

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ КемГУ

Дата и время: 2025-04-23 00:00:00

471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

Кузбасский гуманитарно-педагогический институт
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»

Факультет информатики, математики и экономики

Кафедра информатики и вычислительной техники им. В. К. Буторина

И. А. Жибинова

Разработка технической документации информационных систем

*Методические рекомендации к выполнению контрольной работы
для обучающихся по направлению подготовки*

09.03.03 - Прикладная информатика

(направленность (профиль) Прикладная информатика в экономике

Новокузнецк

2024

УДК 378.147.88(072)
ББК 74.484(2Рос-4Кем)я73
Ж66

Жибинова И. А.

Ж66 «Разработка технической документации информационных систем: метод. реком. к выполнению контрольной работы» для обучающихся по направлению подготовки 09.03.03 Прикладная информатика (направленность (профиль) «Прикладная информатика в экономике») / И. А. Жибинова ; Кузбасский гуманитарно-педагогический ин-т Кемеров. гос. ун-та. – Новокузнецк: КГПИ ФГБОУ ВО «КемГУ», 2024. – 52 с.

Приводятся: общие сведения по теме работы; задание; требования к представлению результатов; порядок и критерии оценивания контрольной работы; пример составления спецификации требований к программному обеспечению; список рекомендованных источников.

Предназначены для студентов, обучающихся по направлению подготовки 09.03.03 Прикладная информатика (направленность (профиль) «Прикладная информатика в экономике»). Могут быть использованы обучающимися по направлению подготовки 09.03.01 Информатика и вычислительная техника (направленность (профиль) подготовки «Автоматизированные системы обработки информации и управления»).

Рекомендовано
на заседании кафедры информатики и
вычислительной техники
им. В. К. Буторина
21 ноября 2024 г.

Утверждено
методической комиссией фа-
культета информатики, матема-
тики и экономики
12 декабря 2024 г..

Заведующий кафедрой



А. В. Маркидонов

Председатель методкомиссии



И. А. Жибинова

©Жибинова И. А., 2024

© Кузбасский гуманитарно-педагогический институт федерального государственного бюджетного образовательного учреждения высшего образования «Кемеровский государственный университет», 2024

Текст представлен в авторской редакции

Содержание

Введение	4
1 Задание к контрольной работе	6
2 Требования к представлению результатов контрольной работы	7
3 Порядок и критерии оценивания контрольной работы	8
4 Спецификация программных требований. Общие сведения	9
4.1 Сущность SRS	10
4.2 Характеристики правильно составленной SRS	11
4.3 Шаблоны SRS	16
4.3.1 SRS по IEEE 830-1998	16
4.3.2 SRS по ISO/IEC/ IEEE 29148-2018	18
4.3.3 Адаптированный шаблон спецификации требований Карла Вигерса	20
5 Пример составления спецификации требований к программному обеспечению	22
5.1 Исходные данные	23
5.2 Пример спецификации требований	24
Список рекомендуемых источников	52

Введение

Разработка технической документации является неотъемлемой частью профессиональной деятельности специалиста в IT-сфере.

Федеральными государственными стандартами высшего образования (ФГОС ВО) по направлениям подготовки укрупнённой группы специальностей «Информатика и вычислительная техника» в числе общепрофессиональных компетенций выпускника предусмотрено формирование компетенции ОПК-4 – *способность участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью*. Аналогичное требование предъявляют к специалистам и профессиональные стандарты.

Одной из самых частых задач, которую решает бизнес-аналитик в данной профессиональной сфере является разработка требований к программному продукту и их оформление в виде технического задания или спецификации требований к программному обеспечению (ПО). На закрепление материала, полученного на лекционных и практических занятиях, в процессе самостоятельной работы, а также на формирование у обучающихся практических навыков разработки спецификации ПО (Далее – спецификация) направлена контрольная работа, предусмотренная учебным планом.

Методические рекомендации к выполнению контрольной работы предназначены для обучающихся в заочной форме по направлению подготовки 09.03.03 Прикладная информатика (Прикладная информатика в экономике) при освоении ими дисциплины «Разработка технической документации информационных систем».

Кроме того, материалы методических рекомендаций будут полезны студентам очной формы обучения при выполнении практических заданий по дисциплине, а так же обучающимся по направлению подготовки 09.03.01 Информатика и вычислительная техника (Автоматизированные системы обработки информации и управления).

Методические рекомендации содержат:

1. Задание к контрольной работе.
2. Требования к представлению результатов контрольной работы.
3. Порядок и критерии оценивания контрольной работы.
4. Общие сведения. В этом разделе представлена предварительная информация, которую необходимо рассмотреть при составлении требований к программному обеспечению (ПО).
5. Пример составления спецификации требований к программному обеспечению.
6. Список рекомендованных источников.

1 Задание к контрольной работе

В контрольной работе по дисциплине «Разработка технической документации информационных систем» студент должен на основе изучения рекомендаций международных стандартов (см. п. 4.3) составить документ «Спецификация программных требований».

Формулировка задания является общей «Составить спецификацию требований к «_____» по
название ПО (программного изделия, программы или набора программ)
рекомендациям _____
название стандарта

Варианты различаются объектами разработки, т.е. ПО и стандартом, примененным для разработки спецификации.

ПО выбирается на усмотрение студента. Это может быть ПО, которое самостоятельно разрабатывается (дорабатывается) студентом при освоении дисциплины «Проектный практикум», по теме выпускной квалификационной работы и т.п.

Стандарт для разработки спецификации так же выбирается на усмотрение студента.

Например, задание контрольной работы может быть сформулировано следующим образом:

1. Составить спецификацию требований к программному обеспечению прототипа усовершенствованной транзитной системы ЦАРЭС по стандарту ISO/IEC/ IEEE 29148-2018 Systems and software engineering -- Life cycle processes -- Requirements engineering.

2. Составить спецификацию требований к модулю «АССИСТЕНТ» Единой биометрической системы по IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications –Description.

Обратите внимание! Задание должно быть согласовано с преподавателем.

2 Требования к представлению результатов контрольной работы

1. Контрольная работа оформляется в виде текстового документа.
2. Листы в контрольной работе приводятся в следующей последовательности:

1) Титульный лист

2) Содержание

3) Задание

4) Основная часть

5) Список использованных источников.

3. Наименования структурных элементов **Содержание, Задание, Список использованных источников**, служат заголовками и не нумеруются.

4. Содержание должно быть автособираемым.

5. Наименование «**Основная часть**» в заголовок не выносится.

Основную часть контрольной работы следует назвать в соответствии с заданием «Спецификация требований к
« _____ ».

название ПО (программного изделия, программы или набора программ)

Заголовки структурных элементов основной части (разделов, подразделов, пунктов подпунктов) формулируются в соответствии с выбранным шаблоном и им присваивается сквозная нумерация по установленным правилам¹.

Например, если спецификация составлена по стандарту ISO/IEC/ IEEE 29148-2018, то заголовки разделов, подразделов, пунктов формулируются в соответствии с рисунком Рисунок 4.2).

Обратите внимание! 1) Спецификация должна содержать все разделы (части), в соответствии с выбранным шаблоном. В случае их отсутствия,

¹ Правила оформления учебных работ студентов [Текст] : учебно-методическое пособие / И. А. Жибинова [и др.] ; Новокузнец. ин-т (фил.) Кемеров. гос. ун-та ; под ред. И. А. Жибиновой. – Новокузнецк: НФИ КемГУ, 2018. – 124 с.

должен быть приведен заголовок раздела и сделана соответствующая запись.

2) При составлении спецификации следует руководствоваться правилами, приведенными в п.п. 4.3.3.

6. Контрольная работа оформляется в соответствии с требованиями и правилами, установленными в вузе (см. сноску 1).

7. Контрольная работа размещается студентом на образовательном портале в соответствии с графиком учебной работы.

3 Порядок и критерии оценивания контрольной работы

Выполненная контрольная работа подлежит защите. Защита осуществляется в назначенное преподавателем время в форме собеседования и включает устный доклад студента о выполненной работе и ее результатах с демонстрацией на компьютере, а также ответы на вопросы, поставленные преподавателем.

Работа оценивается по балльно-рейтинговой системе.

Максимальный балл (41 балл) ставится в том случае, если работа выполнена в установленные сроки; структура и содержание разработанной спецификации соответствуют рекомендациям выбранного стандарта; формулировки отвечают критериям качественных требований; при собеседовании студент уверенно и доказательно отвечает на поставленные вопросы, показывает глубокие знания вопросов темы, свободно оперирует данными; количество правильных ответов на защите составляет от 80 до 100 процентов; текст оформлен в полном соответствии требованиям к содержанию и оформлению.

Пороговый балл (20 баллов) ставится в том случае, если без уважительной причины нарушен установленный срок представления работы; не достаточно четко просматривается соответствие структуры и содержания разработанной спецификации рекомендациям выбранного стандарта; формулировки не в полной мере отвечают критериям качественных требований;

при защите студент проявляет неуверенность, не дает полного, аргументированного ответа на заданные вопросы; количество правильных ответов на защите составляет от 51 до 65 процентов; текст не в полной мере отвечает установленным требованиям к оформлению.

Работа, набравшая больше 20-ти баллов оценивается на «зачтено», о чем в зачетной ведомости делается соответствующая запись.

