

УДК 004.738.5

Г. Р. Гумерова

ФБГОУ ВО «Казанский национальный исследовательский технический университет им. А.Н. Туполева», Набережные Челны, e-mail: grgumerova@kai.ru

Т. Г. Мансурова

Набережночелнинский институт ФГАОУ ВО «Казанский (Приволжский) федеральный университет», Набережные Челны, e-mail: mansurovatg@yandex.ru

МОДЕЛИРОВАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Ключевые слова: анализ предметной области, требования к программному продукту, разработка требований, моделирование требований, виды требований, свойства требований, диаграммы вариантов использования, описание вариантов использования, BPMN.

В статье рассматривается подход к разработке требований к программному обеспечению, базирующийся на применении комбинации методов выявления требований, а именно, комплексной визуализации аспектов проектируемой предметной области. Конкретизируются свойства, предъявляемые к требованиям, уточняется группировка требований, учитывающая функциональный и нефункциональный характер использования. Для каждой категории требований даны шаблоны формализации и примеры правильной формулировки. Процесс моделирования требований к предметной области представлен практическим описанием проектирования предметной области обслуживания читателей библиотеки. Моделирование требований визуализировано диаграммами вариантов использования и моделирования бизнес-процессов в нотации BPMN. Для рассматриваемой предметной области представлено сравнительное описание вариантов использования и пользовательских историй, конкретизированы принципиальные отличия в их формулировке. В заключении представлен перечень выявленных требований к разработке продукта рассматриваемой предметной области.

G. R. Gumerova

Kazan National Research Technical University named after A.N. Tupolev, Naberezhnye Chelny, e-mail: grgumerova@kai.ru

T. G. Mansurova

Naberezhnye Chelny Institute, Kazan (Volga Region) Federal University, Naberezhnye Chelny, e-mail: mansurovatg@yandex.ru

MODELING SOFTWARE REQUIREMENTS

Keywords: domain analysis, software product requirements, requirements development, requirements modeling, types of requirements, properties of requirements, use case diagrams, description of use cases, BPMN.

The article considers the approach to software requirements development based on the application of a combination of requirements elicitation methods, namely, complex visualization of aspects of the designed subject area. The properties of requirements are specified, the grouping of requirements that takes into account the functional and non-functional nature of use is defined more precisely. Formalization templates and examples of correct formulation are given for each category of requirements. The process of requirements modeling to the subject area is presented by the practical description of designing the subject area of library readers' service. Requirements modeling is visualized by diagrams of use cases and business process modeling in BPMN notation. A comparative description of use cases and user stories for the reviewed subject area are presented, the principal differences in their formulation are specified. The conclusion presents a list of identified requirements for product development of the reviewed subject area.

Изречение римского философа Л. Сенеки о том, что «для человека не знающего, к какой пристани он держит путь, ни один ветер не будет попутным» как никогда свидетельствует об актуальности вопросов формулировки требований [6, с. 182]. Так, не имея

четких требований к разработке программного обеспечения невозможно создать качественный программный продукт, удовлетворяющий требованиям всех стейкхолдеров.

Согласно определению IEEE под требование к разработке программного обеспече-

ния – это сумма трех составляющих: функционал, требуемый пользователю; функционал, удовлетворяемый продуктом разработчи- ки и документальное представление данного функционала. Требования определяют цели разработки, потребности в автоматизации, проблемы, решение которых предполагается после автоматизации и ограничения. Требования очерчивают границы проекта, как компас и карта, которые помогут прийти до намеченной цели.

А. Девис, специалист в области требований, утверждает, что ни одно представление требований в каком-то едином виде не демонстрирует полноценную картину проекта. Поэтому необходима комбинация текстовых и графических способов представления требований на различных уровнях представлений, для формирования целостной картины создаваемой системы [1, с. 260].

К. Вигерс в предложенной классификации выделяет требования функционального и нефункционального характера. Функциональные требования определяют то, что система, программное обеспечение или приложение должно выполнять, основные требования к функциональности системы. Требования нефункционального характера определяют какими атрибутами должна обладать система для корректной работы.

В свою очередь требования функционального характера представлены бизнес-требованиями, пользовательскими и функциональными требованиями. Бизнес-требования говорят о выгоде для заказчика, в формулировке требований употребляются контекст «автоматизация», «увеличение прибыли», «снижение издержек», например, необходимо автоматизировать процесс формирования оперативной отчетности всех подразделений предприятия [4, с. 18]. В формулировке бизнес-требований нет конкретики, вся формулировка в общем виде. Это связано с тем, что бизнес-требования содержат высокоуровневые цели организации или заказчиков системы и определяют вектор разработки в целом.

