

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 2023.12.22 18:28:15

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

ПРАКТИЧЕСКАЯ РАБОТА № 1

Планирование работ проекта, выбор метода управления проектами

Цель: научиться анализировать особенности и возможности существующих методов управления проектами, научиться осуществлять и обосновывать выбор метода управления проектом, исходя из специфики проекта.

Выполнение работы

1. Лабораторная работа выполняется индивидуально.
2. Необходимо определить проект, для которого будет проводиться анализ и выбор метода управления и организации работ.

Предлагаемые варианты:

- a. Разработка CRM-системы на заказ для SMM-агентства, в команде 1-2 человека
- b. Разработка CRM-системы собственным техническим отделом компании-производителя одежды, в команде 1-2 человека
- c. Разработка интернет-магазина на заказ для крупного производителя игрушек, в команде 1-2 человека
- d. Разработка youtube-like видеосервиса на заказ: мобильное приложение + веб-приложение + админ панель, в команде 1-3 человека
- e. Разработка игрового VR-приложения на заказ, в команде 1-2 человека
- f. Разработка системы для бронирования переговорных комнат: клиентское приложение + админ панель, собственным техническим отделом для офиса сети продуктовых магазинов, в команде 1-2 человека
- g. Разработка корпоративной нефтедобывающей компании социальной собственным сети для техническим отделом, в команде 1-2 человека

Можно также предложить и согласовать собственную идею проекта.

3. С помощью анализа открытых источников в сети интернет, необходимо определить параметры, которые влияют на выбор метода управления проектом.
4. По этим параметрам необходимо выбрать метод управления, подходящий для выбранного проекта, либо сформировать правила управления, комбинируя принципы нескольких методов. Необходимо обосновать свой выбор.
5. Результаты работы необходимо оформить в отчет в формате .doc или .pdf и прикрепить в Microsoft Teams в раздел “Задания” к заданию “Практическая работа No1”.

ПРАКТИЧЕСКАЯ РАБОТА № 2

Визуальные средства представления проекта. Основы UML.

Цель: освоить базовые приемы проектирования систем и процессов с использованием универсального языка моделирования (UML), получить навыки построения диаграмм классов UML.

Краткие теоретические сведения.

Диаграмма классов.

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации).

Основными элементами являются классы и связи между ними. Классы характеризуются при помощи атрибутов и операций.



Рисунок 1 – Основные элементы диаграммы классов

Атрибуты описывают свойства объектов класса. Большинство объектов в классе получают свою индивидуальность из-за различий в их атрибутах и взаимосвязи с другими объектами. Однако, возможны объекты с идентичными значениями атрибутов и взаимосвязей. Т.е. индивидуальность объектов определяется самим фактом их существования, а не различиями в их свойствах. Имя атрибута должно быть уникально в пределах класса. За именем атрибута может следовать его тип и значение по умолчанию.

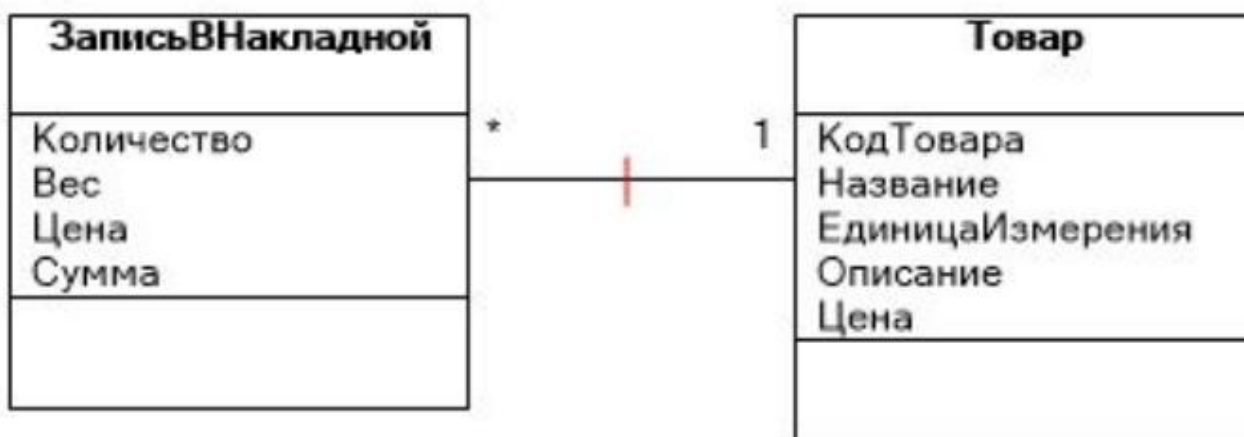
Операция есть функция или преобразование. Операция может иметь параметры и возвращать значения.