4 Спецификация программных требований. Общие сведения

Чтобы требования, выявленные и описанные, приняли силу соглашения между Заказчиком и Разработчиком, их необходимо оформить в виде документа.

В российской практике для этого обычно используется документ «*Техническое задание*» (ТЗ), в западной – «*Software Requirements Specification*» (SRS) – спецификация программных требований

Структура и содержание ТЗ определяется специальным шаблоном – межгосударственными стандартами в зависимости от объекта разработки:

- ГОСТ 34. 602-2020 Информационные технологии. КОМПЛЕКС СТАНДАРТОВ НА АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. Техническое задание на создание автоматизированной системы
- ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению

Однако, сегодня большинство разработчиков и аналитиков считают эти отечественные ГОСТы морально устаревшими и все чаще вместо них для оформления требований к ПО используют шаблоны зарубежных стандартов, которые регламентируют составление *спецификаций требований* к ПО (Далее – SRS).

4.1 Сущность SRS

Документ спецификации требований к программному обеспечению (SRS) представляет собой подробное, структурированное описание функциональных и нефункциональных требований к программной системе. Предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком (или подрядчиками) о том, как должен функционировать программный продукт

В спецификации требований к ПО указываются функции и возможности, которыми должно обладать ПО, а также необходимые ограничения. Она должна содержать достаточно подробное описание поведения системы при различных условиях, а также необходимые качества системы, такие как производительность, безопасность и удобство использования. Спецификация требований служит основой для дальнейшего планирования, дизайна и кодирования, а также базой для тестирования пользовательской документации

Основными вопросами, которые должны рассматривать составитель SRS, являются следующие:

а) **Функциональные возможности.** Каковы предполагаемые функции программного обеспечения?

б) **Внешние интерфейсы.** Как программное обеспечение взаимодействует с пользователями, аппаратными средствами системы, другими аппаратными средствами и другим программным обеспечением?

в) **Рабочие характеристики.** Каково быстродействие, доступность, время отклика, время восстановления различных функций программного обеспечения и т.д.?

г) **Атрибуты.** Каковы мобильность, правильность, удобство сопровождения, защищенность программного обеспечения и другие критерии?

д) **Проектные ограничения, налагаемые на реализацию изделия.** Предъявляются ли требования к соответствию каким-либо действующим стандартам, к языку реализации, к политикам целостности баз данных, к за-

действующим ресурсам, к операционной среде (-ам) и т.д.?

Составителю SRS следует избегать размещения в SRS архитектурных решений или требований к организации и выполнению проекта.

Так как SRS играет определенную роль в процессе разработки программного обеспечения, составителю SRS следует проявлять осторожность, чтобы не выйти за пределы этой роли. Это означает, что SRS:

а) Должна правильно определять все требования к ПО. Причиной существования какого-либо требования к ПО может являться характер решаемой задачи или особая характеристика проекта.

б) Не должна описывать детали проектирования или реализации. Они должны быть описаны на этапе проектирования.

в) Не должна налагать дополнительные ограничения на ПО. Эти ограничения надлежащим образом определяются в других документах, таких как план обеспечения качества программных средств.

Следовательно, правильно составленная SRS ограничивает диапазон допустимых архитектурных решений, но не определяет какой-либо конкретной архитектуры.

4.2 Характеристики правильно составленной SRS

SRS должна быть:

- а) Корректной;
- б) Однозначной;
- в) Полной;
- г) Непротиворечивой;
- д) Упорядоченной по ее значимости и/или устойчивости;
- е) Проверяемой;
- ж) Модифицируемой;
- з) Отслеживаемой.

Корректность SRS является корректной, если, и только, если каждое

требование, изложенное в ней, является требованием, которому должно удовлетворять ПО. Не существует какого-либо средства или процедуры, которые гарантирует корректность. SRS должна сравниваться с любой применимой спецификацией более высокого уровня, такой как спецификация требований к системе, с другой документацией проекта и с другими применимыми стандартами, для обеспечения её согласованности с ними. В качестве альтернативы заказчик или пользователь может определить, правильно ли SRS отражает фактические потребности.

Однозначность. SRS является однозначной, если и только, если каждое изложенное в ней требование имеет только одно толкование. Как минимум, для этого требуется, чтобы каждое свойство конечного изделия было описано посредством одного уникального выражения.

В тех случаях, когда термин, используемый в специфическом контексте, может иметь множественные значения, этот термин должен быть включен в глоссарий, в котором его значение описывается более конкретно.

SRS является важной частью процесса составления требований в жизненном цикле ПО и используется при проектировании, реализации, мониторинге проекта, верификации и проверке правильности, а также при обучении. SRS должна быть однозначной как для тех, кто составляет её, так и для тех, кто её использует. Однако, эти группы часто не обладают общим контекстом, благодаря различиям в своём опыте, подготовке и организационной роли и, следовательно, не склонны описывать требования к ПО схожим образом. Способы представления требований, которые улучшают спецификацию требований для разработчика, могут оказаться неэффективными в том, что они ухудшают понимание пользователем, и наоборот.

Завершенность. SRS является полной, если и только, если она включает следующие элементы:

а) Все существенные требования, независимо от того, относятся ли они к функциональным возможностям, рабочим характеристикам, проектным ограничениям, атрибутам или внешним интерфейсам. В частности, должны

быть подтверждены и обработаны любые внешние требования, налагаемые спецификацией системы (частью которой будет являться программа).

б) Определение откликов ПО на все классы входных данных, которые могут быть реализованы, во всех возможных ситуациях. Следует заметить, что важно определить отклики, как на допустимые, так и недопустимые входные значения.

в) Полные обозначения и ссылки на все рисунки, таблицы и схемы в SRS и определение всех терминов и единиц измерения.

Непротиворечивость. Непротиворечивость обозначает внутреннюю непротиворечивость. Если SRS не согласована с каким-то документом более высокого уровня, таким как, например, спецификация системных требований (System Requirements Specification, SyRS), то она является некорректной.

Упорядочивание по значимости и/или устойчивости. SRS является упорядоченной по значимости и/или устойчивости, если каждое требование в ней имеет идентификатор, указывающий или значимость или устойчивость этого конкретного требования.

Как правило, все требования, которые касаются программного изделия, не являются важными в равной степени. Некоторые требования могут быть неотъемлемыми, особенно для приложений, от которых зависит жизнь людей, в то время как другие могут быть лишь желательными.

Каждое требование в SRS должно быть идентифицировано, чтобы сделать эти различия четкими и явными. Идентификация требований следующим образом помогает:

а) заказчикам более тщательно рассмотреть каждое требование, что часто позволяет разъяснить любые скрытые допущения, которые могут быть заключены в них.

б) разработчикам принять правильные проектные решения и приложить соответствующие усилия к различным составляющим программного изделия.

Один метод идентификации требований использует величину *устойчи-*

вости. Устойчивость может быть выражена в количестве ожидаемых изменений для каждого требования, основанных на опыте или знании предстоящих событий, которые влияют на организацию, функции, и пользователей, поддерживаемых системой программного обеспечения.

Другой способ упорядочения требований состоит в том, чтобы различать классы требований как *необходимые, условные и необязательные*.

а) *Необходимые*. Подразумевают, что программное обеспечение не будет пригодным, если эти требования не будут обеспечены согласованным образом.

б) *Условные*. Подразумевают, что эти требования расширяют программное изделие, но не сделают его непригодным при их отсутствии.

в) *Необязательные*. Подразумевают класс функций, реализация которых может быть целесообразной, а может и нет. Это даёт разработчику возможность предложить что-либо, что выходит за пределы SRS.

Проверяемость. SRS является проверяемой, если и только, если каждое требование, изложенное в ней, может быть проверено. Требование являются проверяемым, если и только, если существует некий конечный экономически-эффективный (рентабельный) процесс, используя который пользователь или машина могут убедиться, что программное изделие удовлетворяет этому требованию. В целом, любое неоднозначное требование не проверяемо.

Непроверяемые требования включают формулировки типа «работает хорошо», «хороший пользовательский интерфейс» и «обычно должно происходить». Эти требования не могут быть проверены, так как невозможно определить термины «хороший», «хорошо» или «обычно». Утверждение о том, что «программа никогда не должна заикливаться» является непроверяемым, так как тестирование этого свойства теоретически невозможно.

Примером проверяемого утверждения является следующее:

Программа должна выдавать результат в течение 20 секунд после заданного события в 60% случаев; и должна выдавать результат в течение

30 секунд после заданного события в 100% случаев.

Это утверждение может быть проверено, так как в нём используются конкретные термины и измеримые величины.

Если нельзя изобрести метод, чтобы определить, отвечает ли ПО определённому требованию, то это требование следует исключить или пересмотреть.

Модифицируемость. SRS является модифицируемой, если и только, если ее структура и стиль таковы, что любые изменения требований могут быть выполнены легко, полностью и непротиворечивым образом с сохранением структуры и стиля. Как правило, модифицируемость требует, чтобы SRS:

- а) Имела связанную и легкую в использовании структуру с оглавлением, алфавитным указателем и явно выраженными перекрёстными ссылками;
- б) Не была избыточной (то есть, одно и то же требование не должно появляться в SRS более чем в одном месте);
- в) Выражала каждое требование отдельно, не смешивая его с другими требованиями.

Избыточность сама по себе не является ошибкой, но она легко может привести к появлению ошибок.

