Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

**ОТЧЕТ**

**О ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине "Проектная документация"

2019-2020 учебный год

по теме:

РАЗРАБОТКА ФРАГМЕНТА КОРПОРАТИВНОГО СТАНДАРТА ДОКУМЕНТИРОВАНИЯ ЭТАПА ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ДАННЫХ»

Руководитель практической работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Литвинова

подпись

**СПИСОК ИСПОЛНИТЕЛЕЙ**

Исполнитель практической работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Наскальнюк

подпись

Санкт-Петербург

2020

**РЕФЕРАТ**

Отчет 26 с., 8 рис., 2 табл., 8 источников, 1 прил.

Объектом исследования является формирование требований к системе, как один из объектов жизненного цикла информационной системы.

Цель работы – разработка фрагмента корпоративного стандарта документирования этапа жизненного цикла программного обеспечения «Проектирование архитектуры данных».

Основными задачами являются:

* выбор стандартов по выбранной области;
* выделение основных понятий выбранной области;
* формирование тезаурусной статьи;
* анализ стандартизированной документации по выбранной области.

Результатом работы является сформированный фрагмент требований корпоративного стандарта документирования этапа жизненного цикла программного обеспечения «проектирование архитектуры данных».

## 

## СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc38704975)

[ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 4](#_Toc38704976)

[ВВЕДЕНИЕ 5](#_Toc38704977)

[ОСНОВНАЯ ЧАСТЬ 6](#_Toc38704978)

[1. Анализ этапа разработки ПО 6](#_Toc38704979)

[1.1. Формирование перечня стандартов 6](#_Toc38704980)

[1.2. Выделение ключевых понятий этапа жизненного цикла ПО из стандартов. Сопоставление определений этих понятий из разных стандартов 8](#_Toc38704981)

[1.2.1. Основные понятия 8](#_Toc38704982)

[1.2.2. Дополнительные понятия 9](#_Toc38704983)

[1.3 Соотнесение ключевых понятий между собой, построение тезауруса 11](#_Toc38704984)

[2. Анализ рекомендаций стандартов и лучших практик документирования процессов и результатов этапа жизненного цикла ПО, разработка фрагмента корпоративного стандарта документирования этапа жизненного цикла ПО 11](#_Toc38704985)

[2.1. Определение номенклатуры документов 11](#_Toc38704986)

[2.2 Формирование требований к содержанию и оформлению, разработка шаблонов документов и примеров их заполнения 13](#_Toc38704987)

[2.2.1 Точка зрения на архитектуру (согласно ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011) 13](#_Toc38704988)

[2.2.2. Data View 16](#_Toc38704989)

[2.3. Разработка регламента работы с документами 22](#_Toc38704990)

[**ЗАКЛЮЧЕНИЕ** 24](#_Toc38704991)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 25](#_Toc38704992)

[**ПРИЛОЖЕНИЯ** 26](#_Toc38704993)

[Приложение А 26](#_Toc38704994)

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете о НИР применяют следующие термины, обозначения и сокращения с соответствующими определениями:

*Информационная система (ИС)* – это среда, обеспечивающая целенаправленную деятельность организации и представляющая собой совокупность таких компонентов как информация, процедуры, персонал, аппаратное и программное обеспечение. Все это объединено регулируемыми взаимоотношениями для формирования организации как единого целого и обеспечения ее целенаправленной деятельности.

*Автоматизированная система (АС)* – это система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.

*Программное обеспечение (ПО)* — программа или множество программ, используемых для управления компьютером.

*Функциональная архитектура* – набор функций и их подфункций, определяющих преобразования, осуществляемые системой при выполнении своего назначения.

*Информационная архитектура* — сочетание схем организации, предметизации   
и навигации, реализованных в информационной системе.

## ВВЕДЕНИЕ

Задачей практикума по дисциплине «Проектная документация» является разработка фрагмента корпоративного стандарта документирования для одного этапа жизненного цикла ПО.

Практикум состоит из трех модулей, каждый из которых включает в себя ряд этапов. Общая структура этапов практикума:

Модуль 1. Анализ этапа разработки ПО.

1. Формирование перечня стандартов.

2. Выделение ключевых понятий этапа жизненного цикла ПО из стандартов. Сопоставление определений этих понятий из разных стандартов

3. Соотнесение ключевых понятий между собой, построение тезауруса.

Модуль 2. Анализ рекомендаций стандартов и лучших практик документирования процессов и результатов этапа жизненного цикла ПО, разработка фрагмента корпоративного стандарта документирования этапа жизненного цикла ПО.

4. Определение номенклатуры документов.

5. Формирование требований к содержанию и оформлению, разработка шаблонов документов и примеров их заполнения.

6. Разработка регламента работы с документами.

