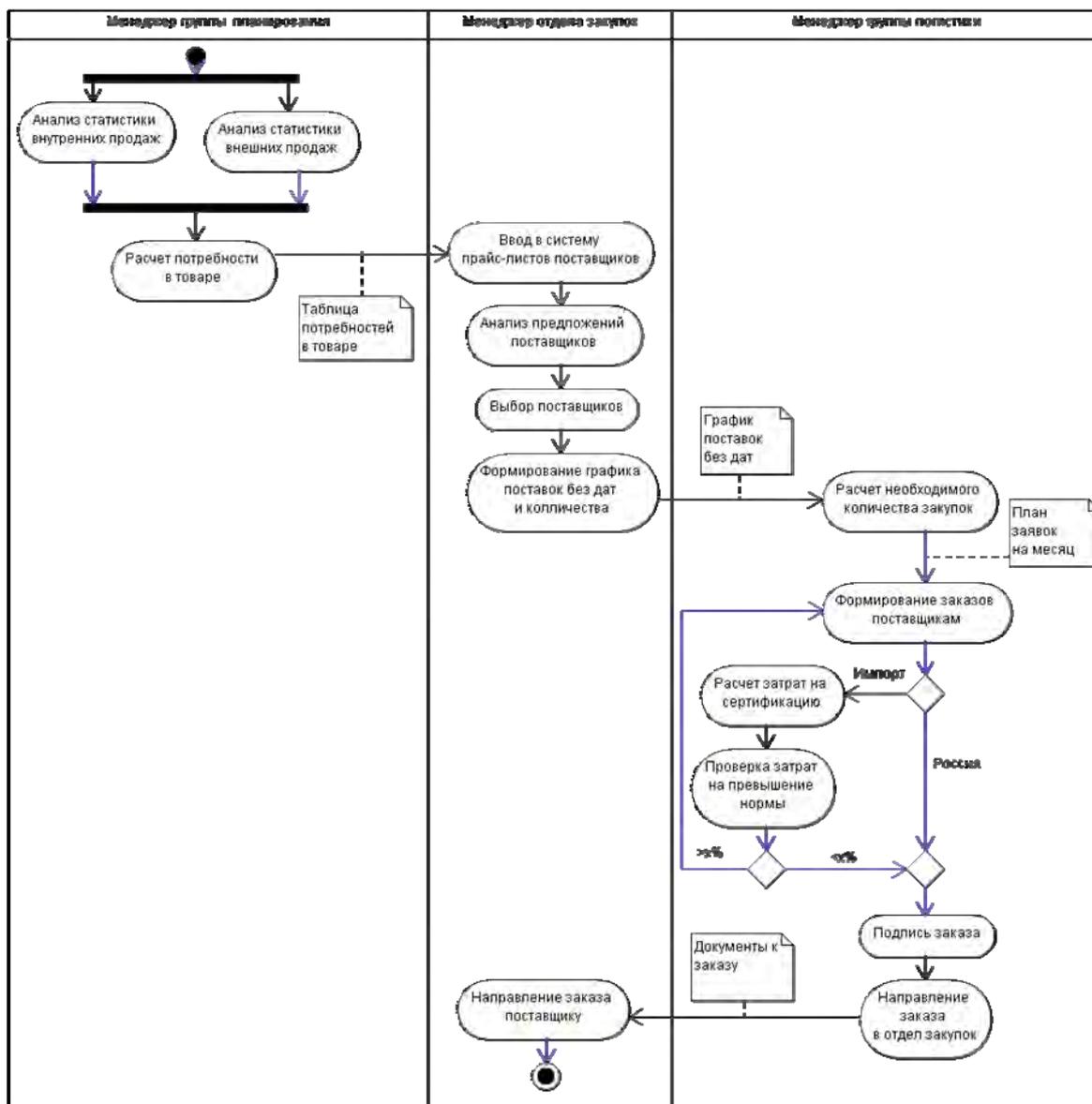


Рис. 15. Диаграмма деятельности бизнес-процесса «Планирование закупок и размещение заказов поставщикам»

В указатели поместим названия основных подпроцессов, а в мнемоблоки — их описание (участников, входы и выходы). ГНОМ-схема бизнес-процесса «Планирование закупок и размещение заказов поставщикам» показана на рис. 16.