Пользовательские требования описывают необходимый функционал для решения своих задач, например, как пользователь с ролью «Кассир», я должен формировать отчет по продажам в конце каждого рабочего дня. Пользовательские требования занимают нишу между бизнес-требованиями, которые формулируют цели проекта, и функциональными требованиями, которые

ставят задачи реализации разработчикам. Формулировка функционального требования будет выглядеть следующим образом: система должна предоставлять возможность поиска авиабилетов без указания конкретных дат вылета или прилета.

Таким образом, каждый последующий вид требований – это детализация предыдущего вида, то есть бизнес-требование задает высокоуровневую цель разработки, далее пользовательское требование определяет задачи пользователя, которые должны быть решены в результате автоматизации, а функциональные требования указывают как это должно быть реализовано.

Требования нефункционального характера делятся на бизнес-правила, системные требования, атрибуты качества, внешние системы и интерфейсы и ограничения [1, с. 235]. Бизнес-правила сами по себе не являются требованиями к программному обеспечению, но они накладывают ограничения и регламентирующее воздействие на разработку, они базируются на директивных документах, например, таких как корпоративные политики, законы и постановления, распоряжения, промышленные стандарты, вычислительные алгоритмы и т.п. Пример формулировки бизнес-правила: перерегистрация читателя в библиотечной системе должна соответствовать правилам пользования библиотекой, описанным в «Инструкции по регистрации читателей и учету посещений пользователей».

Системные требования описывают высокоуровневые требования автоматизации системы или компонента системы, например, доступ к личному кабинету пользователя для web-версии и мобильного приложения должен обеспечиваться через единый сервер приложений.

Атрибуты качества отражают описание дополнительных характеристик продукта, например, таких как легкость и простота использования, эффективность и устойчивость к сбоям и т.п. Атрибуты качества отражают необязательные системные требования, их можно рассматривать как конкурентное преимущество, как дополнительный фактор в пользу выбора приложения среди аналогов-конкурентов. В целом атрибуты качества отражают удобство использования системы, например, время загрузки главной страницы не должно превышать трех секунд при нагрузке до сорока посетителей в минуту.

Внешние системы и интерфейсы, основное назначение таких требований – определить формат взаимодействия между системой и пользователем, другой системой или оборудованием, например, взаимодействие системы с сервером компании-контрагента должно быть осуществлено с помощью протокола HTTPS.

Требования типа ограничения – это условия, сужающие выбор возможных решений, модифицируя требование или набор требований [1, с. 198]. Обычно это внешние по отношению к системе условия или принятые ранее решения, например, сайт должен корректно отображаться в браузере Яндекс.

Каждый вид требований собираются из разных источников на различных этапах работы над проектом, имеют различные цели и аудиторию и должны документироваться по-разному. Для одной автоматизируемой функции дается описание множеств видов требования к ней. Требования будет много, при взгляде на них должно быть понятно, что за функция, как выполняется ее результат, какой критерий перехода в одно из состояний и т.д. Таким образом, состав требований к разработке данной функции должен быть исчерпывающим. Проверка того, насколько хорошо, качественно были сформулированы требования предполагает соответствие ряду свойств каждого требования: полноты, ясности, корректности и согласованности, верифицируемости и необходимости и осуществимости.

Цикл работы с требованием включает процедуры валидации и верификации. Под валидацией подразумевается процесс взаимодействия со стейкхолдерами проекта на предмет корректности требований, а именно, доказательство и подтверждение того, что требования конкретного пользователя, к разрабатываемому приложению, к системе или компоненту системы удовлетворены. Верифицируемость (возможность проверки, установления истинности утверждений) включает проверку внутренних свойств набора требований: если требование изложено на языке, понятном и одинаково воспринимаемом участниками процесса создания информационной системы, причем оно является полным, т.е. ни одна из важных для реализации деталей не упущена – значит, это требование можно проверить, а именно, провести процедуру верификации. Для этого вырабатывается набор «проверок» сценариев, который может быть выполнен для того, чтобы удостовериться,

что разработка работает так как надо согласно представленным требованиям.

Свойство необходимости и осуществимости определяется разумным балансом между ценностью (степенью необходимости и полезности) и потребляемыми ресурсами. То есть затраты на реализацию требования к программному продукту не должны быть выше, чем эффект, полученный в результате использования продукта.

Для построения целостного набора требований используются методы моделирования требования. Комплексная визуализация представлений диаграмм различного вида создадут целостную картину, границ разрабатываемого продукта и способствуют определению всех существенных взаимоотношений и связей между элементами системы, выработке требований, удовлетворяющих всем условиям. Комплексная визуализация предполагает построения диаграмм вариантов использования, диаграмм процессов в нотациях BPMN и IDEF0, диаграмм деятельности, диаграмм классов [2, с. 70].

Моделирование требования будет рассмотрено на примере разработки требований к проектированию компонента системы управления библиотечной системой – модуля обслуживания читателей библиотеки.