Иногда избыточность может помочь сделать SRS более читаемой, но могут возникнуть проблемы при обновлении избыточного документа. Например, требование может быть изменено только в одном из тех мест, где оно появляется. Тогда SRS становится противоречивой. Каждый раз, когда избыточность необходима, SRS должна включать явные перекрёстные ссылки, чтобы сделать её модифицируемой.

Отслеживаемость. SRS является отслеживаемой, если четко прослеживается источник каждого из её требований и если она позволяет ссылаться на каждое из требований при дальнейшей разработке или модернизации документации. Рекомендуются следующие два типа отслеживаемости:

- а) Обратная отслеживаемость (то есть, к предыдущим стадиям разра-

ботки). Зависит от того, ссылается ли в явном виде каждое из требований на его источник в более ранних документах.

б) Прямая отслеживаемость (то есть, ко всем документам, порождаемым SRS). Зависит от того, имеет ли каждое требование из SRS однозначно определённое имя или номер ссылки.

Прямая отслеживаемость SRS особенно важна, когда программное изделие вступает в стадию эксплуатации и сопровождения. По мере изменения кода и архитектурных документов необходимо иметь возможность определить полный набор требований, на которые могут повлиять эти изменения.

4.3 Шаблоны SRS

Шаблон спецификации требований к ПО представляет собой структурированный, последовательный способ организации функциональных и нефункциональных требований.

Доступны различные шаблоны спецификации, например:

- IEEE 830-1998
- ISO IEEE 29148-2011
- RUP
- SWEBOOK, BABOK и пр.

В тексте указанных выше стандартов содержатся подробные сведения о том, как составлять SRS, а также шаблоны.

Рекомендуется принять один или несколько стандартных шаблонов спецификации требований к ПО для использования в проектах.

4.3.1 SRS по IEEE 830-1998

Достаточно хорошее определение стандарта **IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications – Description** (Стандарт IEEE 830-1998. Рекомендуемая методика для спе-

цификаций требований к программному обеспечению – описание) дано в самом его описании: Описывается содержание и качественные характеристики правильно составленной SRS и приводятся несколько шаблонов. Данная рекомендуемая методика имеет своей целью установление требований к разрабатываемому ПО, но также может применяться, чтобы помочь в выборе собственных и коммерческих программных изделий.

Рекомендуемая структура SRS показана на рисунке 4.1 в виде макета (шаблона), который может служить примером для составления SRS.

Несмотря на то, что SRS не должна точно повторять этот эскиз или использовать наименования, приведенные здесь для ее частей, качественно составленная SRS должна включать всю информацию, описанную в стандарте.

Содержание	
1. Введение	
1.1 Назначение	
1.2 Область действия	
1.3 Определения, акронимы и сокращения	
1.4 Публикации	
1.5 Краткий обзор	
2. Полное описание	
2.1 Контекст изделия	
2.1.1 Системные интерфейсы;	
2.1.2 Интерфейсы пользователя;	
2.1.3 Аппаратные интерфейсы;	
2.1.4 Интерфейсы программного обеспечения;	
2.1.5 Интерфейсы связи;	
2.1.6 Память;	
2.1.7 Операции;	
2.1.8 Требования по адаптации места использования	
2.2 Функции изделия	
2.3 Характеристики пользователя	
2.4 Ограничения	
2.5 Допущения и зависимости	
2.6 Разделение требований.	
3. Специфические требования	
3.1 Внешние интерфейсы	
3.2 Функции (функциональные требования)	
3.3 Требования к рабочим характеристикам (производительности)	
3.4 Требования к логическому устройству базы данных	
3.5 Архитектурные ограничения	
3.6 Атрибуты (качества) программной системы	
3.6.1 Надёжность	
3.6.2 Доступность	
3.6.3 Защита	
3.6.4 Удобство сопровождения	
3.6.5 Мобильность	
Приложения	
Алфавитный указатель	

Рисунок 4.1– Эскиз макета SRS по IEEE 830-1998

4.3.2 SRS по ISO/IEC/ IEEE 29148-2018

Стандарт ISO/IEC/ IEEE 29148-2018² **Systems and software engineering -- Life cycle processes -- Requirements engineering (Программная и системная инженерия. Процессы жизненного цикла. Разработка требований)** обеспечивает единую трактовку процессов и продуктов, используе-

² Взамен ISO/IEC/ IEEE 29148-2011

мых при разработке требований на протяжении всего жизненного цикла систем и программного обеспечения.

Данный стандарт содержит два шаблона спецификации требований:

- *System requirements specification* (SyRS)
- *Software requirements specification* (SRS)

System Requirements Specification (SyRS) определяет требования для выбранной системы и удобства взаимодействия предполагаемой системы и человека. Она определяет высокоуровневые требования к системе с точки зрения предметной области, а также информацию об общей цели системы, ее целевой среде и ограничениях, допущениях и нефункциональных требованиях. Она может включать в себя концептуальные модели, спроектированные для иллюстрации содержания системы, сценариев использования, основных сущностей предметной области, данных, информации и рабочих процессов. Из определения следует, что это аналог ТЗ, описанного в ГОСТ 34.

Software requirements specification (SRS) – это спецификация требований для определенного программного изделия, программы или набора программ (продукт), которые выполняют определенные функции в конкретном окружении. Из определения следует, что это аналог ТЗ, описанного в ГОСТ 19, а по структуре очень напоминает SRS из стандарта IEEE 830.

SRS может содержать следующие разделы в соответствии с рисунком 4.2.

1. Введение
 - 1.1. Назначение
 - 1.2. Охват
 - 1.3. Окружение
 - 1.4. Обзор функций продукта
 - 1.5. Характеристики пользователей
 - 1.6. Ограничения
 - 1.7. Допущения и зависимости
 - 1.8. Термины и определения
 - 1.9. Сокращения и аббревиатуры
2. Требования
 - 2.1. Внешние интерфейсы
 - 2.2. Функциональные требования
 - 2.3. Требования к удобству использования
 - 2.4. Требования к производительности
 - 2.5. Требования к логической структуре данных
 - 2.6. Ограничения при проектировании
 - 2.7. Соответствие стандартам
 - 2.8. Характеристики программного продукта
 - 2.8.1. Надежность
 - 2.8.2. Доступность
 - 2.8.3. Безопасность
 - 2.8.4. Ремонтопригодность
 - 2.8.5. Переносимость (портируемость)
 - 2.8.6.
3. Проверка результата
4. Вспомогательная информация
5. Ссылки

Рисунок 4.2 – Содержание спецификации на программный продукт по ISO/IEC/IEEE 29148:2018.

4.3.3 Адаптированный шаблон спецификации требований Карла Вигерса

На рисунке 4.3 показан адаптированный шаблон спецификации требований Карла Вигерса³, который подходит для многих проектов, не является официальным комплексом требований, но *de facto* используется многими

³ Вигерс Карл, Битти Джой? Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М. : Издательство «Русская редакция» ; СПб. : БХВ-Петербург, 2014. — 736 стр. : ил.

аналитиками как стандарт разработки спецификации.

-
1. Введение
 - 1.1 Назначение
 - 1.2 Соглашения, принятые в документах
 - 1.3 Границы проекта
 - 1.4 Ссылки
 2. Общее описание
 - 2.1 Общий взгляд на продукт
 - 2.2 Классы и характеристики пользователей
 - 2.3 Операционная среда
 - 2.4 Ограничения дизайна и реализации
 - 2.5 Предположения и зависимости
 3. Функции системы
 - 3.x Функция системы X
 - 3.x.1 Описание
 - 3.x.2 Функциональные требования
 4. Требования к данным
 - 4.1 Логическая модель данных
 - 4.2 Словарь данных
 - 4.3 Отчеты
 - 4.4 Получение, целостность, хранение и утилизация данных
 5. Требования к внешним интерфейсам
 - 5.1 Пользовательские интерфейсы
 - 5.2 Интерфейсы ПО
 - 5.3 Интерфейсы оборудования
 - 5.4 Коммуникационные интерфейсы
 6. Атрибуты качества
 - 6.1 Удобство использования
 - 6.2 Производительность
 - 6.3 Безопасность
 - 6.4 Техника безопасности
 - 6.x [Другие]
 7. Требования по интернационализации и локализации
 8. Остальные требования
- Приложение А. Словарь терминов
- Приложение Б. Модели анализа

Рисунок 4.3 – Шаблон для спецификации требований к ПО

Там же приведены советы, которым рекомендуется следовать, чтобы сделать требования ясными и понятными:

- для структурирования всей необходимой информации используйте соответствующий шаблон;
- разделы, подразделы и отдельные требования должны быть имено-

ваться единообразно;

- используйте средства визуального выделения (такие, как полужирное начертание, подчеркивание, курсив и различные шрифты) последовательно и в разумных пределах;
- создайте оглавление, чтобы облегчить пользователям поиск необходимой информации;
- пронумеруйте все рисунки и таблицы, озаглавьте их и, ссылаясь на них, используйте присвоенные номера;
- если при хранении требований в документе вы ссылаетесь в документе на другие его части, используйте возможности работы с перекрестными ссылками в вашем редакторе, а не сложную кодировку страниц;
- если вы используете документы, применяйте гиперссылки, чтобы читатель смог быстро перейти к соответствующим разделам спецификации или другим файлам;
- при хранении требований в специализированном средстве используйте ссылки, чтобы облегчить читателю навигацию по нужной информации;
- включайте графическое представление информации, где это возможно для облегчения понимания;
- обеспечьте последовательность документа и единообразие терминологии и структуры.