Далее построим ДРАКОН-схему бизнес-процесса. Ветки ДРАКОН-схемы — это основные подпроцессы моделируемого бизнес-процесса «Планирование закупок и размещение заказов поставщикам». Икону «Ввод» используем для описания входной информации (от кого и какие документы поступили). Для описания выходной информации используется икона «Вывод».

Операции, выполняемые при реализации подпроцесса, описываем с помощью иконы «Полка». В верхней части иконы помещаем название исполнителя, а в нижней — название операции.

Если требуется указать периодичность операции, то используем правый комментарий. Для указания дополнительных источников информации используем левый комментарий.

Результат построения ДРАКОН-схемы бизнес-процесса «Планирование закупок и размещение заказов поставщикам» представлен на рис. 17.

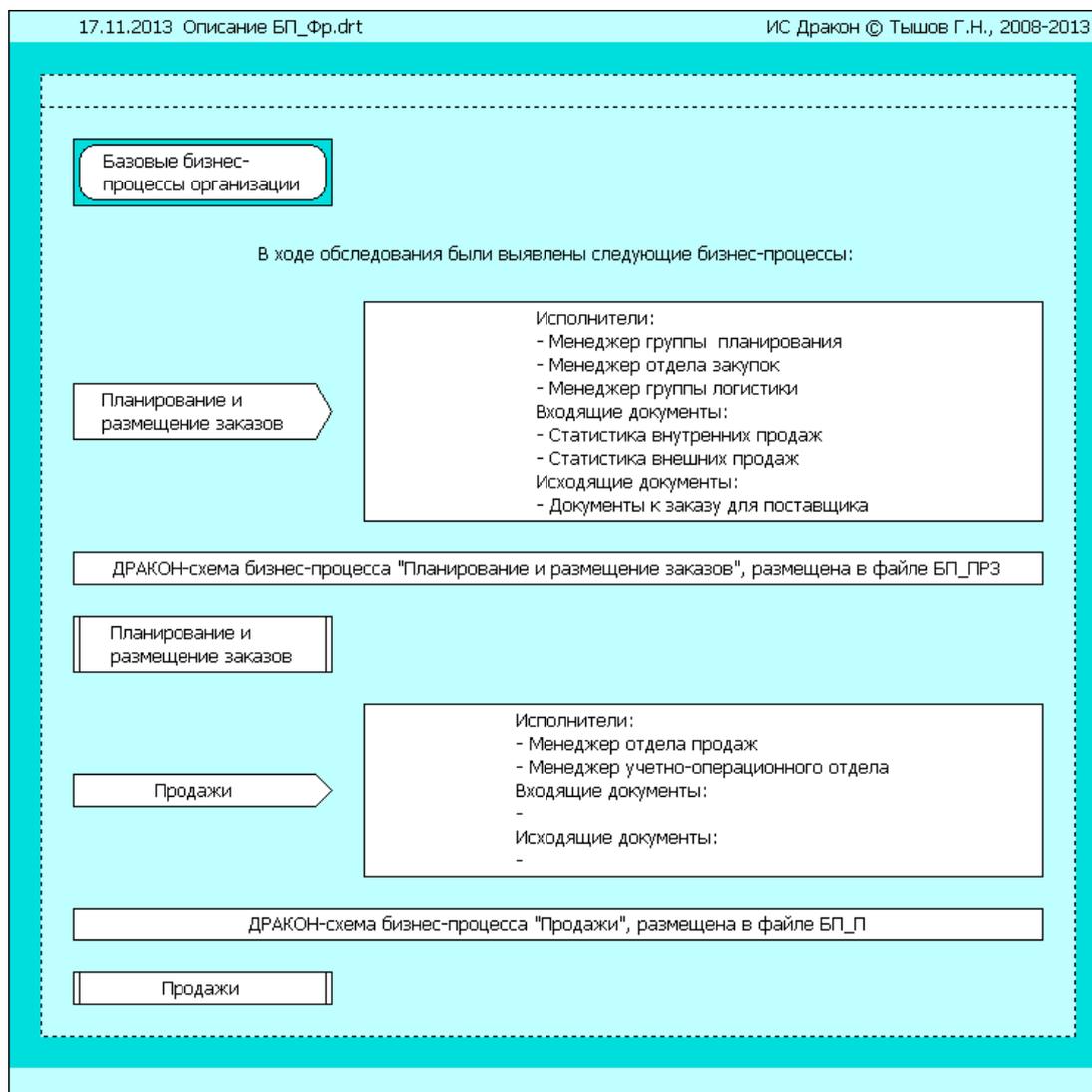


Рис. 16. ГНОМ-схема бизнес-процесса «Планирование закупок и размещение заказов поставщикам»

Сравнение трех вариантов описания бизнес-процесса «Планирование закупок и размещение заказов поставщикам» позволяет сделать некоторые выводы об удобстве их использования.

Наиболее неудобной является методология IDEF0, поскольку в ней отсутствует ветвление. Использование диаграмм деятельности UML и ДРАКОН-схем позволяет избавиться от этого недостатка.

Сравнивая диаграммы деятельности и ДРАКОН-схемы, следует отметить, что в первых выделяются действия отдельного исполнителя, а во вторых — подпроцессы. Второй подход позволяет добиться максимальной наглядности графического описания бизнес-процесса.