Диаграмма «сущность-связь» моделирует отношения между данными, детализируя объекты предметной области, их связи и характер этих связей (рис. 1). ER-модель отображает семантические связи между объектами предметной области, поэтому диаграмму рекомендуется применять в качестве инструмента анализа требований, такой анализ позволяет выявить и связать компоненты данных проблемной области [1, с. 262]. Создание ER-модели в ходе разработки продукта позволяет определить логику структуры системы, поэтому такое представление реализации расширяет и дополняет понимание системы. Диаграмма сущность-связь демонстрирует характер связей между сущностями, что позволяет выявить и выделить необходимую функциональность разрабатываемого продукта. На рис.1 показано, что между сущностями «Формуляр читателя» и «Штрафы библиотеки» существует связь «назначается», поэтому необходима зафиксировать в виде требования функциональность, которая предоставит пользователю доступ к истории нарушений читателем условий библиотеки, например, зафиксированную в форме прецедента или пользовательской истории.

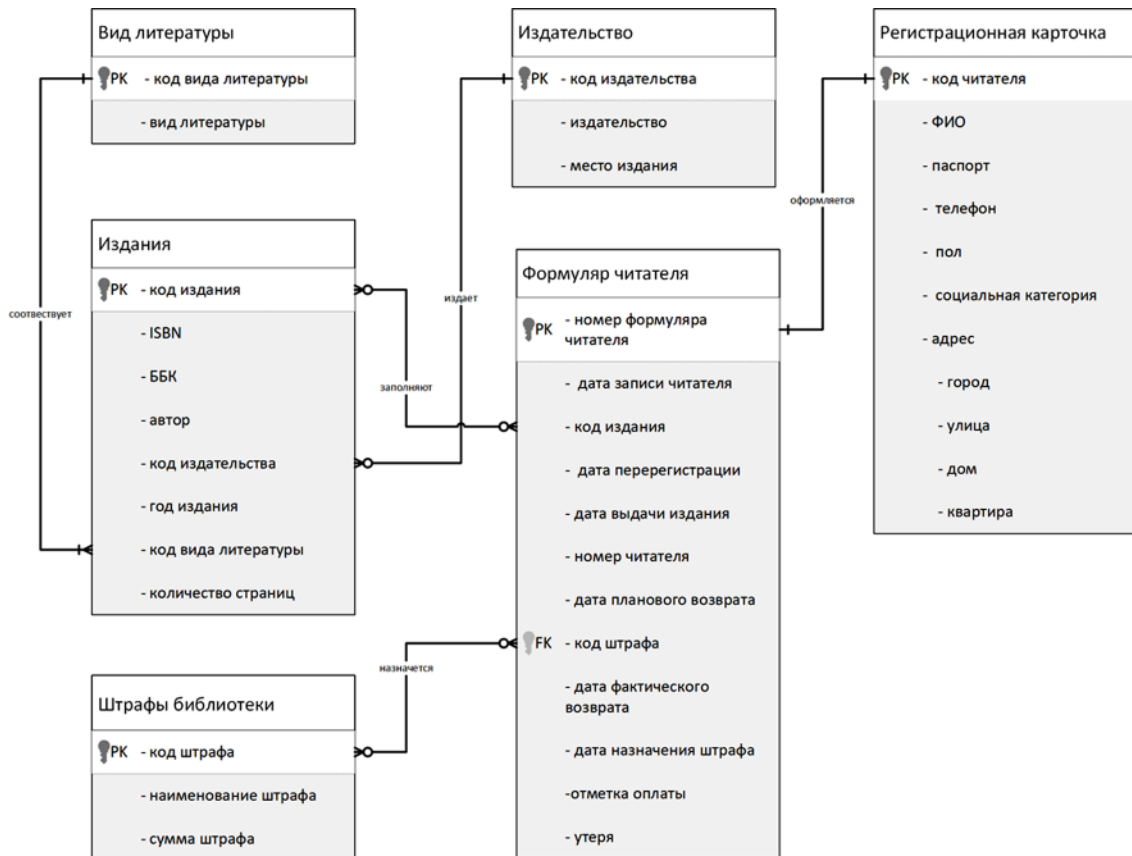


Рис. 1. ER-диаграмма модуля обслуживания читателей нотации Дж. Мартина

Диаграмма вариантов использования (прецедентов) применяется для визуализации функциональных и пользовательских требований к продукту, она может отразить сценарии работы разрабатываемым приложением относительно ролей пользователей. Вариант использования, другими словами, сценарий, прецедент или user case, описывает, то, как внешнее действующее лицо взаимодействует с системой, имея доступ к определенному функционалу [1, с. 266]. Имена прецедентов имеют глагольную форму и пишутся по шаблону «глагол (в неопределенной форме) + объект». Названия вариантов использования должны отражать то, какую ценность обеспечит каждый из них. Также user case диаграмма может конкретизировать необходимые для выполнения сценария ресурсы: то, что необходимо для выполнения процесса, что является основание бизнес-правил, с помощью чего осуществляется выполнения процесса, а также, что пользователь имеет на выходе [4, с. 25]. Таким образом, визуализирует