В разделе 4 показан пример спецификации требований к ПО, производной от этого шаблона.

5 Пример составления спецификации требований к программному обеспечению

Пример заимствован из книги Карла Вигерса «Разработка требований к программному обеспечению» (см. сноску 3).

5.1 Исходные данные

Из документа «О концепции и границах проекта»

Большинство сотрудников Process Impact в настоящее время тратят в среднем 65 минут в день, чтобы выбрать, оплатить и съесть обед в кафетерии. Около 20 минут уходит на то, чтобы дойти до кафетерия и вернуться на рабочее место, выбрать еду и оплатить ее наличными или с помощью кредитной карты. Таким образом, сотрудники проводят около 90 минут вне рабочих мест.

Некоторые звонят в кафетерий заранее, чтобы блюда были готовы к их приходу. Они не всегда получают то, что заказали, потому что некоторые блюда заканчиваются до их прихода. Кафетерий впустую расходует значительный объем продуктов, которые не реализуются, и их приходится выбрасывать. Те же проблемы возникают утром и вечером, хотя гораздо меньше сотрудников завтракает и ужинает, чем обедает.

Многие сотрудники попросили создать систему, которая позволила бы посетителям кафетерия делать заказ (определенный как набор из одного или большего числа блюд из меню) по сети, чтобы его можно было забрать или заказать доставку в назначенное место в офисе компании в указанный день в указанное время. Подобная система сэкономит значительное время и позволит получать те блюда, которые хотят клиенты. Это улучшит как качество рабочей среды, так и производительность труда. Кроме того, предоставленные заранее сведения о том, какие блюда хотят получить клиенты, позволит уменьшить потери и увеличит эффективность работы персонала кафетерия. Если сотрудники получают в будущем возможность заказывать доставку блюд из близлежащих ресторанов, то у них появится больший выбор при меньшей цене, так как фирма сможет заключать оптовые соглашения с ресторанами.

Видение решения

Для сотрудников, желающих заказывать еду в кафетерии компании или в местных ресторанах через Интернет, Cafeteria Ordering System — это ин-

тернет-приложение или приложение для смартфона, которое принимает индивидуальные или групповые заявки на питание, взимает оплату и инициирует доставку готовых блюд к указанному пункту на территории Process Impact. В отличие от имеющихся в настоящее время служб заказа по телефону и вручную, сотрудникам, использующим Cafeteria Ordering System, не придется приходить в кафетерий, чтобы получать заказанные блюда, что сэкономит им время, кроме того, увеличится ассортимент доступных им блюд.

5.2 Пример спецификации требований

Обратите внимание! 1. Курсивом выделены рекомендации к содержанию раздела (подраздела) спецификации. После ***** следует пример изложения раздела (подраздела) спецификации.

2. Нумерация рисунков в примере – сквозная.

1. Введение

Введение представляет собой обзор, помогающий читателям разобраться в структуре и принципе использования спецификации требований к ПО.

1.1. Назначение

Определите продукт или приложение, требования для которого указаны в этом документе, в том числе редакцию или номер выпуска. Если эта спецификация требований к ПО относится только к части системы, идентифицируйте эту часть или подсистему. Опишите типы читателей, которым адресован этот документ, например разработчикам, менеджерам проектов, маркетологам, пользователям, тестирующим или составителям документации.

Эта спецификация требований к ПО описывает функциональные и нефункциональные требования к выпуску 1.0 Cafeteria Ordering System (COS).

Этот документ предназначен для команды, которая будет реализовывать и проверять корректность работы системы. Кроме специально обозначенных случаев, все указанные здесь требования имеют высокий приоритет и приписаны к выпуску 1.0.

1.2. Соглашения, принятые в документах

Опишите все стандарты или типографические соглашения, включая значение стилей текста, особенности выделения или нотацию. Если вы нумеруете требования вручную, можно определить принятый формат на случай, если кому-нибудь позже понадобится добавить требование.

В этой спецификации нет никаких типографических условных обозначений.

1.3. Границы проекта

Кратко опишите ПО и его назначение. Покажите, как связан продукт с пользователями или корпоративными целями, а также с бизнес-целями и стратегиями. Если имеется отдельный документ о концепции и границах проекта, не повторяйте его содержимое, а просто сошлитесь на него. Если спецификацию требований к ПО предполагается разрабатывать постепенно, она должна содержать собственное положение о концепции и границах продукта в качестве подраздела долгосрочной стратегической концепции. Можно предоставить высокоуровневую сводку главной функциональности выпуска или функций, которые он должен выполнять.

Cafeteria Ordering System позволит сотрудникам Process Impact заказывать блюда в кафетерии компании через Интернет для доставки в указанные пункты на территории компании. Детальное описание продукта приведено в документе «Cafeteria Ordering System Vision and Scope Document» [1], где перечислены функции, полная или частичная реализация которых запланирована в этом выпуске.

1.4. Ссылки

Перечислите все документы или другие ресурсы, на которые вы ссылаетесь в этой спецификации, в том числе гиперссылки на них, если их местоположение меняться не будет. Это могут быть руководства по стилям пользовательского интерфейса, контракты, стандарты, спецификации к системным требованиям, спецификации интерфейса и спецификации требований к ПО связанных продуктов. Объем информации должен быть достаточным для того, чтобы пользователь сумел при необходимости получить доступ к каждому указанному материалу, а именно: название, имя автора, номер версии, дата, источник, место хранения или URL-адрес.

1. Вигерс, Карл. Документ о видении и объеме системы заказа кафетерия, [www.rocessimpact.com/projects/COS/COS Видение и Scope.docx](http://www.rocessimpact.com/projects/COS/COS%20Видение%20и%20Scope.docx)
2. Битти, Джой. Process Impact Intranet Development Standard, версия 1.3, [www.processimpact.com/corporate/standards/PI Intranet Development Standard.pdf](http://www.processimpact.com/corporate/standards/PI%20Intranet%20Development%20Standard.pdf)
3. Рат, Эндрю. Process Impact Internet Application User Interface Standard, версия 2.0 [www.processimpact.com/corporate/standards/PI Internet UI Standard](http://www.processimpact.com/corporate/standards/PI%20Internet%20UI%20Standard.pdf). Формат pdf.

2. Общее описание

В этом разделе представлен общий обзор продукта и среды, в которой он будет применяться, предполагаемая пользовательская аудитория, а также известные ограничения, предположения и зависимости.

2.1. Общий взгляд на продукт

Опишите контекст и происхождение продукта. Поясните, является он новым членом растущего семейства продуктов, новой версией существующей системы, заменой существующего приложения или совершенно новым продуктом.

Если спецификация требований определяет компонент более крупной системы, укажите, как это ПО соотносится со всей системой и определи-

те основные интерфейсы между ними. Неплохо также включить визуальные модели, такие как, например контекстная диаграмма, чтобы показать взаимосвязь продукта с другими системами.

Cafeteria Ordering System — это новая система, которая заменяет текущие ручные процессы заказа и получения обедов в кафетерии Process Impact.

Контекстная диаграмма на рисунке 1 показывает внешние объекты и системные интерфейсы для версии 1.0.

Предполагается выпустить несколько версий системы, чтобы в конечном итоге удалось встроить ее в службу заказов нескольких близлежащих ресторанов, работающую через Интернет, а также в службы авторизации кредитных и дебетовых карт.

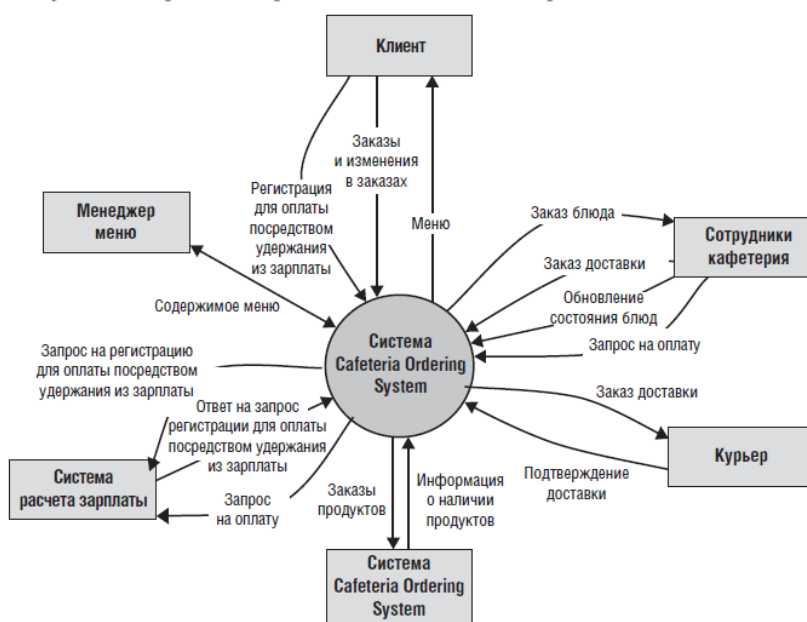


Рисунок 1 – Контекстная диаграмма для выпуска 1.0 системы Cafeteria Ordering System

2.2. Классы и характеристики пользователей

Определите различные классы пользователей, которые, как предполагается, будут работать с вашим продуктом, и опишите их соответствующим образом.