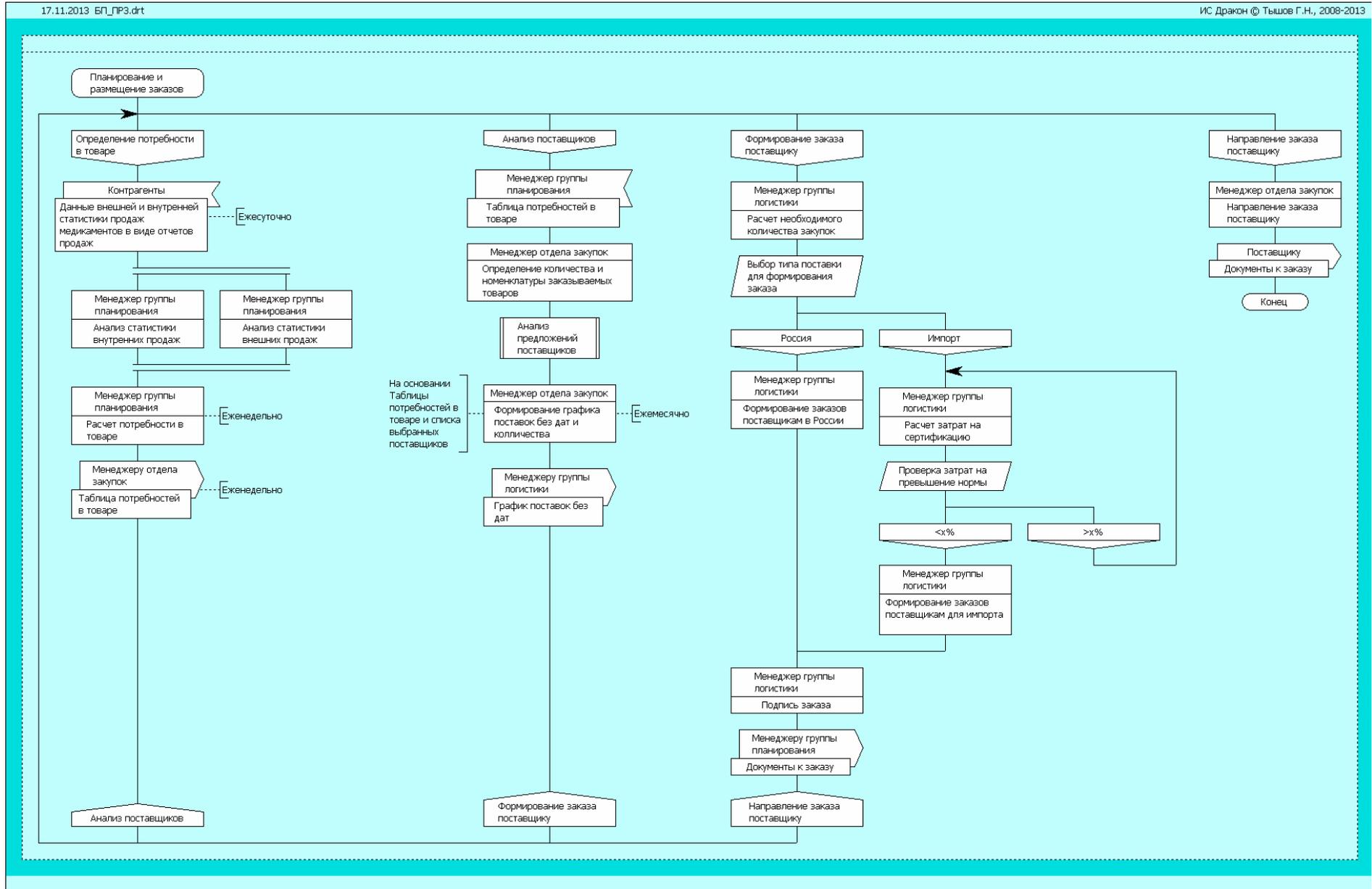


Рис. 17. ДРАКОН-схема бизнес-процесса «Планирование закупок и размещение заказов поставщикам»

Контрольные вопросы

1. В чем преимущество качественных исследований?
2. Какие основные вопросы позволяют раскрыть качественные исследования?
3. Назовите основные виды качественных исследований.
4. Какими недостатками обладает анкетирование?
5. Назовите преимущества и недостатки совместных семинаров.

6. МЕТОД ПЕРСОНАЖЕЙ

6.1. Кто такие персонажи?

Поскольку мы проектируем для пользователей, важно суметь понять и визуализировать характерные аспекты их отношений друг с другом, социальной и физической средой и, конечно, продуктом, который мы намереваемся спроектировать.

Проектировщики должны создавать модели пользователей на основе данных наблюдений за их поведением, интуитивно выявляя общие зависимости. Лишь после формализации найденных зависимостей можно надеяться на систематическое конструирование таких шаблонов взаимодействия пользователя с продуктом, которые будут хорошо соответствовать поведению, ментальным моделям и целям пользователей. Такую формализацию обеспечивают персонажи.