Модуль 3. Заключительный этап. Оформление отчета о выполнении практикума в соответствии с требованиями стандартов

## ОСНОВНАЯ ЧАСТЬ

## 1. Анализ этапа разработки ПО

### 1.1. Формирование перечня стандартов

*Таблица 1 Международные и российские стандарты, описывающие выбранный жизненный цикл программного обеспечения*

|  |  |
| --- | --- |
| Стандарт | Область применения |
| ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011 Системная и программная инженерия. Описание архитектуры | Настоящий стандарт обращается к созданию, анализу и самообеспечению архитектур систем с помощью описаний архитектуры.  Настоящий стандарт обеспечивает основную онтологию для описания архитектуры. Условия настоящего стандарта способствуют реализации желаемых свойств описаний архитектуры. В настоящем стандарте также определены условия для реализации желаемых свойств структур архитектуры и языков описания архитектуры в порядке целесообразной поддержки разработки и использования описаний архитектуры. В настоящем стандарте содержатся основы для сравнения и объединения структур архитектуры и языков описания архитектуры путем обеспечения общей онтологии для определения их содержания. [1] |
| Дата введения |
| 01.09.2017 |
| РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов | Распространяются на автоматизированные системы, используемые в различных сферах деятельности (управление, исследование, проектирование и т.п.), включая их сочетание, и устанавливают требования к содержанию документов, разрабатываемых при создании АС. [2] |
| Дата введения |
| 01.01.1992 |
| ГОСТ Р 57193-2016. Системная и программная инженерия. Процессы жизненного цикла систем (Взамен [ГОСТ Р ИСО/МЭК 15288-2005](http://files.stroyinf.ru/cgi-bin/ecat/ecat.fcgi?b=0&c2=3&f1=%D0%93%D0%9E%D0%A1%D0%A2%20%D0%A0%20%D0%98%D0%A1%D0%9E%2F%D0%9C%D0%AD%D0%9A%2015288-2005&f2=1)) | Настоящий стандарт устанавливает общие основы для описаний процессов и применяется в отношении жизненного цикла систем, созданных человеком. Настоящий стандарт определяет множество процессов и соответствующую терминологию с инженерной точки зрения.  Процессы могут применяться на любом уровне иерархии в структуре системы. Выбранные из них множества могут быть использованы в течение жизненного цикла для управления и осуществления стадий жизненного цикла системы, что реализуется путем вовлечения всех участников, заинтересованных в достижении конечной цели-удовлетворенности заказчиков. [3] |
| Дата введения |
| 01.11.2017 |
| ISO/IEC/IEEE 24765:2017  Systems and software engineering — Vocabulary | Настоящий стандарт представляет собой словарь, применимый ко всем системам и программным разработкам. Он был подготовлен для стандартизации терминологии. Настоящий стандарт предназначен для использования в качестве справочного пособия для специалистов в области информационных технологий, а также для поощрения использования стандартов разработки систем и программного обеспечения, подготовленных ISO и связанными организациями IEEE Computer Society и Project Management Institute. Настоящий стандарт содержит ссылки на действующие исходные стандарты для определений, для дополнительного изучения концепций и требований к системам и разработке программного обеспечения. [4] |
| Дата введения |
| 28.09.2017 |
| ISO/IEC 10746-2:2009  Информационные технологии. Открытая распределенная обработка. Эталонная модель. Часть 2. Основы | SO/IEC 10746 provides a coordinating framework for the standardization of open distributed processing (ODP). This supports distribution, interworking, portability, and platform and technology independence. It establishes an enterprise architecture framework for the specification of ODP systems.  ISO/IEC 10746 defines the essential concepts necessary to specify open distributed processing systems from five prescribed viewpoints. It provides a well-developed framework for the structuring of specifications for large-scale, distributed systems. [5] |
| Дата введения |
| 15.12.2009 |
| Guide to the Systems Engineering Body of Knowledge (SEBoK) | The Systems Engineering Body of Knowledge (SEBoK), formally known as Guide to the Systems Engineering Body of Knowledge, is a wiki-based collection of key knowledge sources and references for systems engineering. [6] |
| Дата введения |
| Январь 2011 |
| ГОСТ Р 57193-2016. Системная и программная инженерия. Процессы изненного цикла систем (Взамен ГОСТ Р ИСО/МЭК 15288-2005) | Настоящий стандарт устанавливает общие основы для описаний процессов и применяется в отношении жизненного цикла систем, созданных человеком. Настоящий стандарт определяет множество процессов и соответствующую терминологию с инженерной точки зрения. Процессы могут применяться на любом уровне иерархии в структуре системы. Выбранные из них множества могут быть использованы в течение жизненного цикла для управления и осуществления стадий жизненного цикла системы, что реализуется путем вовлечения всех участников, заинтересованных в достижении конечной цели-удовлетворенности заказчиков. [7] |
| Дата введения |
| 01.11.2017 |

### 1.2. Выделение ключевых понятий этапа жизненного цикла ПО из стандартов. Сопоставление определений этих понятий из разных стандартов