ющие характеристики. Некоторые требования могут относиться только к определенным классам пользователей. Определите привилегированные классы пользователей. Классы пользователей представляют подмножество заинтересованных в проекте лиц, их описание приводится в документе концепции и границ проекта. Описания классов пользователей являются повторно используемым ресурсом. Если есть главный каталог классов пользователей, можно включить описания классов пользователей, просто указав их в каталоге, не дублируя информацию.

Класс пользователей	Описание
Клиент (привилегированный)	Клиент — это сотрудник Process Impact, желающий заказывать питание с доставкой из кафетерия компании. Всего потенциальных клиентов — 600, из которых 400, как ожидается, будут использовать Cafeteria Ordering System в среднем 5 раз в неделю. Иногда клиенты будут заказывать питание на нескольких человек (мероприятия или гости). Ожидается, что 60% заказов будут поступать через корпоративную интрасеть, а 40% — с домашних компьютеров или с применением приложений для смартфонов или планшетов
Сотрудники кафетерия	В кафетерии Process Impact в настоящее время работает около 20 сотрудников, которые будут получать заказы через Cafeteria Ordering System, готовить блюда, упаковывать их для доставки, печатать инструкции по доставке и запрашивать доставку. Большинство сотрудников кафетерия придется обучать работе с компьютером и использованию Cafeteria Ordering System
Менеджер меню	Менеджер меню — это сотрудник кафетерия, отвечающий за создание и поддержку меню на каждый день, в котором указано, какие блюда имеются в наличии в кафетерии. Некоторые блюда в меню могут быть недоступны для доставки. Менеджер меню также

Класс пользователей	Описание
	определяет спецпредложение дня кафетерия. Менеджер меню должен периодически редактировать меню
Курьер	Готовя заказы к доставке, сотрудники кафетерия будут отправлять запросы на доставку на смартфон курьера. Курьер будет забирать заказ и доставлять их клиентам. Главное взаимодействие сотрудника по доставке с системой будет заключаться в подтверждении успеха (или неудачи) доставки

2.3. Операционная среда

Опишите рабочую среду, в которой будет работать ПО, включая аппаратную платформу, операционные системы и их версии, а также географическое местоположение пользователей, серверов и баз данных вместе с организациями, в которых располагаются соответствующие базы данных, серверы и веб- сайты. Перечислите все остальные компоненты ПО или приложения, с которыми система должна быть совместима. Если в связи с разработкой новой системы нужно произвести значительную работу с технической инфраструктурой, стоит подумать о создании отдельных требований к инфраструктуре, в которой детально изложить подробности этой работы.

ОЕ-1 Система Cafeteria Ordering System работает со следующими браузерами: Windows Internet Explorer версии 7, 8 и 9, Firefox версии с 12 по 26, Google Chrome (все версии) и Apple Safari версии с 4.0 по 8.0.

ОЕ-2 Система Cafeteria Ordering System установлена на сервере, работающем под управлением текущих утвержденных корпорацией версий Red Hat Linux и Apache HTTP Server.

ОЕ-3 Cafeteria Ordering System должна допускать доступ пользователей через корпоративную интрасеть, VPN-канал и со смартфонов и планшетов под управлением Android, iOS и Windows.

2.4. Ограничения дизайна и реализации

Бывает, что нужно использовать вполне определенный язык программирования, определенную библиотеку, на разработку которой уже потрачено время, и т. п. Опишите все факторы, которые ограничат возможности, доступные разработчикам, и логически обоснуйте каждое положение. Требования, которые включают или написаны в форме идей по решению, а не потребностей накладывают ограничения на дизайн, часто неоправданные, поэтому за этим надо следить.

СО-1 Документация системы по дизайну, коду и сопровождению должна соответствовать Process Impact Intranet Development Standard, версия 1.3 [2].

СО-2 Система должна использовать текущую версию СУБД Oracle, являющуюся корпоративным стандартом.

СО-3 Весь код HTML должен соответствовать стандарту HTML 5.0.

2.5. Предположения и зависимости

Предположение (assumption) — это утверждение, которое предполагается верным в отсутствие знаний или доказательств иного. Проблемы возможны в том случае, если предположение неверны, устарели, не находятся в совместном использовании или изменяются, поэтому определенные предположения можно отнести к группе рисков проекта. Один читатель спецификации требований к ПО может считать, что продукт будет соответствовать особому стандарту пользовательского интерфейса, тогда как другой предположит нечто совершенно иное. Разработчик может думать, что определенный набор функций написан специально для этого приложения, бизнес-аналитик — что он будет взят из предыдущего проекта, а менеджер проекта — что предполагается приобрести коммерческую библиотеку функций. Включаемые здесь предположения относятся к системной функциональности; предположения относящиеся к бизнесу представлены в документе концепции и границ проекта.

Определите все зависимости проекта или создаваемой системы от внешних факторов или компонентов вне ее контроля. Например, до установки продукта может потребоваться установить Microsoft .NET Framework 4.5 или более позднюю версию — это зависимость.

AS-1 Кафетерий открыт для завтраков, обедов и ужинов каждый рабочий день компании, когда предполагается присутствие сотрудников на рабочих местах.

DE-1 Работа Cafeteria Ordering System зависит от изменений в системе расчета зарплат, позволяющих принимать запросы на оплату за питание, заказанное через COS.

DE-2 Работа Cafeteria Ordering System зависит от изменений в системе учета запасов кафетерия, позволяющих обновлять информацию о наличии блюд по мере принятия заказов Cafeteria Ordering System.

3. Функции системы

Шаблон на рисунке 3.3 структурирован по функциям системы — это один из способов систематизации функциональных требований. Другие методы классификации — по функциональным областям, рабочим потокам, вариантам использования, режимам работы, классам пользователей, стимулам и реакциям. Возможны также иерархические комбинации этих элементов, например варианты использования внутри классов пользователей. Не существует единственно правильного метода организации; выберите тот, при котором пользователям будет легче понять предполагаемые возможности продукта.

3.1. Заказ блюд из кафетерия

Опишите название особенности несколькими словами, например «3.1 Проверка правописания».

3.1.1. Описание

Кратко опишите функцию системы и укажите ее приоритет: высокий, средний или низкий приоритет. Приоритеты являются динамичной ха-

рактической, они могут изменяться в ходе проекта.

Клиент кафетерия, личность которого подтверждена, может заказывать набор блюд либо с доставкой в указанное место на территории компании, либо для получения его в кафетерии. Клиент может отменить или изменить заказ, если блюда еще не приготовлены. Приоритет — высокий.

3.1.2. Функциональные требования

Перечислите по пунктам конкретные функциональные требования, которые связаны с этой функцией. Именно эти функции ПО нужно реализовать, чтобы пользователь мог использовать сервисы этой функции или реализовать вариант использования. Опишите, как продукт должен реагировать на ожидаемые ошибки, неправильный ввод информации или неверные действия.

Присвойте каждому функциональному требованию уникальное имя.

Заказ.Размещение:	Размещение заказа блюд
.Регистрация:	Система должна подтвердить, что клиент зарегистрирован для оплаты посредством удержания из заработной платы для размещения заказа
.Нет:	Если клиент не зарегистрирован для оплаты посредством удержания из заработной платы, система должна предложить клиенту следующие варианты: зарегистрироваться сейчас и продолжать размещать заказ, сделать заказ и самому получить его в кафетерии (без доставки) или выйти из системы
.Дата:	Система должна спрашивать клиента о дате заказа (см. BR-8)
.КрайнийСрок:	Если дата доставки заказа — текущий день, а крайний срок приема заказов уже прошел, то система должна известить клиента, что уже слишком поздно размещать заказ на сегодня. Клиент должен либо изменить дату, либо отменить заказ

Заказ.Доставка: .Выбор: .Место: .Время: .Интервалы:	Доставка или получение в кафетерии Клиент должен указать, получит ли он заказ в кафетерии, или заказ должен быть доставлен Если заказ должен быть доставлен и все еще есть свободные интервалы времени доставки на дату заказа, клиент должен указать доступное место доставки Система должна известить клиента, если на дату заказа нет доступных интервалов времени доставки. Клиент должен либо отменить заказ, либо указать, что получит его в кафетерии Система должна показывать свободные интервалы времени доставки на дату заказа Система должна позволять клиенту выбрать один из показанных интервалов доставки, сделать заказ без доставки или отменить заказ
Заказ.Меню: .Дата: .Наличие:	Просмотр меню Система должна отображать меню на выбранную дату Меню на выбранную дату должно показывать только те блюда, которые хотя бы в одном экземпляре есть в системе учета запасов кафетерия и могут быть доставлены
Заказ.Единицы: .Несколько: .СлишкомМного:	Заказ нескольких блюд и нескольких экземпляров одного блюда Система должна позволять клиенту заказывать несколько одинаковых наборов блюд, вплоть до минимального числа любого из указанных блюд в меню, если таковое есть в заказе Если клиент заказывает больше единиц одного блюда, чем в настоящее время указано в системе учета запасов кафетерия, система должна извещать клиента о максимальном количестве единиц того блюда, которое он может заказать
Заказ. Подтверждение: .Отображение:	Подтверждение заказа Когда клиент указывает, что не хочет больше зака-