границы автоматизации бизнеса и дает четкое представление о том, что необходимо реализовать. Совместно с вариантами использования можно применять метод пользовательских историй, который дает более точную формулировку того, что необходимо пользователю [1, с. 234]. Например, вариант использования «Зарегистрировать читателя» в формате пользовательской истории будет звучать как «Как администратор я хочу иметь возможность создавать новую запись объекта «Формуляр читателя» для регистрации нового читателя». Такая формулировка обладает преимуществом перед более емким названием варианта использования, несмотря на то что и прецеденты, и пользовательские истории описывают цель пользователя, пользовательская история, в свою очередь, конкретизирует класс пользователя и дает обоснование необходимой функциональности системы. Пользовательские истории могут формулироваться в соответствии со следующим шаблоном (1):

Как <роль/персонаж пользователя>, я хочу <цель>, <с такой-то целью> (1)

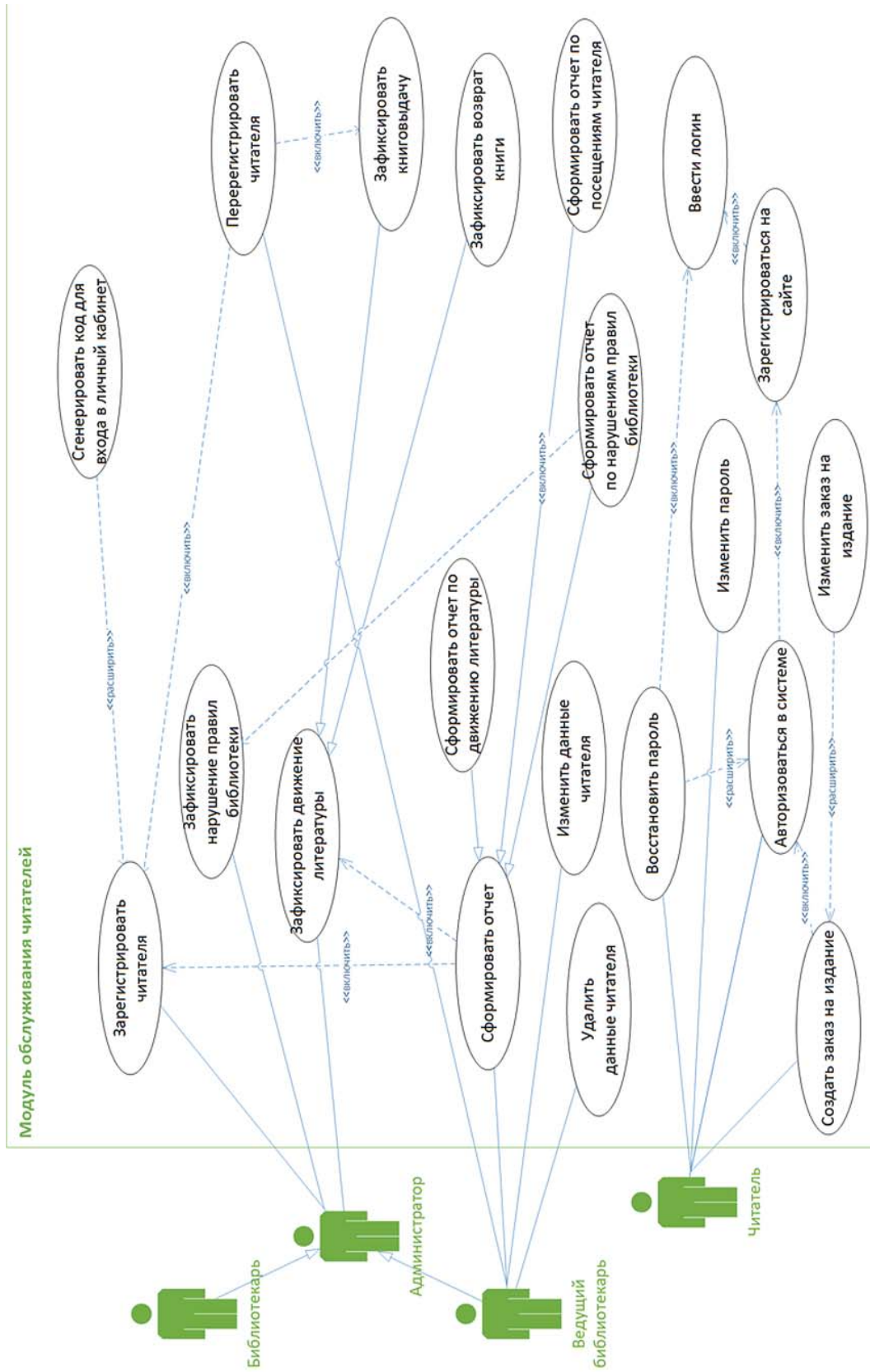


Рис. 2. Диаграмма прецедентов модуля обслуживания читателей системы управления библиотечной деятельностью