### 1.2.1. Основные понятия

1. Архитектура (системы) (architecture)
   1. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Основные понятия или свойства системы в окружающей среде, воплощенной в ее элементах, отношениях и конкретных принципах ее проекта и развития.
   2. ISO/IEC 10746-2:2009. Set of rules to define the structure of a system and the interrelationships between its parts.
2. Архитектурное представление (architecture view):
   1. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Рабочий продукт, выражающий архитектуру некоторой системы с точки зрения определенных системных интересов.
   2. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Это результат применения точки зрения к конкретной рассматриваемой системе.
3. Точка зрения на архитектуру (architecture viewpoint):
   1. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Рабочий продукт, устанавливающий условности конструирования, интерпретации и использования архитектурного представления для структуризации определенных системных интересов.
   2. SEBoK. specifies the kinds of model to be used in developing the system architecture to address the concern (or set of concerns) of a particular stakeholder, the ways in which the models should be generated, and how the models are related and used to compose a view
   3. ISO/IEC 10746-2:2009. Форма абстракции, достигнутой с использованием отобранного множества архитектурных конструкций и структурирования правил в порядке сосредоточения на конкретных интересах в пределах системы
4. Описание архитектуры (architecture description):
5. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Рабочий продукт, используемый для выражения архитектуры.
6. Is an artifact describing the architecture for some system of interest.

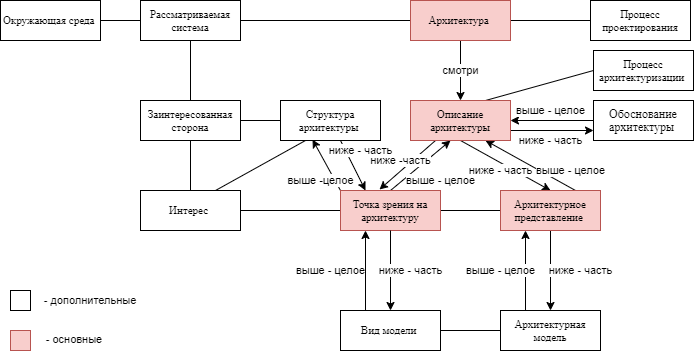
### 1.2.2. Дополнительные понятия

1. Процесс проектирования (design):
   1. ГОСТ Р 57193-2016. Процесс определения системных элементов, взаимодействий и других характеристик рассматриваемой системы в соответствии с требованиями и архитектурой.
   2. ГОСТ Р ИСО/МЭК 15288-2005. Процесс, при котором определяются и исследуются одна или несколько стратегий реализации системы со степенью детализации, соответствующей техническим и коммерческим требованиям и рискам и, исходя из этого, выбирается решение о проектировании архитектуры. Оно определяется на основе требований к набору системных элементов, из которых компонуется система.
2. Структура архитектуры (architecture framework):
3. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Условности, принципы и практики для описания архитектур, установленные в пределах заданной области применения и/или объединения заинтересованных сторон.
4. SEBoK. Collections of standardized viewpoints, views, and models that can be used when developing architectural descriptions of the enterprise
5. Процесс архитектуризации (architecting):
6. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Процесс понимания, определения, выражения, документирования, взаимодействия, соответствующей сертификации при реализации, сопровождении и улучшении архитектуры в жизненном цикле системы.
7. Заинтересованная сторона, правообладатель (stakeholder):
8. ГОСТ Р 57193-2016. Индивидуум или организация, имеющие право, долю, требование или интерес в системе или в обладании ее характеристиками, удовлетворяющими их потребности и ожидания.
9. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Индивидуум, команда, организация или их группы, имеющие интерес в системе.
10. ГОСТ Р ИСО/МЭК 15288-2005. Сторона, имеющая право, долю или претензии на систему или на владение ее характеристиками, удовлетворяющими потребности и ожидания этой стороны.
11. Интерес (системы) (concern):
12. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Польза или проблемы в системе, относящиеся к одной или нескольким заинтересованным сторонам.
13. Окружающая среда (системы) (environment):
14. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Контекст, определяющий параметры и обстоятельства всех воздействий на систему.
15. ISO/IEC/IEEE 24765:2017. anything affecting a subject system or affected by a subject system through interactions with it, or anything sharing an interpretation of interactions with a subject system
16. Вид модели (model kind):
17. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Условности для типа моделирования.
18. Рассматриваемая система (system-of-interest):
19. ГОСТ Р 57193-2016. Система, жизненный цикл которой рассматривается в рамках настоящего стандарта.
20. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Архитектура, которая находится на рассмотрении в подготовке описания архитектуры.
21. Архитектурная модель (architecture model):
22. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Составная часть архитектурного представления, использует условности моделирования, соответствующие интересам, на разрешение которых они будут направлены.
23. Обоснование архитектуры (architecture rationale):
24. Обоснование, которое регистрирует разъяснения, оправдания или рассуждения о причинах в принятых решениях архитектуры.