.Запрос: .Ответ: .Еще:	зывать никакие блюда, система должна отобразить заказанные блюда, цены на каждое из них и сумму к оплате, подсчитанную согласно BR-12 Система должна предложить клиенту подтвердить заказ. Если клиент может подтвердить, изменить либо отменить заказ. Система должна позволять клиенту заказывать дополнительные блюда на ту же или другие даты. Включение нескольких наборов блюд в один заказ регулируют бизнес-правила BR-3 и BR-4
Заказ.Оплата: .Метод: .Доставка: .Самовывоз: .Удержание: .Да: .Нет:	Оплата заказа Когда клиент указывает, что закончил размещать заказы, система должна попросить пользователя выбрать метод оплаты См. бизнес-правило BR-11 Если клиент сам получит блюда в кафетерии, система должна предложить ему варианты оплаты: через удержание из зарплаты или наличными в кафетерии в момент получения заказа Если клиент подтвердил заказ и выбрал оплату через удержание из зарплаты, система должна выдать запрос на оплату системе расчета зарплат Если запрос на оплату принят, система должна вывести сообщение о подтверждении заказа с номером транзакции удержания из зарплаты Если запрос на оплату не принят, система должна вывести сообщение с причиной отказа. Клиент должен либо отменить заказ, либо изменить метод оплаты на «наличные» и сделать запрос на получение заказа в кафетерии
Заказ.Завершение: .Сохранение: .Запасы:	После того как клиент подтвердил заказ, система должна сделать следующее как одну транзакцию Назначить заказу следующий доступный номер и сохранить заказ с начальным состоянием «Принят» Отправить сообщение инвентарной системе кафе-

	рия, в котором указано количество единиц каждого блюда в заказе
.Меню:	Обновить меню на дату заказа, отражая возможные изменения в наличии блюд в системе управления запасами кафетерия
.Интервалы:	Обновить список остающихся доступными периодов доставки на дату заказа
.Клиент:	Отправить клиенту сообщение электронной почты с информацией о заказе и оплате
.Кафетерий:	Отправить сотрудникам кафетерия сообщение электронной почты с информацией о заказе
.Ошибка:	Если какой-либо шаг транзакции Заказ.Завершение не выполняется, система должна провести откат и сообщить пользователю, что заказ не был принят, с указанием причины неудачи

[Примечание: Функциональные требования по переупорядочению блюд и изменению и отмене блюд в этом примере не приводятся.]

3.2. Заказ блюд из ресторанов

[В этом примере детали не приводятся. Значительная часть функциональности, описанной в разделе «3.1. Заказ блюд из кафетерия» могут использоваться повторно, поэтому в этом разделе нужно указать только дополнительную функциональность, связанную с интерфейсом ресторанов.]

3.3. Создание, просмотр, модификация и удаление подписки на стандартные блюда

[В этом примере детали не приводятся.]

3.4. Создание, просмотр, модификация и удаление меню кафетерия

[В этом примере детали не приводятся.]

4. Требования к данным

Ценность информационных систем заключается в том, что они предоставляют возможность манипулировать данными. Используйте этот раздел шаблона для описания различных аспектов данных, которые будут

потреблять система в качестве входной информации, как-то обрабатывать и возвращать в виде выходной информации. Известно много схем для точного документирования данных.

4.1. Логическая модель данных

Модель данных это визуальное представление объектов и наборов данных, которые будет обрабатывать система, а также отношений между ними. Существует много видов нотации для моделирования данных, в том числе диаграммы отношений «сущность–связь» и диаграммы классов UML. Можно включить модель данных для бизнес-операций, выполняемых системой или логическое представление данных, с которыми будет работать система. Это не то же самое, что модель данных реализации, которая реализуется в виде дизайна базы данных.

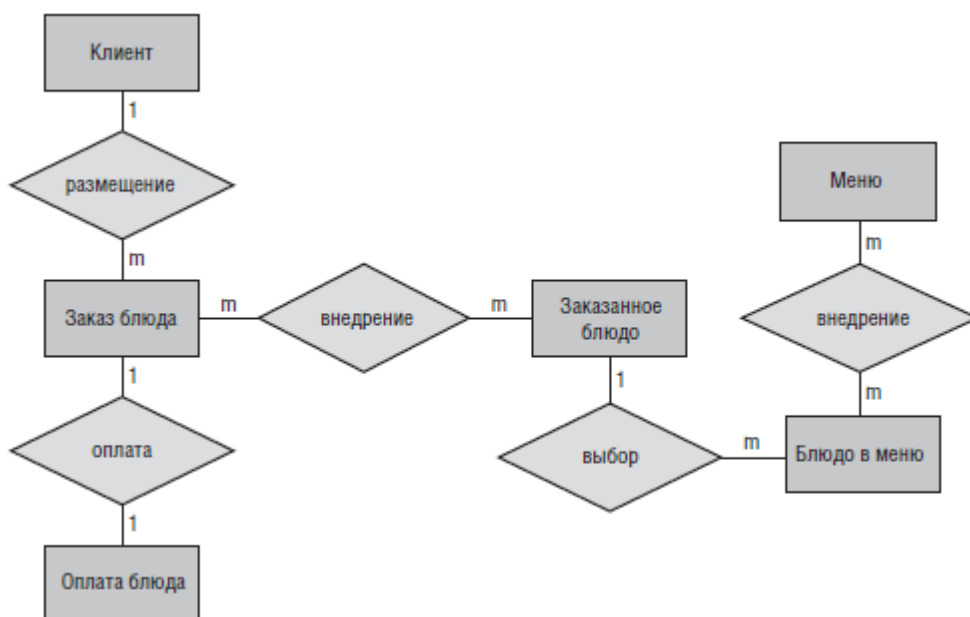


Рисунок 2 – Фрагмент модели данных
для выпуска 1.0 системы Cafeteria Ordering System

4.2. Словарь данных

Словарь данных определяет состав структур данных, а также их значение, тип данных, длину, формат и разрешенные значения элементов дан-

ных, из которых состоят эти структуры. Серийные средства моделирования данных часто включают компонент-словарь данных. Во многих случаях словарь данных лучше хранить как отдельный артефакт, не внедряя его в спецификацию требований к ПО. Это повышает возможности повторного использования в других проектах.

Элемент данных	Описание	Структура или тип данных	Длина	Значения
указания по доставке	куда и кому должно быть доставлено блюдо, если его не забирают в кафетерии	имя клиента + телефон клиента + дата доставки блюда + пункт назначения поставки + временной интервал доставки		
пункт назначения поставки	строение или комната, куда нужно доставить заказанное блюдо	алфавитно-числовое значение	50	разрешены дефисы и запятые
временной интервал доставки	начало 15-минутного интервала на дату заказа блюда, во время которого должно быть доставлено блюдо	время	чч:мм	местное время; чч = 0-23 включительно; мм = 00, 15, 30 или 45
идентификатор сотрудника	корпоративный идентификатор сотрудника, который разместил заказ	целое	6	
описание блюда	описание блюда в	буквенное значение	100	

Элемент данных	Описание	Структура или тип данных	Длина	Значения
	меню			
цена блюда	стоимость блюда до налога	числовое, рубли и копейки	pp.кк	
дата доставки блюда	дата, когда блюдо должно быть доставлено или получено в кафетерии	дата, дд.мм.гггг	10	по умолчанию — текущая дата, если она не выходит за рамки крайнего срока заказов, в противном случае — следующий день; не может быть раньше текущей даты
заказ блюда	подробности блюда, заказанного клиентом	номер заказа блюда + дата заказа + дата заказа блюда + 1: многие {заказанное блюдо} + указания по доставке + состояние заказа		
номер заказа блюда	уникальный идентификатор, назначаемый системой Cafeteria Ordering System каждому принятому заказу	целое	7	начальное значение — 1
состояние заказа	состоянием заказа, созданного клиентом	буквенное значение	16	незавершенный, принят, готовый, ожидаю-

Элемент данных	Описание	Структура или тип данных	Длина	Значения
				щий до- ставки, доставлено, отменен
оплата блюда	информация о принятом системой Cafeteria Ordering System платеже за блюдо	размер платежа + способ оплаты + номер транзакции		
меню	список блюд, доступных для покупки на определенную дату	дата меню + 1: многие {блюдо меню}		
дата меню	дата на которую доступно определенное меню	дата, дд.мм.гггг	10	
блюдо меню	описание блюда в меню	описание блюда + цена блюда		
крайний срок заказов	время дня, до которого должны делаться заказы на эту дату	время, чч: мм	5	
дата заказа	дата на которую клиент размещает заказ	дата, дд.мм.гггг	10	
заказанное блюдо	одно из блюд, которое входит в заказ клиента	блюдо меню + заказанное количество		
клиент	сотрудник Process Impact, имеющий право заказывать еду	имя клиента + идентификатор клиента + телефон клиента + местоположение		

Элемент данных	Описание	Структура или тип данных	Длина	Значения
		клиента + электронная почта клиента		
электронная почта клиента	корпоративный идентификатор клиента, который разместил заказ	алфавитно-числовое значение	50	
местоположение клиента	номера строения и комнаты сотрудника, который разместил заказ	алфавитно-числовое значение	50	разрешены дефисы и запятые
имя клиента	имя клиента, который разместил заказ	буквенное значение	30	
телефон клиента	телефон клиента, который разместил заказ	AAA-EEENNNN хXXXXX где А — код территории, Е — телефонная станция, N — номер и X — расширение	18	
размер платежа	общая цена заказа в рублях и копейках, вычисленная в соответствии с BR-12	числовое, рубль и копейки	rrrrr. kk	
способ оплаты	как клиент оплачивает заказ	буквенное значение	16	удержание из зарплаты, наличные, кредитная карта, дебетовая карта
заказанное количество	число единиц	целое	4	по умолчанию

Элемент данных	Описание	Структура или тип данных	Длина	Значения
число	каждого блюда, который клиент включил в заказ			— 1; максимум — количество, имеющееся в запасе
номер транзакции	уникальное последовательное число, которое Cafeteria Ordering System назначает каждой платежной операции	целое	12	

4.3. Отчеты

Если приложение будет генерировать отчеты, перечислите их здесь и опишите их характеристики. Если отчет должен соответствовать определенному готовому макету, можно указать это как ограничение, возможно с примером. В противном случае сосредоточьтесь на логических описаниях, порядке сортировки, уровнях суммирования и т. п., отложив подробный макет до этапа дизайна.