Фрагмент представлений пользовательских историй
и соответствующих вариантов использования

№	Вариант использования	Пользовательская история
1	Зарегистрировать читателя	Как пользователь с ролью «Администратор», я хочу создать нового читателя для внесения его данных в формуляр читателя
2	Сгенерировать код для входа в личный кабинет	Как пользователь с ролью «Администратор», я хочу сгенерировать код доступа читателя в личный кабинет на сайте библиотеки для формирования онлайн заказа на книги
3	Зафиксировать книговыдачу	Как пользователь с ролью «Администратор», я хочу зафиксировать выдачу книги читателю для ведения учета выданных экземпляров книги
4	Зафиксировать возврат книги	Как пользователь с ролью «Администратор», я хочу зафиксировать возврат книги читателю для ведения учета выданных экземпляров книги
5	Формировать отчеты	Как пользователь с ролью «Ведущий библиотекарь», я хочу формировать отчеты для анализа обслуживания читателей
6	Сформировать отчет по движению литературы	Как пользователь с ролью «Ведущий библиотекарь», я хочу формировать отчет по движению литературы для выявления более или менее популярных изданий
7	Изменить данные читателя	Как пользователь с ролью «Ведущий библиотекарь», я хочу изменить данные читателя для корректировки ФИО, контактной информации, категории
8	Зарегистрироваться на сайте	Как пользователь с ролью «Читатель», я хочу зарегистрироваться на сайте библиотеки, введя код авторизации, для получения доступа к личному кабинету и оформления заказа на книги онлайн
9	Изменить пароль	Как пользователь с ролью «Читатель», я хочу изменить пароль, для входа в личный кабинет на сайте библиотеки
10	Создать заказ на издание	Как пользователь с ролью «Читатель», я хочу создать заказ в личном кабинете на выдачу книг для оформления заказа онлайн
11	Изменить заказ на издание	Как пользователь с ролью «Читатель», я хочу изменить заказ на выдачу книг в личном кабинете пока заказ находится в статусе «в обработке»

Диаграммы нотации BPMN и диаграммы деятельности строятся для отражения хода процесса в разрезе ролей пользователей. С помощью них можно отразить ход выполнения конкретного сценария из диаграммы прецедентов, с учетом выполнения всех условий и ограничений процесса. Нотация BPMN позволяет визуализировать логику выполнения варианта использования [5, с. 39]. Таким образом, детализируется представление пользовательских требований, функциональных требований со стороны системы, визуализируется применение бизнес-правил, атрибутов качества, учета работы с внешними системами и интерфейсами, работа ограничений разработки.

Из рис. 2 видно, что в автоматизация предполагает наличие трех ролей пользователей: администратор, ведущий библиотекарь и читатель. Отдельно роль «Библиотекарь» не реализуется, так как роли «Библиотекарь» и «Ведущий библиотекарь» обладают схожим функционалом поэтому их обобщили до роли «Администратор», при этом роль «Ведущий библиотекарь» об-

ладает собственным функционалом. На диаграмме представлены сценарии работы всех ролей пользователей системы, с указанием связей-зависимостей.

Для вариантов использования рис. 2. представим соответствующие пользовательские истории (таблица), формулировка которых более детально представит необходимый функционал для разработки требований модуля обслуживания читателей.

Далее для более подробного рассмотрения выберем сценарий «Зарегистрировать читателя». Описание сценария «Зарегистрировать читателя»

2.1-1 Зарегистрировать читателя

Краткое описание: Пользователь с ролью Администратор (библиотекарь, либо ведущий библиотекарь) выбирает опцию «Создать формуляр читателя», вводит данные читателя, распечатывает документы для подписания и выдает код регистрации после успешной регистрации читателя. Код регистрации выдается для авторизации пользователя на сайте библиотеки для формирования заказов на литературу онлайн.

Действующие лица: Администратор (1ДЛ), Читатель (2ДЛ)

Триггер: Читатель попросил зарегистрировать его в библиотеке.

Предусловия: Администратор авторизован в библиотечной системе. Система находится в статусе «ожидание».