Логика может подсказать, что для создания продукта, который призван удовлетворить самую большую аудиторию пользователей, нужно сделать его функциональность как можно более широкой, чтобы приспособить продукт к нуждам людей. Однако эта логика порочна. Лучший способ успешно удовлетворить потребности широкой аудитории — проектировать для конкретных типов людей с конкретными потребностями.

Решение состоит в том, чтобы сначала правильно выбрать представителей, чьи потребности лучше всего отражают интересы большей части интересующей нас аудитории, а затем назначить приоритеты этим представителям так, чтобы учет потребностей самых важных пользователей не приносил существенных неудобств второстепенным. Персонажи представляют собой мощный инструмент для обсуждения различных типов пользователей и их потребностей, а также для выбора наиболее важных пользователей, на которых следует ориентироваться при проектировании функциональности и поведения продукта.

Персонажи — действенное и универсальное средство проектирования, способствующее преодолению ряда проблем, которые в настоящее время неотступно преследуют разработчиков цифровых продуктов. Персонажи помогают проектировщикам определять, что должен делать продукт и каким

должно быть его поведение; общаться с заинтересованными лицами, разработчиками и другими проектировщиками; оценивать эффективность решений.

Кроме того, персонажи позволяют решить три ключевые проблемы проектирования, возникающие при разработке продукта.