### 1.3 Соотнесение ключевых понятий между собой, построение тезауруса

Тезаурус — особая разновидность словарей общей или специальной лексики, в которых указаны семантические отношения (синонимы, антонимы, паронимы, гипонимы, гиперонимы и т. п.) между лексическими единицами. Таким образом, тезаурусы, особенно в электронном формате, являются одним из действенных инструментов для описания отдельных предметных областей. В отличие от толкового словаря, тезаурус позволяет выявить смысл не только с помощью определения, но и посредством соотнесения слова с другими понятиями и их группами. Задачей этапа является определение соотношений между понятиями, выделенными на предыдущем этапе.

*Рисунок 1 Тезаурус по соотнесению ключевых понятий*



## 2. Анализ рекомендаций стандартов и лучших практик документирования процессов и результатов этапа жизненного цикла ПО, разработка фрагмента корпоративного стандарта документирования этапа жизненного цикла ПО

### 2.1. Определение номенклатуры документов

Для документирования этапа «проектирование архитектуры» используется документ архитектуры программного обеспечения (Software Architecture Document).

Этот рабочий продукт предоставляет всесторонний архитектурный обзор системы с использованием ряда различных архитектурных представлений для отражения различных аспектов системы.

Методология разработки программного обеспечения Rational Unified Process (RUP) предлагает следующий шаблон SAD: [SAD\_RUP](https://drive.google.com/open?id=1C_7YFuav4mKhsFBgHEZ2Ud34OF-ybDoX)

Шаблон SAD, соответствующий ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011: [SAD\_42010](https://drive.google.com/open?id=1rE3LhWO4EOSREEEkeI7PE3N9vTL0pXDn)

Подходы к документированию архитектуры программного обеспечения:

* Модель Захмана;
* Модель "4+1";
* Стратегическая модель архитектуры SAM;
* Архитектурные концепции и методики Microsoft;
* SEI's Views and Beyond Approach for Documenting Software Architectures;
* Прочие.

Структура архитектурной документации:

* 1. Введение (описание структуры документа, Documentation Roadmap) – содержит обзор документа и его аудитории.
  2. Краткое описание программного обеспечения (Overview) – содержит описание программного обеспечения, функциональных и качественных требований.
  3. Представления – данный раздел содержит набор архитектурных представлений. В случае с архитектурой данных нас интересует data view.
  4. Глоссарий и ссылки (Directory) - содержит объяснения использованных сокращений и терминов, а также ссылки на внешние источники.

Представление или точка зрения на архитектуру является основной единицей архитектурной документации программного обеспечения. Представление (architectural view) является описанием архитектуры, которое пригодно для чтения одной из заинтересованных сторон.

Согласно ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011, точку зрения на архитектуру следует определять, как часть описания архитектуры, часть структуры архитектуры или как самостоятельную единицу. Таким образом, задокументировать этап «проектирование архитектуры данных» можно с помощью data view.

Набор архитектурных представлений:

* Use-case view
* Logical view
* Process view
* Deployment view
* Implementation view
* **Data view** (optional)

### 2.2 Формирование требований к содержанию и оформлению, разработка шаблонов документов и примеров их заполнения

### 2.2.1 Точка зрения на архитектуру (согласно ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011)

Разделы документа:

1. Наименование точки зрения

Наименование для точки зрения. Если существуют синонимы или другие общие наименования, которые известны для этой точки зрения, следует записать их.

1. Обзор точки зрения

Резюме или краткий обзор точки зрения и ее главных особенностей.

1. Интересы и "противоположные интересы"

Перечисление связанных с архитектурой интересов, которые будут структурированы этой точкой зрения. Для архитектора это является критичной информацией, т.к. она помогает решать, будет ли данная точка зрения полезна для рассматриваемой системы.

Может оказаться полезным зарегистрировать виды источников, для которых точка зрения не является приемлемой. Формулирование противоположных интересов может оказаться хорошим противодействием для определенных чрезмерно используемых моделей и нотаций.

1. Типичные заинтересованные стороны

Перечисление заинтересованных сторон системы, ожидаемых в качестве пользователей или публики для подготовленных представлений, использующих эту точку зрения.

Примечание - После того, как точка зрения выбрана для использования и применена в описании архитектуры, это описание архитектуры требуется задокументировать ассоциацией фактических заинтересованных сторон системы с интересами, структурированными с помощью каждой точкой зрения.

1. Виды моделей
   1. Введение

Определяется каждый вид модели, используемый в этой точке зрения.

Для каждого используемого вида модели описываются его соглашения, язык или методики моделирования. Они являются основными ресурсами моделирования, которые точка зрения делает доступными, и определяют словари для конструирования представления и включают операции на моделях конкретного вида моделей.