Основной поток:

1. Администратор получает запрос регистрации нового читателя.
2. Администратор предъявляет правила пользования библиотекой и запрашивает документ, удостоверяющий личность.
3. Читатель знакомится с правилами пользования библиотекой и предоставляет документ, удостоверяющий личность.
4. Если читатель принимает правила библиотеки, то управление переходит на следующий шаг.
5. Если читатель предъявляет документ, удостоверяющий личность и имеет регистрацию РФ, то управление переходит на следующий шаг.
6. Если читатель имеет регистрацию РФ, то управление переходит на следующий шаг.
7. Администратор выбирает в системе опцию «Создать формуляр читателя».
8. Система отображает форму для заполнения сведений о новом формуляре читателя.
9. Администратор вводит данные читателя.
10. Администратор сохраняет данные по новому формуляру читателя.
11. Система отображает уведомление об успешном создании формуляра читателя.
12. Администратор создает запрос на генерацию кода для входа в личный кабинет на сайте библиотеки.
13. Система генерирует и выдает код.
14. Система формирует и выводит на печать документы, дающие право пользоваться библиотечными услугами.
15. Администратор выдает документы читателю для подтверждения обязательств

по выполнению условий библиотечно-обслуживания.

16. Читатель подписывает документы.

17. Администратор выдает формуляр читателя и код регистрации, статус читателя «зарегистрирован».

18. Вариант использования завершает свою работу.

Поток исключение:

4а. Читатель отказывается принимать правила библиотеки:

1. Администратор уведомляет читателя о невозможности его регистрации.

2. Вариант использования завершает свою работу.

5а. Читатель отказывается представить документ, удостоверяющий личность:

1. Администратор уведомляет читателя о невозможности его регистрации.

2. Вариант использования завершает свою работу.

6а. Читатель не имеет регистрации в РФ:

1. Администратор уведомляет читателя о невозможности его регистрации.

2. Вариант использования завершает свою работу.

Постусловие: В случае успешного выполнения основного потока, в системе управления библиотечной деятельностью появляется новая запись в объекте «Формуляр читателя», читателю выдается формуляр читателя и код регистрации.

Результат: Если сценарий выполнен успешно, то читатель зарегистрирован.

На рис.3а и 3б представлен процесс регистрации нового читателя.

Формализации представления требования предполагает описание официально деловым стилем либо возможностей и ограничений пользователя, либо функций и ограничений системы [1, с. 87]. Описание с точки зрения возможностей и функций осуществляется по маскам ввода: описания требований ролей (2) и описания требований к системе (3).

<Роль> должен иметь возможность <возможность> (2)

<Система> должна <выполняемая функция> (3)

Описание с точки зрения ограничений осуществляется по маскам ввода: для ролей (4) и для системы (5).

<Роль> не должен <действие> <документ/регламент, регулирующий ограничение> (4)

<Система> должна <выполняемая функция> каждые <производительность>
<единица измерения> (5)