Виды связей:

- ассоциация;
- агрегация;
- наследование.

Ассоциация (association) – представляет собой отношения между экземплярами классов. Каждый конец ассоциации обладает кратностью (синоним – мощностью, ориг. – multiplicity), которая показывает, сколько объектов, расположенных с соответствующего конца ассоциации, может участвовать в данном отношении. В примере на рисунке каждый Товар имеет сколь угодно Записей в накладной, но каждая Запись в накладной обязательно один Товар. В общем случае кратность может быть задана любым множеством.

Ассоциации может быть присвоено имя. В качестве имени обычно выбирается глагол или глагольное словосочетание, сообщающие смысл и назначение связи. Также, на концах ассоциации под кратностью может указываться имя роли, т.е. какую роль выполняют объекты, находящиеся с данного конца ассоциации.



Агрегация (aggregation) – это ассоциация типа «целое-часть».

Агрегация в UML представляется виде прямой с ромбом на конце.

Ромб на связи указывает, какой класс является агрегирующим (т.е. «состоящим из»), – класс с противоположного конца – агрегированным (т.е. те самые «части»).

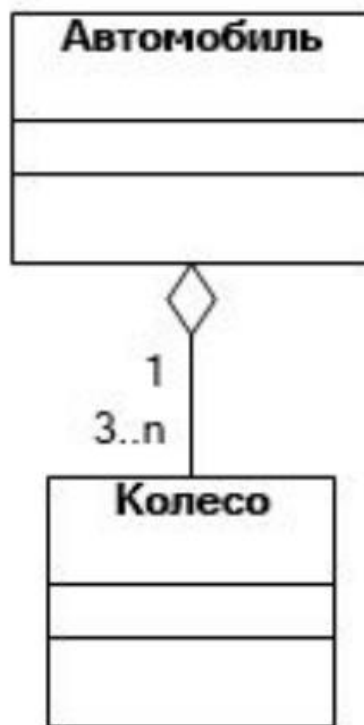


Рисунок 3 – Агрегация

Композиция (composition) – это такая агрегация, где объекты-части не могут существовать сами по себе и уничтожаются при уничтожении объекта агрегирующего класса. Композиция изображается так же, как ассоциация, только ромбик закрашен.



Рисунок 4 – Композиция

Важно понимать разницу между агрегацией и композицией: при агрегации объекты-части могут существовать сами по себе, а при композиции – нет. Пример агрегации: автомобиль-колесо, пример композиции: дом-комната.

Наследование (inheritance) – это отношение типа «общее-частное».

Позволяет определить такое отношение между классами, когда один класс обладает поведением и структурой ряда других классов. При создании производного класса на основе базового (одного или нескольких) возникает иерархия наследования. Реализация принципов наследования является ключевой предпосылкой возможности повторного использования кода, поскольку это основной инструмент достижения полиморфизма.

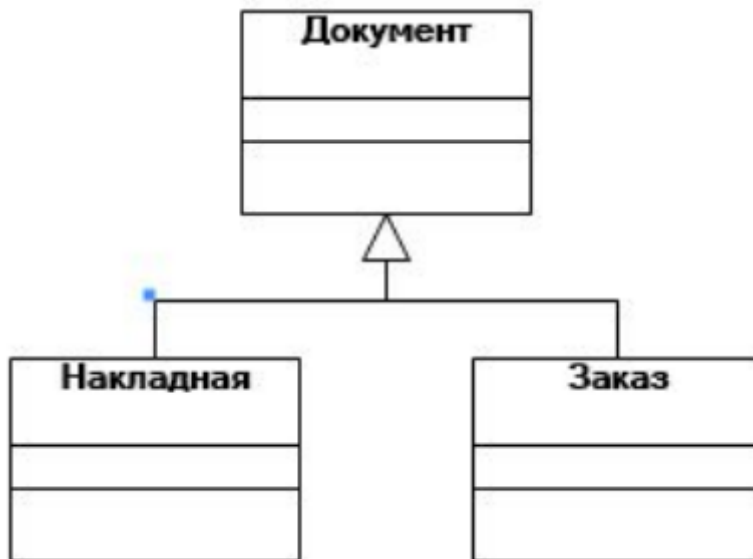


Рисунок 5 – Наследование

Для построения диаграммы классов необходимо проанализировать предметную область и построить диаграмму классов. Должна получиться диаграмма, подобная следующему примеру.