Вид модели может быть зарегистрирован многими способами, включая:

* задание метамодели, которая определяет его основные конструкции;
* обеспечение шаблона модели для заполнения пользователями;
* через языковое определение или с помощью ссылки к существующему языку моделирования;
* некоторую комбинацию этих методов.
  1. Вид модели: метамодель (опционально)

Метамодель представляет собой элементы описания архитектуры, которые включают в себя словарь вида моделей. Существуют различные способы представления метамодели. Метамодель следует представлять как:

* сущности (объекты): Каковы главные типы элементов, которые присутствуют в моделях этого вида?
* атрибуты: Какие свойства реализуют сущностные (объектовые) процессы в моделях этого вида?
* отношения: Какие отношения определены среди сущностей (объектов) в моделях этого вида?
* ограничения: Какие виды ограничений существуют для сущностей (объектов), атрибутов и/или отношений в моделях этого вида?

Сущности (объекты), атрибуты, отношения и ограничения - это все элементы описания архитектуры.

Примечание - Когда точка зрения определяет множественные виды моделей, полезно найти единственную точку зрения метамодели, унифицирующую определения видов моделей. Кроме того, часто бывает полезным использовать единственную метамодель, чтобы выразить множественные, связанные точки зрения (например такой, когда определяется структура архитектуры).

* 1. Вид модели: шаблон (опционально)

Обеспечивается шаблон или форма, определяющие формат и/или содержание моделей этого вида моделей.

* 1. Вид модели: языки (опционально)

Определяется существующая нотация или язык модели так, чтобы они могли использоваться для моделей этого вида. Описывается, если это необходимо, их синтаксис, семантика, поддерживающие инструментарии.

* 1. Вид модели: операции(опционально)

Определяются операции, доступные на моделях этого вида.

1. Правила связи

Документируются любые правила связи, определенные конкретной точкой зрения или ее видами моделей. Обычно эти правила будут "пересекающейся моделью" или "пересекающимся представлением", так как ограничения в пределах вида моделей будут определены как часть соглашений этого вида моделей.

1. Операции на представлениях

Операции определяют собой методы, которые применяются в представлениях или их моделях. Операции могут быть разделены на категории:

- методы создания - это средства, с помощью которых представления подготовлены с использованием этой точки зрения. Они могут быть представлены в форме руководства процесса (как начать, что делать в дальнейшем); руководства для рабочих продуктов (шаблоны для представлений этого типа); эвристики, стилей, образцов или других выражений;

- интерпретирующие методы - это средства, с помощью которых представления становятся понятными заинтересованным сторонам системы и читателям;

- методы анализа - используются для того, чтобы проверять, рассуждать, преобразовывать, прогнозировать, применять и оценивать архитектурные результаты из конкретного представления;

- методы проектирования и реализации - используются для того, чтобы реализовывать или конструировать системы, применяя информацию из конкретного представления.

1. Примеры

Этот раздел содержит примеры.

1. Примечания

Любая дополнительная информация, в которой пользователи этой точки зрения могут нуждаться или находят ее полезной.

1. Источники

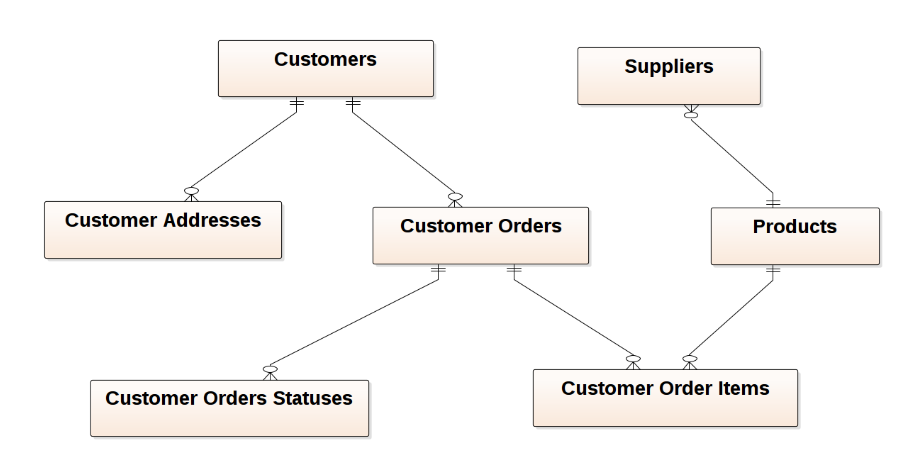
Определяются источники конкретной точки зрения, если таковые имеются, включая автора, историю, литературные ссылки, предшествующие наработки.

### 2.2.2. Data View

Data view касается архитектуры хранения, поиска, обработки, архивирования и защиты данных. Это представление рассматривает поток данных при их хранении и обработке, а также то, какие компоненты потребуются для поддержки и управления как хранением, так и обработкой. Разрабатывается для администраторов и разработчиков баз данных, системных инженеров.