Как любая модель, персонажи должны основываться на наблюдениях за реальностью. Основным источником данных при синтезе персонажей служат интервью, контекстное исследование либо другие формы обсуждений с действительными и потенциальными пользователями, а также наблюдения за ними. Качество данных, собранных в ходе этого процесса напрямую определяет действенность персонажей как средства, позволяющего упростить и направить процесс проектирования. Прочие данные, способные помочь в процессе создания персонажей и уточнить информацию о них, включают в себя (в примерном порядке убывания эффективности):

данные интервью с пользователями вне контекста использования;

информацию о пользователях, предоставленную заинтересованными лицами и экспертами в предметной области (ЭПО);

результаты обзора литературы и более ранних исследований.

Персонажи — это модели пользователей, представленные в виде конкретных людей. Они не являются реальными людьми, но синтезируются непосредственно по результатам наблюдений за реальными людьми.

Успех персонажей как моделей пользователей во многом определяется тем, что персонажи олицетворяют личности.

Хотя персонажи изображаются как конкретные личности (поскольку выполняют функцию архетипов), каждый из них является представителем класса или типа пользователей конкретного интерактивного продукта.

Персонаж вбирает в себя уникальный набор шаблонов поведения, связанных с использованием определенного продукта (или с аналогичной деятельностью в предметной области, если продукт еще не существует), которые выявляются посредством анализа данных интервью и при необходимости подкрепляются дополнительными количественными данными. Наряду с конкретными мотивами и целями эти шаблоны задают персонажей. Иногда персонажей называют также составными архетипами пользователей, поскольку они создаются путем группировки родственных шаблонов использования, наблюдавшихся в ходе фазы исследований у пользователей, играющих сходные роли.

6.2. Разработка персонажей

Как уже говорилось ранее, персонажи возникают на основе общих тенденций, выявленных в ходе интервью с пользователями и потенциальными пользователями (а иногда и покупателями) продукта, а также в процессе наблюдений за ними. Пробелы в этих данных восполняются путем дополнительных исследований, а также с помощью информации, представленной ЭПО и заинтересованными лицами и почерпнутой из существующей литературы.

Создание правдоподобных и действенных персонажей в равной степени требует детального анализа и творческого синтеза. Оба вида деятельности в значительной степени выигрывают от применения стандартизированного процесса.

Вот основные шаги процесса:

1. Выявить поведенческие переменные.
2. Сопоставить респондентов с поведенческими переменными.
3. Выявить значимые шаблоны поведения.
4. Синтезировать характеристики и соответствующие им цели.
5. Проверить полноту и выявить избыточность.
6. Расширить описание атрибутов и поведений.
7. Назначить персонажам типы.
8. Рассмотрим каждый из этапов более подробно.

Шаг 1. Выявление поведенческих переменных

После завершения исследования и проведения предварительной обработки данных необходимо перечислить самостоятельные аспекты наблюдавшихся вариантов поведения, то есть создать набор поведенческих переменных.

В общем случае наиболее важные различия между шаблонами поведения определяются при рассмотрении следующих типов переменных:

деятельность: чем занят пользователь, частота и объем;

взгляды: каким образом пользователь думает о предметной области и технологии продукта;

наклонности: каковы образование и подготовка пользователя, его способность обучаться;

мотивация: каким образом пользователь вовлечен в предметную область продукта;

навыки: умения пользователя, связанные с предметной областью продукта и используемой технологией.

Шаг 2. Сопоставление респондентов с поведенческими переменными

Каждому респонденту назначается соответствующее место в диапазоне каждой переменной. Некоторые из переменных будут отражать непрерывный диапазон поведения (к примеру, от новичка до эксперта в компьютерной области), а некоторые — дискретные варианты выбора (скажем, использование цифрового либо пленочного фотоаппарата).

Здесь важна не столько точность значений, сколько взаимное расположение респондентов. Результатом этого шага должна стать группировка всех респондентов по каждой из осей.

Шаг 3. Выявление значимых шаблонов поведения

Разместив респондентов по осям, необходимо найти группы (кластеры) отдельных респондентов, близких сразу по нескольким диапазонам или переменным. Группа респондентов, кластеризованная сразу по шести-восьми различным переменным, вероятнее всего, представляет значимый шаблон

поведения, который ляжет в основу персонажа. У некоторых специализированных ролей может быть лишь один значимый шаблон, однако обычно таких шаблонов два или даже три.