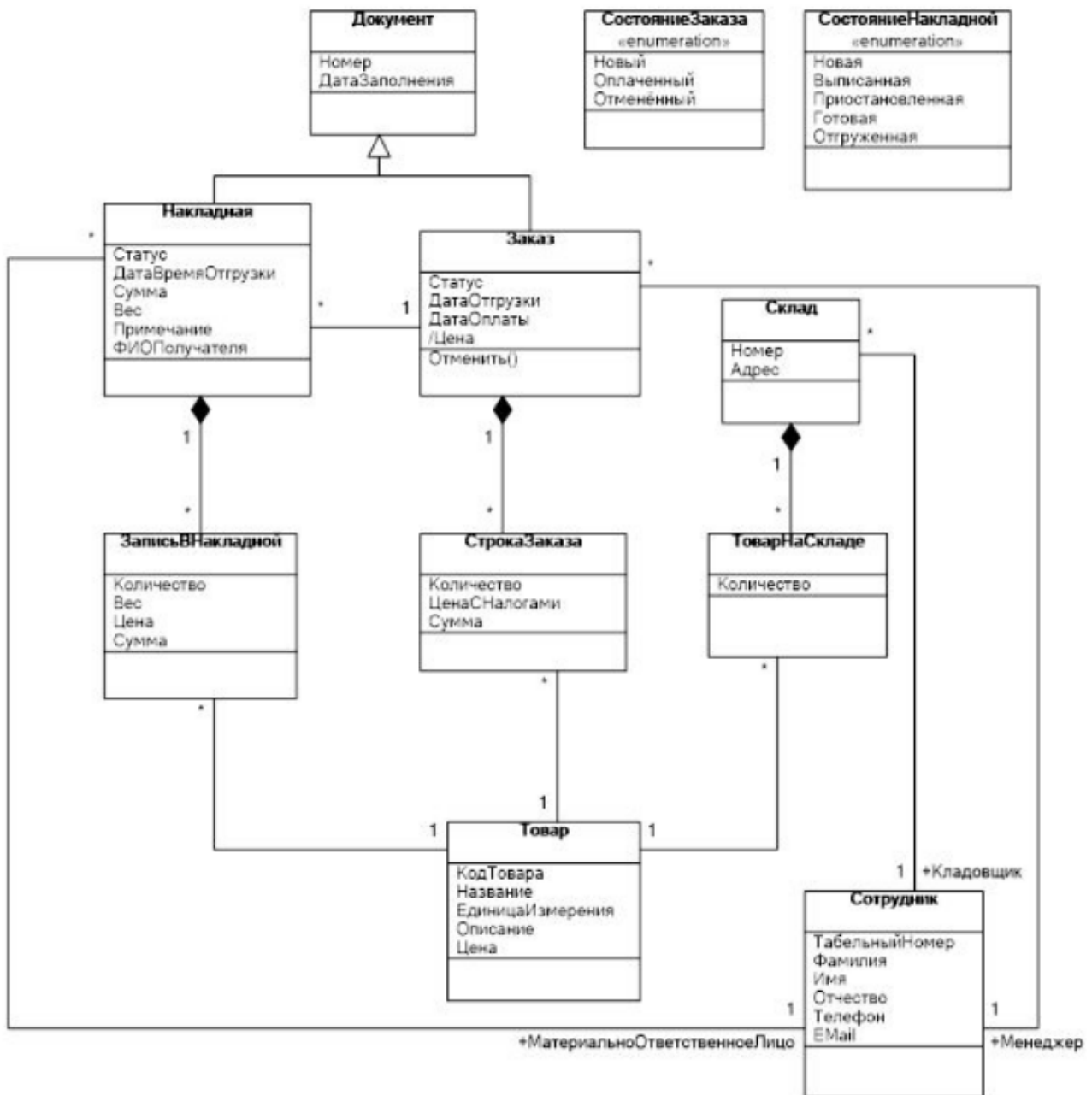


Рисунок 6 – Диаграмма классов

Основной сущностью в системе будет являться товар. Как известно из задания на проектирование, товар хранится на складе. Но понятия товара как некоего описания и товара, лежащего непосредственно на складе, отличаются друг от друга. Товар, лежащий на складе, кроме того, что связан со складом отношением композиции (агрегация не совсем подходит, поскольку в данной системе товар является товаром, пока он не покинет склад), ещё характеризуется количеством.

Аналогично следует рассуждать и при рассмотрении отношения Товара и Заказа, Товара и Накладной. В связи с тем, что Заказ и Накладная в сущности являются документами и имеют сходные атрибуты, они были объединены с помощью общего класса-предка Документ. Примечательно, что на диаграмме

представлены два класса со стереотипом Enumeration (перечисление). Стереотип можно установить из контекстного меню для класса.

Дополнительная информация:

<https://habr.com/ru/post/150041/>

<https://www.intuit.ru/studies/courses/1007/229/lecture/5956?page=1>

Выполнение работы

1. Практическая работа выполняется индивидуально.
2. Для выполнения практической работы рекомендуется использовать графический редактор Draw (<https://app.diagrams.net/>) или любой другой графический редактор, поддерживающий элементы UML.
3. Необходимо выбрать вариант задания для выполнения в рамках практической работы.

Предлагаемые варианты:

- a. Диаграмма классов для CRM-системы для SMM-агентства;
- b. Диаграмма классов для CRM-системы компании-производителя одежды;
- c. Диаграмма классов для интернет-магазина крупного производителя игрушек;
- d. Диаграмма классов для youtube-like видеосервиса на заказ: мобильное приложение + веб-приложение + админ панель;
- e. Диаграмма классов для системы бронирования переговорных комнат: клиентское приложение + админ панель для офиса сети продуктовых магазинов;
- f. Диаграмма классов для корпоративной социальной сети для нефтедобывающей компании;
- g. Диаграмма классов для сервиса по продаже авиа и ж/д билетов.