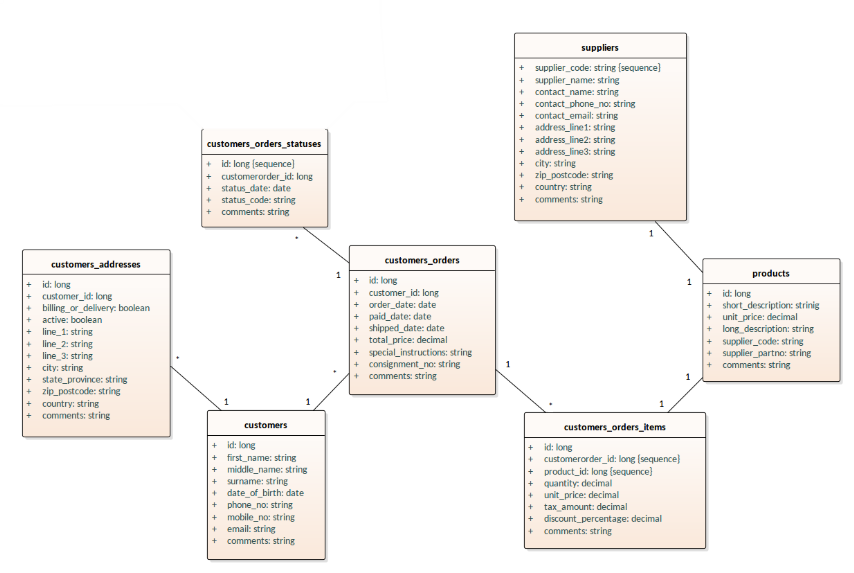
Во время определения целевого состояния архитектура данных разбивает объект на атомный уровень, а затем создает его обратно в желаемую форму. Архитектор данных разбивает субъект вниз, пройдя 3 традиционных архитектурных процесса:

* Концептуальный — представляет все бизнес-единицы.
* Логический — представляет собой логику того, как связаны сущности.
* Физический — реализация механизмов данных для определенного типа функциональности.



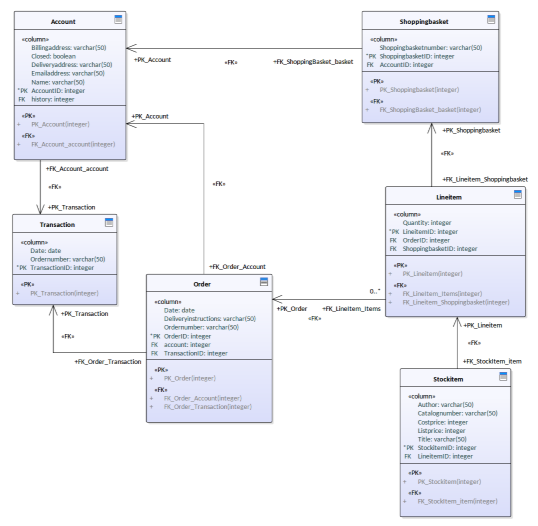
*Рисунок 2 Пример концептуальной модели данных*

Концептуальная модель данных является наиболее абстрактной формой модели данных. Это полезно для передачи идей широкому кругу заинтересованных сторон из-за своей простоты. В этом примере Концептуальной модели данных элементы определены как классы UML, а соединители изображены с использованием нотации «Информационная инженерия».



*Рисунок 3 Пример логической модели данных*

Логические модели данных помогают определить детальную структуру элементов данных в системе и взаимосвязи между элементами данных. Они уточняют элементы данных, представленные концептуальной моделью данных, и составляют основу физической модели данных.

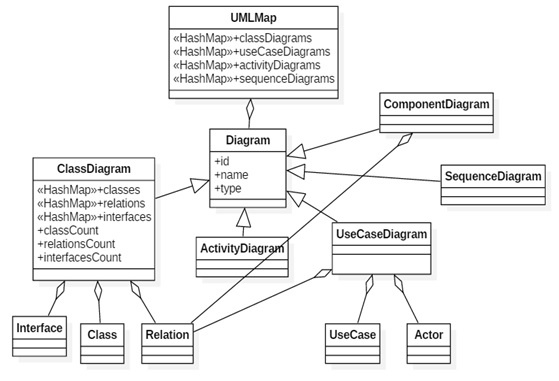


*Рисунок 4 Пример физической модели данных*

В этом примере показана физическая модель данных с использованием нотации отношений UML, которую можно использовать для автоматической генерации схемы базы данных. Каждая таблица представлена классом UML; Столбцы таблицы, первичные ключи и внешние ключи моделируются с использованием атрибутов и операций UML. Тип СУБД - PostgreSQL.