Чтобы шаблон считался корректным, должна существовать логическая или причинно-следственная связь между кластеризованными поведением, а не просто случайная корреляция.

Шаг 4. Синтез характеристик и соответствующих им целей

Для каждого выявленного значимого шаблона поведения необходимо синтезировать детали на основе собранных данных. Нужно описать среду потенциального использования, типичный рабочий день (или иной подходящий контекст), существующие решения задач и причины недовольства пользователей этими решениями, а также значимые отношения с окружающими.

На этом этапе достаточно простого перечисления различных характеристик поведения, представленного в сжатой форме.

Шаг 5. Проверка полноты и выявление избыточности

Если обнаружится, что два персонажа различаются лишь демографическими характеристиками, то можно исключить одного из них или подстроить его характеристики таким образом, чтобы персонажи стали более непохожими. Каждый персонаж должен отличаться от всех прочих по меньшей мере одним значимым вариантом поведения.

Шаг 6. Расширение описания атрибутов и поведений

Перечень характеристик и целей, полученный на четвертом шаге, очерчивает сущность сложного поведения, однако оставляет многие вещи нераскрытыми. Типичное описание персонажа — это синтез наиболее важных деталей, полученных в ходе исследований и относящихся к этому персонажу.

Такое описание становится очень действенным средством коммуникации. В идеальном случае описание персонажа содержит большую часть находок исследования пользовательской аудитории. Тем самым исследования напрямую питают деятельность по проектированию.

Шаг 7. Назначение персонажам типов

К этому моменту персонажи должны восприниматься уже как настоящие, знакомые люди. Последний шаг в конструировании персонажа завершает процесс превращения качественных данных исследования в набор мощных средств проектирования.

У проектирования всегда должна быть цель — та аудитория, на которой сосредоточен процесс проектирования. И как правило, чем конкретнее обозначена эта цель, тем лучше. Создание интерфейсного решения, отвечающего потребностям хотя бы трех-четырех персонажей одновременно, уже может оказаться весьма сложной задачей. Посему необходимо назначить персонажам приоритеты, определяющие, кто из них станет основной целью проектирования. Задача в том, чтобы выбрать из набора того персонажа, чьи

нужды и цели можно полностью удовлетворить посредством одного интерфейса, не забывая при этом об остальных персонажах. Задача решается назначением персонажам типов. Существует пять типов, которые располагаются обычно в таком порядке:

- ключевой;
- второстепенный;
- дополнительный;
- покупатель;
- отрицательный.

Рассмотрим эти типы и их значимость с позиций проектирования:

1. *Ключевые* персонажи задают основную цель в проектировании интерфейса. На один интерфейс в продукте может приходиться только один ключевой персонаж, однако для некоторых продуктов (в особенности для корпоративного пользования) возможно наличие нескольких различных интерфейсов, каждый из которых ориентирован на отдельного ключевого персонажа.

2. *Второстепенный* персонаж в основном оказывается доволен интерфейсом ключевого персонажа, но имеет дополнительные потребности, которые можно включить в продукт, не нарушая его способности служить ключевому персонажу.

3. *Дополнительные* (пользовательские) персонажи не являются ни ключевыми, ни второстепенными. Их нужды полностью представлены сочетанием нужд ключевого и второстепенных персонажей, их полностью удовлетворяет один из ключевых интерфейсов. На каждый интерфейс может приходиться любое количество дополнительных персонажей. Зачастую ими становятся политические персонажи — те, которых включили в набор, чтобы учесть предположения заинтересованных лиц.

4. Персонажи *покупателей* отражают потребности покупателей, а не конечных пользователей, как уже обсуждалось ранее. Обычно персонажи покупателей используются в качестве второстепенных персонажей. Однако в некоторых корпоративных средах кто-то из таких персонажей может оказаться ключевым, если ему предназначается собственный административный интерфейс.

5. *Отвергаемые* персонажи используются, чтобы продемонстрировать заинтересованным лицам и участникам разработки, что существуют пользователи, для которых продукт не предназначен.