4. Выбранную систему необходимо декомпозировать на составляющие элементы.

Для декомпозиции рекомендуется использовать технологию майндмеппинга и программные средства XMind или Mindmeister. Дополнительную информацию про использование технологии майндмеппинга можно найти здесь:

<https://habr.com/ru/company/devexpress/blog/291028/>.

5. По результатам декомпозиции необходимо построить диаграмму классов для системы. Теоретическую информацию по построению диаграммы классов и пример диаграммы классов можно найти выше.

6. Результаты работы необходимо оформить в отчет в формате .doc или .pdf. Отчет необходимо оформить по ГОСТам (вспомните свою бакалаврскую работу и все получится). В отчете необходимо описать процесс и результат декомпозиции системы и создания диаграммы классов. Отчет необходимо прикрепить к заданию “Практическая работа N 2” в LMS (<https://lms.sfedu.ru/>).

ПРАКТИЧЕСКАЯ РАБОТА № 3

Планирование ресурсов проекта, формирование графиков выполнения работ

Цель: освоить базовые методы оценки продолжительности работ и составления графиков выполнения работ.

Выполнение работы

1. Лабораторная работа выполняется индивидуально.
2. Необходимо выбрать вариант задания для выполнения в рамках лабораторной работы.

Предлагаемые варианты

- a. CRM-система для SMM-агентства;
- b. CRM-система компании-производителя одежды;
- c. Интернет-магазин крупного производителя игрушек;
- d. Youtube-like видеосервис: мобильное приложение + веб-приложение + админ панель;
- e. Система бронирования переговорных комнат: клиентское приложение + админ панель для офиса сети продуктовых магазинов;
- f. Корпоративная социальная сеть для нефтедобывающей компании;
- g. Сервис по продаже авиа и ж/д билетов.

Возможно также использовать свою дипломную работу или любой другой ИТ-проект, который вы знаете.

Если вы выбираете собственный вариант проекта, его необходимо согласовать с преподавателем.

3. Разработку выбранной систему необходимо разбить на задачи двух уровней:

- a. крупные задачи / блоки работы
 - b. подзадачи – составляющие крупных задач.
- Суммарно подзадач должно быть не менее 20.