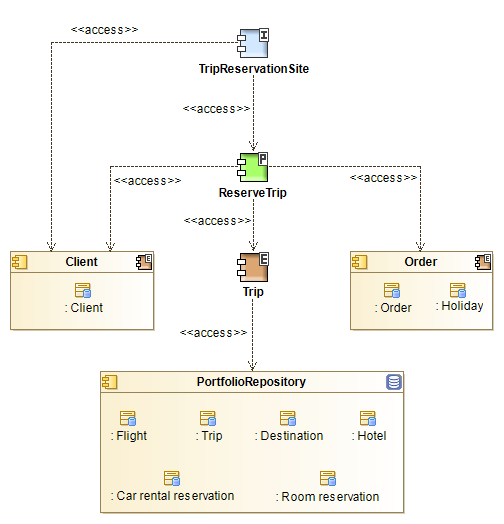
Согласно методологии TOGAF результатом данного этапа могут служить следующие диаграммы:

* Class diagram (диаграмма классов)



*Рисунок 5 Пример диаграммы классов*

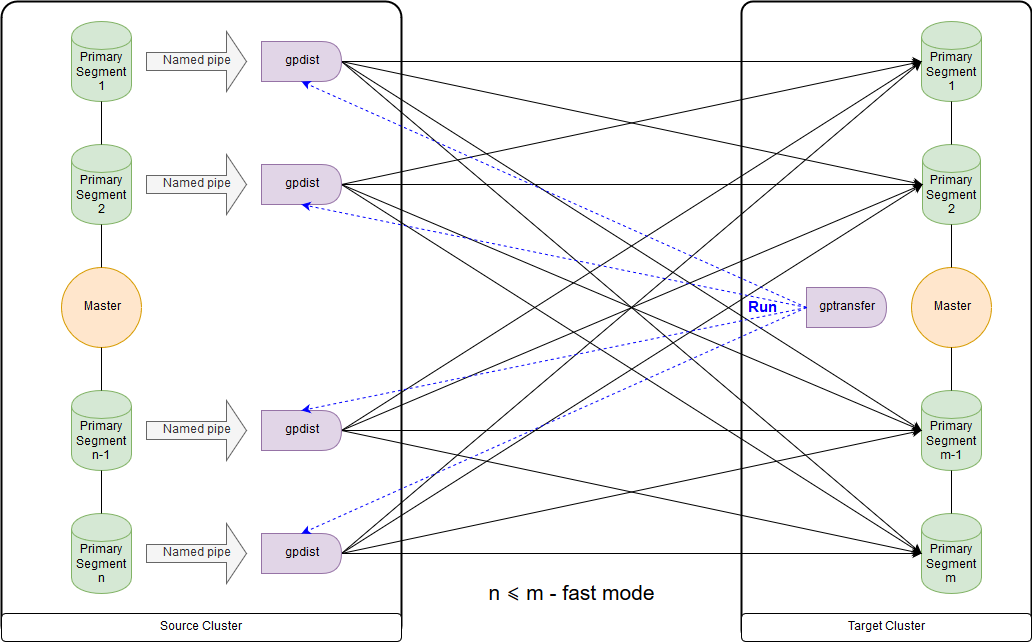
* Data dissemination diagram (диаграмма распространения данных)



*Рисунок 6 Пример диаграммы распространения данных*

Цель диаграммы распространения данных - показать взаимосвязь между объектами данных, бизнес-сервисами и компонентами приложения. Диаграмма показывает, как логические объекты должны быть физически реализованы компонентами приложения.

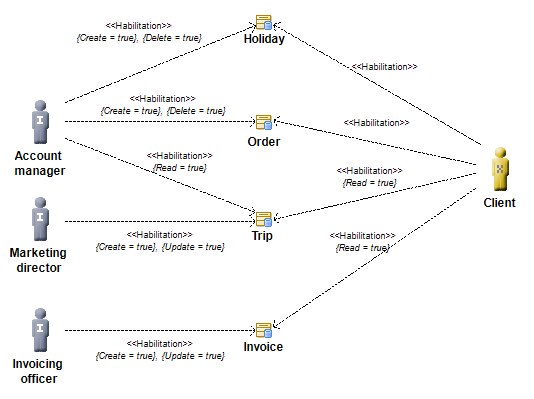
* Data migration diagrams (диаграмма миграции данных)



*Рисунок 7 Пример диаграммы миграции данных*

Цель диаграммы миграции данных - показать поток данных из источника в целевые приложения. Диаграмма обеспечивает визуальное представление о распределении источников / целей и служит инструментом для аудита данных.

* Data security diagram (диаграмма безопасности данных)



*Рисунок 8 Пример диаграммы безопасности данных*

Цель диаграммы безопасности данных - показать, какой субъект может получить доступ к данным.