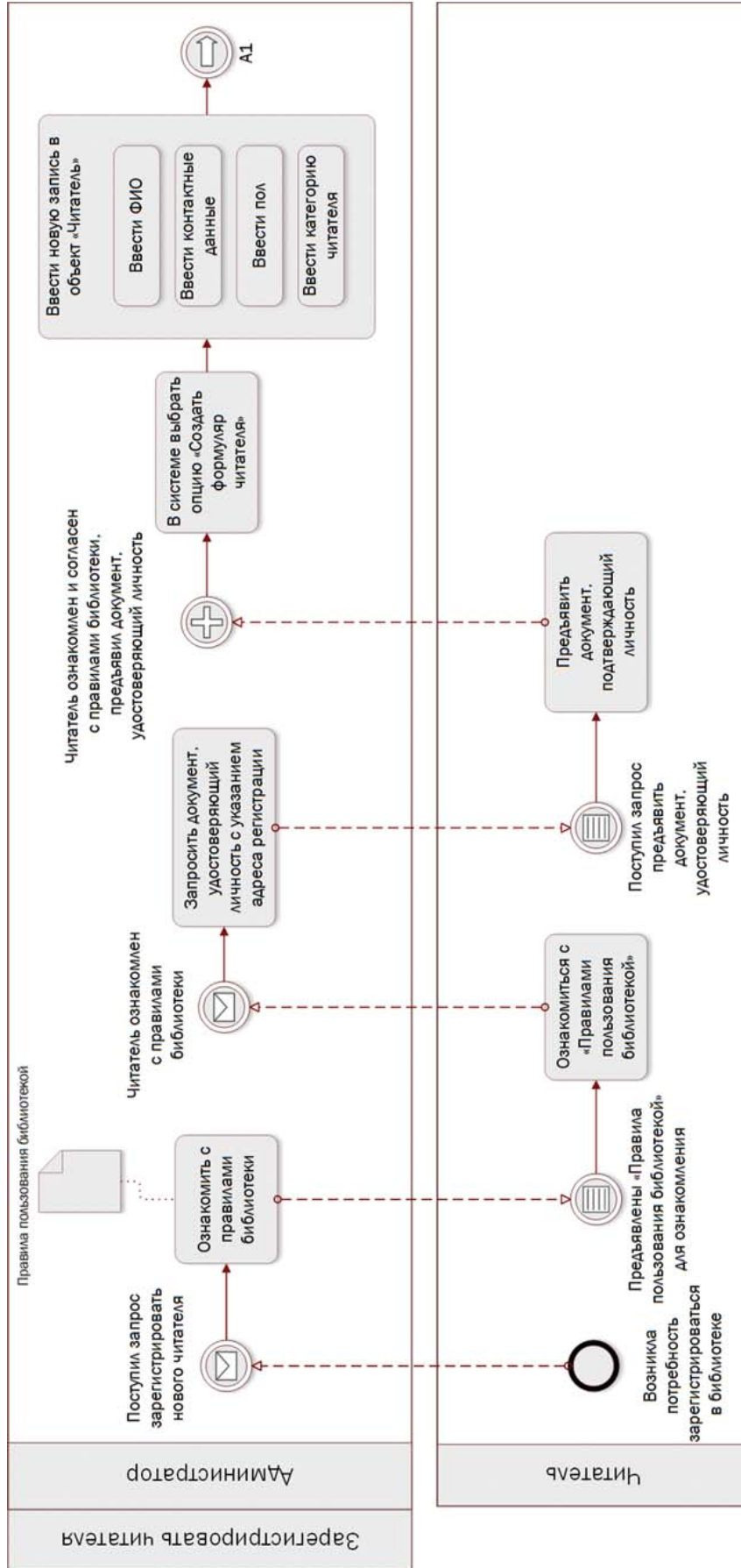


Рис. 3а. Диаграмма процесса регистрации читателя модуля обслуживания читателей системы управления библиотечной деятельностью в нотации BPMN (начало)

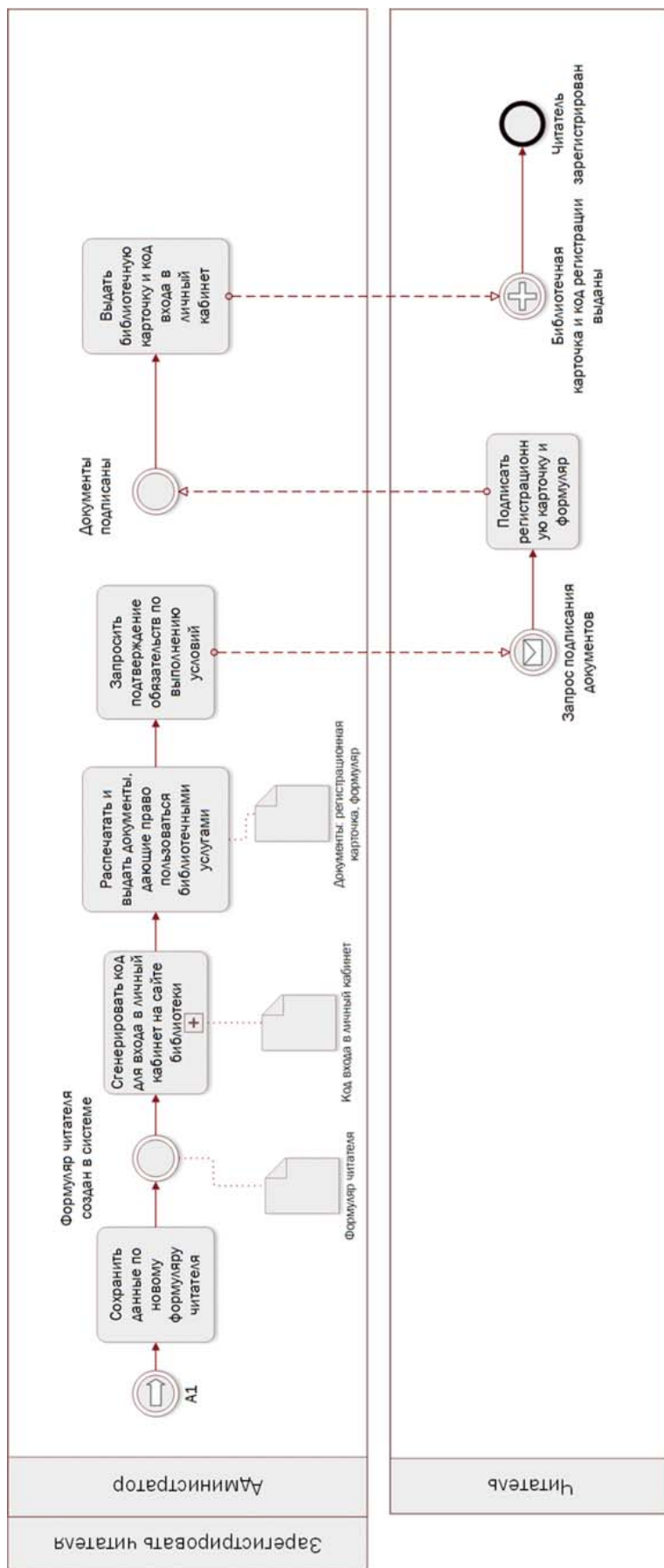


Рис. 36. Диаграмма процесса регистрации читателя модуля обслуживания читателей системы управления библиотечной деятельностью в нотации BPMN (окончание)