4.3.1. Отчет о заказанных блюдах

Идентификатор отчета	COS-RPT-1
Заголовок отчета	История заказов блюд
Цель отчета	Клиент хочет увидеть список всех блюд, которые он раньше заказывал в кафетерии или локальных ресторанах за определенный период времени вплоть до полугода до текущей даты, чтобы можно было повторно заказать понравившееся блюдо
Приоритет	Средний

Пользователи отчета	Постоянные клиенты кафетерия
Источники данных	База данных о ранее размещенных заказах блюд
Частота и использование	Отчет генерируется по запросу клиента. Данные в отчете статичны. Отчет отображается в окне веб-браузера пользователя на компьютере, планшете или смартфоне. Его можно распечатать, если устройство поддерживает печать
Время доступа	Готовый отчет должен отображаться в течение 3 секунд после его запроса
Визуальный макет	Альбомная ориентация
Верхний и нижний колонтитулы	Верхний колонтитул должен содержать заголовок отчета, имя клиента и заданный диапазон дат. При печати в нижнем колонтитуле должен содержаться номер страницы
Тело отчета	• Отображаемые поля и заголовки столбцов: • Номер заказа • Дата заказа блюда • Где заказано (кафетерий или название ресторана) • Заказанные блюда (список всех блюд в заказе с указанием их числа и цен) • Общая цена • Налог • Стоимость доставки • Итого (сумма стоимости блюд с налогом и доставкой) • Критерий отбора: Диапазон дат, определенный клиентом, включая начальную и конечную даты • Критерий сортировки: Обратный хронологический порядок
Признак конца отчета	Нет
Интерактивность	Клиент может просматривать подробности ингредиентов и сведения о пищевой ценности для каждого блюда
Ограничения безопасности доступа	Клиент может просматривать историю только своих заказов

[В этом примере не приводятся другие отчеты системы Cafeteria Ordering System.]

4.3.2. Целостность, сохранение и утилизация данных

Если это важно, опишите, как получают и обслуживают данные. Например, при открытии канала номенклатуры данных может требоваться первым делом выполнить начальный дамп всей номенклатуры данных в принимающую систему, а после этого использовать каналы для передачи только изменений. Укажите все требования, относящиеся к защите целостности данных системы. Укажите все процедуры, которые могут потребоваться, например резервное копирование, создание контрольных точек, зеркальное отображение или проверка корректности данных. Укажите политики, которые должна применять система для хранения или утилизации данных, в том числе временных данных, метаданных, остаточных данных (таких как удаленные записи), данных в кеше, локальных копий, архивов и промежуточных архивов.

DI-1 Cafeteria Ordering System должна хранить заказы клиента на протяжении 6 месяцев в даты доставки.

DI-2 Cafeteria Ordering System должна хранить меню на протяжении года с даты, указанной на меню.

5. Требования к внешним интерфейсам

В этом разделе указывается информация, которая гарантирует, что система будет правильно взаимодействовать с пользователями и компонентам внешнего оборудования и ПО. Выработка согласия по внешнему и внутреннему интерфейсу системы признано оптимальным приемом в области разработки ПО. В сложной системе с множеством подкомпонентов следует использовать отдельные спецификации для интерфейсов или спецификацию системной архитектуры. В документацию по интерфейсу можно включить ссылки на материал из других документов. Например, ссылка

может указать на руководство по работе с устройством, где перечислены коды ошибок, которые устройство может отправить программе.

5.1. Пользовательские интерфейсы

Опишите логические характеристики каждого пользовательского интерфейса, который необходим системе. Некоторые особенные характеристики пользовательских интерфейсов могут упоминаться в разделе «6.1 Удобство использования». Некоторые из них перечислены здесь:

- ссылки на стандарты графического интерфейса пользователей или стили-вые рекомендации для семейства продуктов, которые необходимо соблюдать;

- стандарты шрифтов, значков, названий кнопок, изображений, цветовых схем, последовательностей полей вкладок, часто используемых элементов управления, графики фирменного стиля, уведомления о зарегистрированных товарных знаках и о конфиденциальности и т.п.;

- размер и конфигурация экрана или ограничения разрешения;

- стандартные кнопки, функции или ссылки перемещения, одинаковые для всех экранов, например кнопка справки;

- сочетания клавиш;

- стандарты отображения и текста сообщений;

- стандарты проверки данных (такие как ограничения на вводимые значения и когда нужно проверять содержимое полей);

- стандарты конфигурации интерфейса для упрощения локализации ПО;

- специальные возможности для пользователей с проблемами со зрением, различением цвета и другими ограничениями.

UI-1 Экраны Cafeteria Ordering System должны соответствовать «Process Impact Internet Application User Interface Standard, Version 2.0» [4].

UI-2 Система должна обеспечивать ссылку на справку на каждой HTML-странице, объясняющую, как пользоваться этой страницей.

UI-3 Интернет-страницы должны предоставлять полную возможность

навигации и выбор блюд только при помощи клавиатуры, в дополнение к использованию мыши и клавиатуры.

5.2. Интерфейсы ПО

Опишите связи продукта и других компонентов ПО (идентифицированные по имени и версии), в том числе другие приложения, базы данных, операционные системы, средства, библиотеки, веб-сайты и интегрированные серийные компоненты. Укажите назначение, форматы и содержимое сообщений, данных и контрольных значений, обмен которыми происходит между компонентами ПО. Опишите соответствия между входными и выходными данными между системами и все преобразования, которые должны происходить с данными при перемещении между системами. Опишите службы, необходимые внешним компонентам ПО, и природу взаимодействия между компонентами. Определите данные, которыми будут обмениваться и к которым будут иметь общий доступ компоненты ПО. Определите нефункциональные требования, влияющие на интерфейс, такие как уровни обслуживания для времени и частоты отклика или меры и ограничения безопасности. Часть этой информации может быть определена как требования к данным в разделе 4 или как требования к взаимодействию в разделе «б. Атрибуты качества».

SI-1 Система учета запасов кафетерия.

SI-1.1 Cafeteria Ordering System должна передавать количество единиц заказанных блюд системе учета запасов кафетерия через программный интерфейс.

SI-1.2 Cafeteria Ordering System должна опрашивать систему учета запасов кафетерия для определения наличия запрашиваемого блюда.

SI-1.3 Когда система учета запасов кафетерия сообщает Cafeteria Ordering System, что определенного блюда нет в наличии, Cafeteria Ordering System должна убирать это блюдо из меню на текущую дату.

SI-2 Система расчета зарплаты

Cafeteria Ordering System должна сообщаться с системой расчета зарплат через программный интерфейс, выполняя следующие операции.

SI-2.1 Позволять клиенту регистрироваться и отменять регистрацию для оплаты через удержания из зарплаты.

SI-2.2 Проверять, зарегистрирован ли клиент для оплаты посредством удержания из зарплаты.

SI-2.3 Проверять, может ли клиент регистрироваться для оплаты посредством удержания из зарплаты.

SI-2.4 Передавать запрос на оплату приобретенного набора блюд.

SI-2.5 Возвращать полностью или частично предыдущую оплату, если клиент отменил заказ, или не был удовлетворен им, или заказ не был доставлен согласно подтвержденным инструкциям по доставке.

5.3. Интерфейсы оборудования

Опишите характеристики каждого интерфейса между компонентами ПО и оборудования системы. В описание могут входить типы поддерживаемых устройств, взаимодействия данных и элементов управлений между ПО и оборудованием, а также протоколы взаимодействия, которые будут использоваться. Перечислите входные и выходные данные, их формат, разрешенные значения или их диапазоны, а также все временные характеристики, о которых должны знать разработчики. Если такой информации очень много, лучше создать отдельный документ спецификации интерфейса.

Интерфейсы оборудования не выявлены.

5.4. Коммуникационные интерфейсы

Укажите требования для любых функций взаимодействия, которые будут использоваться продуктом, включая электронную почту, веб-браузер, сетевые протоколы и электронные формы. Определите соответствующие форматы сообщений. Опишите особенности безопасности взаимодействия

или шифрования, скорости передачи данных и механизмов согласования и синхронизации. Укажите все ограничения этих интерфейсов, например допустимость тех или иных типов вложений в сообщениях электронной почты.

CI-1 Cafeteria Ordering System должна отправлять клиенту сообщение электронной почты или СМС-сообщение (определяется параметрами учетной записи) с подтверждением принятия заказа, ценой и инструкциями по доставке.