*Таблица 2 Сущности, отношения, ограничения на этапе проектирования архитектуры данных*

|  |  |
| --- | --- |
| Сущности | Сущность данных.  Свойства:   * + название   + список атрибутов и их типов данных   + какие атрибуты идентифицируют сущность (первичный ключ)   + правила, используемые для предоставления разрешений пользователям или группам пользователей для доступа к объекту в базе данных   + ожидаемое количество сущностей и ожидаемый темп роста |
| Отношения | * Отношение ассоциации (association relationship) * Отношение обобщения (generalization relationship) * Отношение агрегации (aggregation relationship) * Отношение композиции (composition relationship)   Кратность: Один к одному; Один ко многим; Многие ко многим. |
| Ограничения | Нормализация базы данных часто используется для наложения ограничений на основе зависимостей между  сущностями и их атрибутами, чтобы избежать дублирования информации.  Так же на проектирование архитектуры данных могут повлиять: требования предприятия, технологические драйверы, экономика, бизнес-политики и потребности в обработке данных. |
| Для чего необходим данный этап | • служит основой для физической базы данных  • помогает взаимодействовать с заинтересованными сторонами во время анализа предметной области и выявления требований  • помогает анализировать производительность транзакций, которые включают операции с базой данных  • помогает в анализе изменчивости, чтобы оценить влияние изменений, которые затрагивают структуру базы данных  • позволяет генерировать код скриптов создания таблиц данных и кода доступа к данным |

### 

### 2.3. Разработка регламента работы с документами

Основные роли и их обязанности:

1. Руководители организаций разработчика и заказчика системы – утверждение документов
2. Руководитель проекта – согласование документов, контроль за осуществлением проводимых процессов и мероприятий, взаимодействие с представителями организации-заказчика, обеспечение готовности продукта
3. Разработчик программного обеспечения – разработка программных компонентов
4. Технический писатель – разработка документации предприятия
5. Аналитик – составление и поддержание в актуальном состоянии требований, проведение интервью и ревью со стейкхолдерами
6. Архитектор программного обеспечения – проектирование программных компонентов, разработка и систематизация технической документами
   1. Архитектор баз данных - выбор технологии для хранения данных, составление плана разработки и ТЗ для подчиненных, проектирование и оптимизация БД, обеспечение безопасности БД.

Ответственность за выполнение операций с документами:

1. Инициирование – Руководитель проекта
2. Создание – Аналитик, Архитектор программного обеспечения (Архитектор баз данных), Разработчик программного обеспечения, Технический писатель
3. Согласование – Руководитель проекта
4. Утверждение – Руководители организаций, Руководитель проекта
5. Использование – для технического задания - все роли, для остальных – представители организации-заказчика.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения первого этапа практикума был составлен перечень стандартов, используемых при проектировании архитектуры ПО. Из стандартов были выделены ключевые понятия рассматриваемого этапа жизненного цикла ПО. На их основе был построен тезаурус.

На втором этапе были проанализированы выбранные стандарты и фреймворки. На основе проведенного анализа была определена номенклатура документов, обеспечивающая документирование этапа «Проектирование архитектуры данных». Для выделенных документов были определены требования и регламент работы с ними.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011 Системная и программная инженерия. Описание архитектуры. – Введ. 2017-01-09. – М.: Изд-во стандартов, 2016.
2. ISO/IEC 10746-2:2009. Информационные технологии. Открытая распределенная обработка. Эталонная модель. Часть 2. Основы. Дата опубликования: 15.12.2009.
3. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов. – Введ. 1992-01-01. – М.: Изд-во стандартов, 2002.
4. ГОСТ Р 57193-2016. Системная и программная инженерия. Процессы жизненного цикла систем. – Введ. 2017-11-01. – М.: Изд-во стандартов, 2013.
5. ISO/IEC/IEEE. 2017. Systems and Software Engineering -- Vocabulary (SEVocab) Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronic Engineers (IEEE) 2017 ISO/IEC/IEEE 24765:2017, [Электронный ресурс] Режим доступа:   
   <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:24765:ed-2:v1:en>
6. SEBoK Editorial Board. 2019. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.1, R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. www.sebokwiki.org. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society
7. ГОСТ Р 57193-2016. Системная и программная инженерия. Процессы жизненного цикла систем. – Введ. 2017-11-01. – М.: Изд-во стандартов, 2013.
8. Data Model as an Architectural View. Paulo Merson. October 2009.TECHNICAL NOTE. CMU/SEI-2009-TN-024. [Электронный ресурс] Режим доступа: <https://resources.sei.cmu.edu/asset_files/TechnicalNote/2009_004_001_15077.pdf>

# **ПРИЛОЖЕНИЯ**

## Приложение А

**Шаблон для документирования точек зрения на архитектуру**

1. **<Наименование точки зрения>**
2. **Обзор точки зрения**
3. **Интересы и заинтересованные стороны**
   1. **Интересы**
   2. **Типичные заинтересованные стороны**
   3. **"противоположные интересы" (опционально)**
4. **Виды моделей**
5. **< Вид модели >**
   1. **Вид модели: метамодель (опционально)**
   2. **Вид модели: шаблон (опционально)**
   3. **Вид модели: языки (опционально)**
   4. **Вид модели: операции (опционально)**
6. **Правила связи**
7. **Операции на представлениях**
8. **Примеры (опционально)**
9. **Примечания (опционально)**
10. **Источники**

## 