Также формализация требования предполагает присвоение уникального идентификационного номера для каждого требования. Уникальный номер может формироваться произвольно, быть представлен либо в виде уникального порядкового номера, либо задан маской ввода, с указанием вида требования, например, номер ФТ001, где «ФТ» обозначение вида требования «функциональное требование» и 001 – это порядковый номер функционального требования; номер h0017, где h – это высокий приоритет требования, O тип требования «ограничение», 017 – порядковый номер требования.

Проведенная визуализация задачи позволила выявить новые требования и уточнить имеющиеся. Таким образом, требования для модуля обслуживания читателей будут сформулированы следующим образом.

БТ001. Необходимо автоматизировать процесс регистрации нового читателя.

БТ002. Необходимо автоматизировать процесс формирования документов регистрации нового читателя и вывода их на печать.

ПТ001. Администратор должен иметь возможность создавать новую запись объекта «Формуляр читателя».

ПТ002. Администратор должен иметь возможность просматривать уже созданные экземпляры объекта «Формуляр читателя».

ПТ003. Администратор должен иметь возможность формировать документы, дающие право пользоваться библиотечными услугами.

ПТ004. Администратор должен иметь возможность печатать документы, дающие право пользоваться библиотечными услугами.

ПТ005. Администратор должен иметь возможность формировать запрос на генерацию кода для входа в личный кабинет на сайте библиотеки.

ФТ001. Система должна предоставить возможность создать новую запись объекта «Читатель».

ФТ002. Система должна генерировать уникальный номер экземпляра объекта «Формуляр читателя».

ФТ003. Система должна предоставить возможность ввода номера телефона читателя по маске ввода – (999)999-99-99.

ФТ004. Система должна предоставить возможность ввода пола читателя из раскрывающегося списка, определяющего пол.

ФТ005. Система должна предоставить возможность ввода категории читателя из раскрывающегося списка, определяющего категорию читателя.

ФТ006. Система должна автоматически фиксировать дату и время создания новой записи объекта «Формуляр читателя».

ФТ007. Система должна генерировать код для входа в личный кабинет на сайте библиотеки.

ФТ008. Система должна формировать документы «Регистрационная карточка» и «Читательский формуляр» и выводить их на печать.

ФТ009. Система должна выводить на печать документы «Регистрационная карточка» и «Читательский формуляр».

БП001. Регистрация нового читателя в системе должна соответствовать правилам пользования библиотекой.

БП002. Администратор не должен регистрировать нового читателя при отказе следовать правилам пользования библиотекой.

АК001. Время загрузки формы создания нового объекта «Формуляр читателя» не должно превышать 2 секунд.

АК002. Система должна предоставлять возможность параллельного заполнения нормативно-справочной информации (читатель, категория читателя) совместно с созданием новой записи объекта «Формуляр читателя».

О001. Система должна корректно работать на мобильных устройствах.

Практика разработки требований к продукту свидетельствует о том, что ни один формат представления требований не демонстрирует полную картину моделируемой предметной области.

Поэтому необходимо комплексное применение различных способов представления информации, а именно, комбинация текстовых, табличных и графических методов представления. Именно визуализация представления автоматизируемой предметной области детализирует описание объектов, их связей, типов взаимодействия, логической последовательности процессов, дает четкие границы реализации, лучшее понимание проблемной ситуации, устраняет все неоднозначности и неточности и, соответственно, формирует более четкую картину разработки и формулировку требований.

Библиографический список

1. Вигерс К., Битти Д. Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. М.: Русская редакция; СПб.: БХВ-Петербург, 2014. 736 с.
2. Гумерова Г.Р., Абдуллина Э.И., Бурханова Ю.Н. Применение диаграмм классов при проектировании информационных систем субъектов экономической деятельности // Наука Красноярья. 2023. Т. 12, № 3-3. С. 67-81.
3. Гумерова Г.Р., Мансурова Т.Г. Проектирование и разработка компонентов информационной системы образовательного учреждения на примере проекта автоматизации работы специалиста по научно-исследовательской работе. Курск: Университетская книга, 2022. 140 с. DOI 10.47581/2022/Gumerova.01.
4. Коберн А. Современные методы описания функциональных требований к системам / Пер. с англ. Лори, 2002. 288 с.
5. Репин В. Моделирование бизнес-процессов в нотации BPMN. Пособие для начинающих. Часть I. Ridero, 2019. 68 с.
6. Сенека Л.А. Избранные труды / Пер. с латинского П. Краснова, С. Янушевского, С. Соловьева. М.: Эксмо, 2020. 369 с.