Контрольные вопросы

1. В чем преимущество метода персонажей?
2. Какие источники информации используются при создании персонажей?
3. Дайте определение персонажа.
4. Перечислите основные этапы создания персонажей.
5. Назовите основные типы персонажей.

Список использованной литературы

1. Макарова Н. В., Волков В. Б. Информатика : учеб. для вузов. — СПб. : Питер, 2011. — 576 с.
2. Мацяшек Л. А., Лионг Б. Л. Практическая программная инженерия на основе учебного примера / пер. с англ. — М. : БИНОМ. Лаборатория знаний, 2009. — 956 с.
3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы / пер. с англ. — 2-е изд., доп. и испр. — М. : Символ, 2010. — 304 с.
4. Купер А., Рейман Р., Кронин Д. Алан Купер об интерфейсе. Основы проектирования взаимодействия / пер. с англ. — СПб. : Символ-Плюс, 2009. — 688 с.
5. Паронджанов В. Д. Дружелюбные алгоритмы, понятные каждому. Как улучшить работу ума без лишних хлопот. — М. : ДМК Пресс, 2010. — 464 с.
6. Паронджанов В. Д. Почему мудрец похож на обезьяну или Парадоксальная энциклопедия современной мудрости. — М. : РИПОЛ-классик, 2007. — 1154 с.

Список рекомендуемой литературы

1. Брукс, Ф. Проектирование процесса проектирования: записки компьютерного эксперта / Ф. Брукс ; пер. с англ. — М. : И. Д. Вильямс, 2013. — 464 с.
2. Варфел, Т. З. Прототипирование. Практическое руководство / Т. З. Варфел ; пер. с англ. — М. : Манн, Иванов и Фербер, 2013. — 240 с.
3. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс ; пер. с англ. — М. : Издательско-торговый дом «Русская Редакция», 2004. — 576 с.
4. Де Марко, Т. Вальсируя с Медведями: управление рисками в проектах по разработке программного обеспечения / Т. Де Марко, Т. Листер ; пер. с англ. — М. : Компания p.m.Office, 2005. — 208 с.
5. Кесенбери, У. Сторителлинг в проектировании интерфейсов. Как создавать истории, улучшающие дизайн / У. Кесенбери, К. Брукс ; пер. с англ. — М. : Манн, Иванов и Фербер, 2013. — 336 с.
6. Коберн, А. Современные методы описания функциональных требований к системам / А. Коберн ; пер. с англ. — М. : Лори, 2011. — 263 с.
7. Кулябов, Д. С. Введение в формальные методы описания бизнес-процессов : учеб. пособие / Д. С. Кулябов, А. В. Королькова. — М. : РУДН, 2008. — 173 с.
8. Купер, А. Психбольница в руках пациентов. Алан Купер об интерфейсах: почему высокие технологии сводят нас с ума и как восстановить душевное равновесие / А. Купер ; пер. с англ. — М. : Символ-Плюс, 2011. — 336 с.
9. Новиков, Ф. А. Моделирование на UML. Теория, практика, видеокурс / Ф. А. Новиков, Д. Ю. Иванов. — СПб. : Профессиональная литература. Наука и Техника, 2010. — 640 с.

Учебное электронное издание

Игнатъев Александр Владимирович

**МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

Учебное пособие

Начальник РИО *М. Л. Песчаная*

Зав. редакцией *О. А. Шипунова*

Редактор *И. Б. Чижикова*

Компьютерная правка и верстка *А. Г. Сиволобова*

Минимальные систем. требования:

PC 486 DX-33; Microsoft Windows XP; Internet Explorer 6.0; Adobe Reader 6.0.

Подписано в свет 16.10.2014.

Гарнитура «Таймс». Уч.-изд. л. 2,5. Объем данных 0,9 Мбайт.

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Волгоградский государственный архитектурно-строительный университет»

Редакционно-издательский отдел
400074, Волгоград, ул. Академическая, 1
<http://www.vgasu.ru>, info@vgasu.ru