CI-2 Cafeteria Ordering System должна отправлять клиенту сообщение электронной почты с или СМС-сообщение (определяется параметрами учетной записи) о любых проблемах, возникших с заказом или его доставкой после принятия заказа.

6. Атрибуты качества

В этом разделе описываются нефункциональные требования помимо ограничений, описанных в разделе 2.4, и требований к внешним интерфейсам, описанным в разделе 5. Эти характеристики должны быть точно определены и поддаваться проверке и измерению. Укажите относительные приоритеты различных атрибутов, например приоритет простоты использования над легкостью изучения или приоритет безопасности над производительностью.

6.1. Требования по удобству использования

Требования к удобству использования подразумевают легкость изучения, простоту использования, предотвращение ошибок и восстановление, эффективности взаимодействия и специальные возможности. Указанные в этом разделе требования к удобству использования помогут дизайнеру интерфейсов создать максимально удобную для пользователя рабочую среду.

USE-1 Система должна позволять клиенту извлечь ранее заказанное

блюдо одной операцией.

USE-2 95% новых пользователей должны суметь успешно ввести заказ без ошибок с первой попытки.

6.2. Требования к производительности

Укажите конкретные требования к производительности для различных системных операций. Если у различных функциональных требований или функций имеются разные требования к производительности, то следует указывать задачи, связанные с производительностью, там же, в разделе соответствующих функциональных требований, а не включать их все в этот раздел.

PER-1 Система должна обслуживать всего 400 пользователей и 100 пользователей в период пиковой активности с 9:00 до 10:00 по местному времени, со средней продолжительностью сеанса 8 минут.

PER-2 Все веб-страницы, генерируемые системой, должны полностью загружаться не более чем за 4 секунды после запроса их по интернет-подключению со скоростью 20 Мбит/сек.

PER-3 Система должна выводить пользователю сообщение о подтверждении в среднем за 3 секунды и не более чем через 6 секунд после того, как пользователь отослал информацию системе.

6.3. Требования к безопасности

Укажите все требования, касающиеся безопасности или конфиденциальности, которые ограничивают доступ или возможности использования продукта. Это может быть физическая безопасность, а также защита данных или ПО. Источником требований к безопасности, как правило, являются бизнес правила, поэтому определите политики или положения, касающиеся защиты или конфиденциальности, которым продукт должен соответствовать. Если они задокументированы и хранилище бизнес-правил, просто сошлитесь на них.

SEC-1 Все сетевые транзакции, включающие финансовую или подпадающую учету личную информацию, должны быть зашифрованы согласно бизнес-правилу BR-33.

SEC-2 Пользователи обязательно регистрируются для входа в Cafeteria Ordering System для выполнения всех операций, кроме просмотра меню.

SEC-3 Система должна позволять только сотрудникам кафетерия, внесенным в список авторизованных менеджеров меню, создавать или изменять меню, согласно бизнес-правилу BR-24.

SEC-4 Система должна позволять клиентам просматривать только заказы, размещенные ими лично, но не другими клиентами.

6.4. Требования к защите

В этом разделе укажите требования, связанные с возможными потерями, повреждениями или ущербом, которые могут быть результатом использования продукта. Определите меры безопасности или упреждающие действия, которые можно предпринять, так же как и потенциально опасные действия, которые можно предотвратить. Определите сертификаты по безопасности, политики или положения, которым продукт должен соответствовать.

SAF-1 У пользователя должна быть возможность увидеть список всех ингредиентов каждого блюда, причем ингредиенты, известные тем, что могут вызывать аллергическую реакцию у 0,5% населения Северной Америки, должны быть выделены особо.

6.5. Требования к доступности

AVL-1 Cafeteria Ordering System должна быть доступна 98% времени между 5:00 и полуночью по местному времени и 90% времени между полуночью и 5:00 по местному времени, за исключением времени планового обслуживания.

6.6. Требование к надежности

ROB-1 Если соединение между пользователем и системой разрывается до того, как заказ подтвержден или отменен, Cafeteria Ordering System должна позволять пользователю восстановить незавершенный заказ и продолжить работу.

Приложение А. Словарь терминов

Определите все специальные термины, которые читателю необходимо знать для правильного понимания спецификации требований к ПО, включая сокращения и аббревиатуры. Расшифруйте каждое сокращение и приведите его определение. Обратите внимание, что определения данных находятся в словаре данных, а не терминов.

[В этом примере не приводится словарь терминов]

Приложение Б. Модели анализа

В этом необязательном разделе описывается, а точнее напоминает, о таких моделях анализа, как диаграммы потоков данных, деревья функций, диаграммы переходов состояния и диаграммы «сущность-связь».

Часто читателю удобнее, когда определенные модели внедрены в соответствующие разделы спецификации, а не собраны скопом в конце.

На рисунке 3 показана диаграмма состояний, где отображено возможное состояние заказа блюд и его возможные изменения.

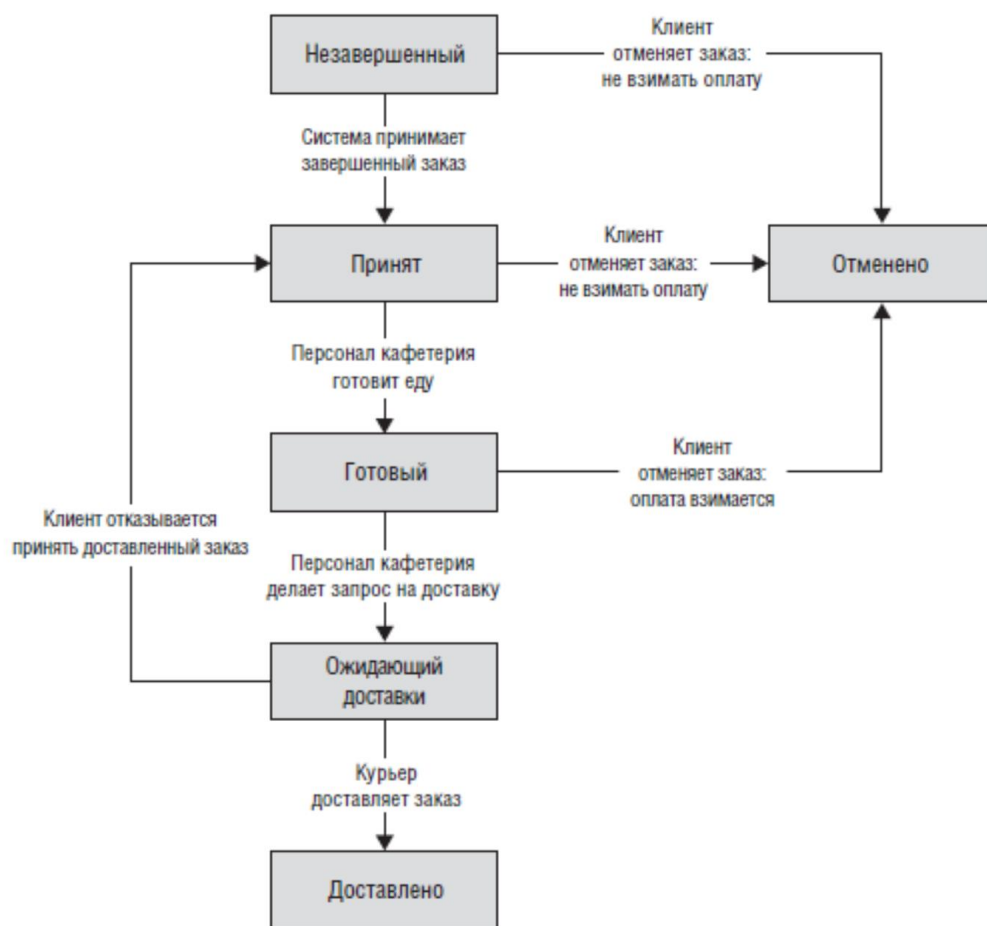


Рисунок 3 – Диаграмма состояний для состояния заказов блюд

Список рекомендуемых источников

1. Вигерс Карл. Разработка требований к программному обеспечению. / К. Вигерс, Б. Джой; Пер. с англ.— 3-е изд., дополненное — М. : Издательство «Русская редакция» ; СПб. : БХВ-Петербург, 2014. — 736 стр. : ил. — ISBN 978-5-7502-0433-5 («Русская редакция»); ISBN 978-5-9775-3348-5 («БХВ-Петербург»). — Текст : непосредственный.
2. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебное пособие / Х. Г. Ахмедова. — Москва : РТУ МИРЭА, 2023. — 104 с. — ISBN 978-5-7339-1934-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/382694> (дата обращения: 23.12.2024).
3. Проектирование информационных систем : учебник и практикум для вузов / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 273 с. — (Высшее образование). — ISBN 978-5-534-20361-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/558007> (дата обращения: 23.12.2024).
4. Гутгарц, Р. Д. Проектирование автоматизированных систем обработки информации и управления : учебное пособие для вузов / Р. Д. Гутгарц. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 351 с. — (Высшее образование). — ISBN 978-5-534-15761-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/541196> (дата обращения: 23.12.2024).
5. Астапчук, В. А. Корпоративные информационные системы: требования при проектировании : учебное пособие для вузов / В. А. Астапчук, П. В. Терещенко. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 175 с. — (Высшее образование). — ISBN 978-5-534-16715-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/531569> (дата обращения: 23.12.2024).