4. Выделенные задачи необходимо оформить в виде таблицы. Для оформления таблицы рекомендуется использовать MS Excel или Google Таблицы. В таблице необходимо указать зависимости задач. По трехточечной оценке необходимо рассчитать длительность каждой работы E. Значения оптимистичного (O), наиболее вероятного (M) и пессимистичного (P) времени вы можете взять исходя из своего опыта или из опыта ваших знакомых разработчиков, дизайнеров и др. специалистов. T – это время на тестирование, $T \approx 0,2E$.

Таблица 1 – Пример расчета

№	Задача	Предшествующие задачи	О, ч	М, ч	Р, ч	Е, ч	Т, ч	Е+Т, д
1	Backend разработка							
1.1	Регистрация пользователя		6	8	10	9,7	2,1	1,6
1.2	Авторизация пользователя	1.1	5	7	9	7	1,4	1

5. Полученную таблицу необходимо использовать для построения диаграммы Ганта. Для выполнения этой задачи рекомендуется использовать приложения для управления проектами ProjectLibre (можно скачать с официального сайта <http://www.projectlibre.com/product/projectlibre-open-source>) или MS Project (доступ по ссылке <https://aka.ms/devtoolsforteaching> с авторизацией по логину/паролю @sfedu.ru).

6. Результаты работы необходимо оформить в отчет в формате .doc или .pdf. Отчет необходимо оформить по ГОСТам (вспомните свою бакалаврскую работу и все получится). В отчет необходимо прикрепить полученную таблицу со списком задач и оценкой по времени и диаграмму Ганта. Отчет необходимо прикрепить к заданию “Практическая работа No3” в LMS (<https://lms.sfedu.ru/>). Вместе с текстовым файлом отчета в LMS необходимо также прикрепить файл таблицы в формате .xlsx (MS Excel) и файл проекта с построенной диаграммой Ганта.

ПРАКТИЧЕСКАЯ РАБОТА N 4

Управление ресурсами проекта

Цель: изучить принципы управления ресурсами проекта на основе построенных диаграмм Ганта, полученных в предыдущей работе.

Любой проект для своей реализации требует ресурсов: люди, оборудование, материалы. Управление проектом будет более эффективным, если каждому виду работ назначить необходимые ему ресурсы, использование которых позволит планировать проект более точно.

Предлагаемые варианты

- a. CRM-система для SMM-агентства;
- b. CRM-система компании-производителя одежды;
- c. Интернет-магазин крупного производителя игрушек;
- d. Youtube-like видеосервис: мобильное приложение + веб-приложение + админ панель;
- e. Система бронирования переговорных комнат: клиентское приложение + админ панель для офиса сети продуктовых магазинов;
- f. Корпоративная социальная сеть для нефтедобывающей компании;
- g. Сервис по продаже авиа и ж/д билетов.

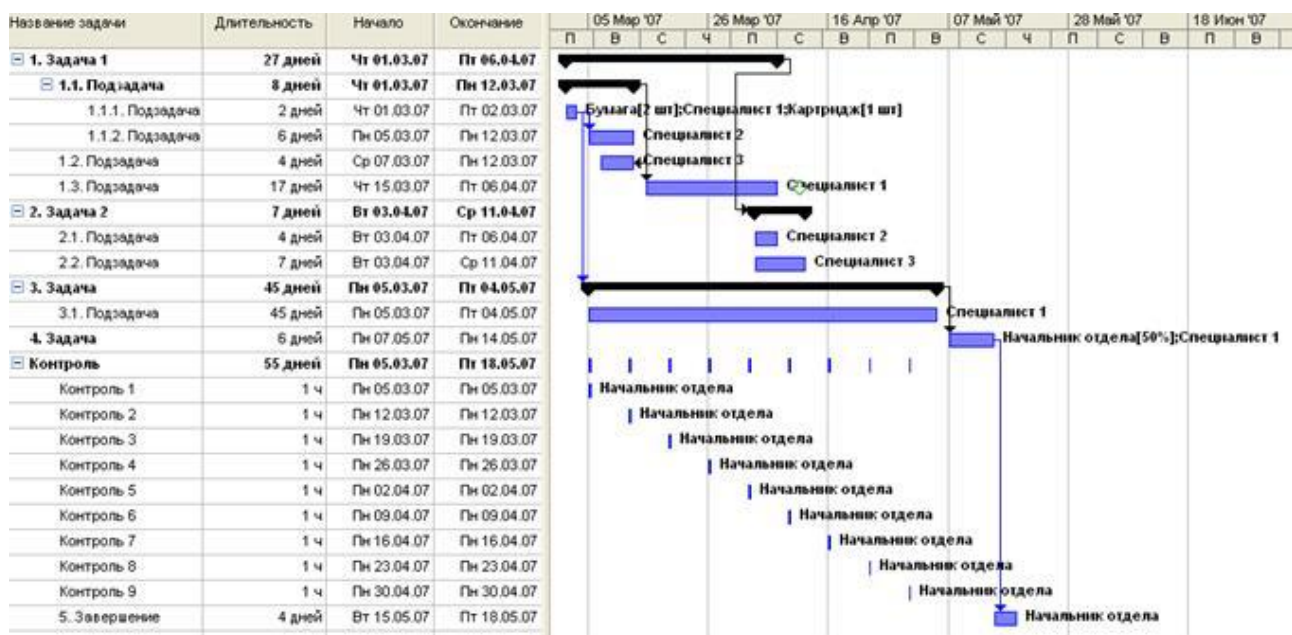
1. По выбранному варианту задания, сначала следует создать список ресурсов проекта. Для этого необходимо с использованием программных средств создать записи о трудовых и материальных ресурсах проекта.

2. Далее следует назначить ресурсы проекта определенным работам.

3. После ввода всех ресурсов следует определить рабочие часы всем трудовым ресурсам.

При назначении ресурсов следует помнить о том, что мы составляем начальный «срез» проекта и эти цифры могут меняться по ходу работы.

В начале работы над проектом мы формируем стартовый комплект работ и стартовое «положение» для планируемых ресурсов.



4. В ходе выполнения задания требуется классифицировать работы по следующим критериям:

- а. длительность.
- б. трудозатраты.
- с. количество человеческих ресурсов.

Данные параметры связаны друг с другом: трудозатраты задачи равны произведению длительности на количество людских ресурсов.

Задачи в плане проекта могут быть трех типов: с фиксированной длительностью, трудозатратами и количеством ресурсов.

5. После распределения необходимо разделить задачи на типы, попробовать поменять тип задачи и посмотреть как от этого изменится картина проекта.

6. Создать график ресурсов. Графическое представление ресурсов обеспечивает в удобной форме анализ использования людей и оборудования в проекте. График ресурсов позволяет отображать загрузку ресурсов, их стоимость и выполняемую работу.

2	⊖	Специалист 1	516 ч
		1.1.1. Подзадача	16 ч
		1.3. Подзадача	136 ч
		3.1. Подзадача	360 ч
		4. Задача	4 ч
3	⊖	Специалист 2	80 ч
		1.1.2. Подзадача	48 ч
		2.1. Подзадача	32 ч
4	⊖	Специалист 3	88 ч
		1.2. Подзадача	32 ч
		2.2. Подзадача	56 ч
5	⊖	Бумага	3 шт
		1. Задача 1	1 шт
		1.1.1. Подзадача	2 шт
6	⊖	Картридж	1 шт
		1.1.1. Подзадача	1 шт
7		Специалист 4	0 ч

Любой проект для своей реализации требует ресурсов: люди, оборудование, материалы. Управление проектом будет более эффективным, если каждому виду работ назначить необходимые ему ресурсы, использование которых позволит планировать проект более точно.

Название ресурса	Тип	Крат. назв	Единицы измерения материалов
Начальник отдела	Трудовой	Н	
Специалист 1	Трудовой	С1	
Специалист 2	Трудовой	С2	
Специалист 3	Трудовой	С3	
Бумага	Материальный	Б	шт
Картридж	Материальный	К	шт

7. Необходимо создать список ресурсов проекта и определить рабочие часы трудовым ресурсам (например, распределить задачи на различных специалистов).

8. В завершении работы требуется сформировать отчет о проекте.

Отчет необходимо прикрепить к заданию “Практическая работа N 4” в LMS (<https://lms.sfedu.ru/>).

ПРАКТИЧЕСКАЯ РАБОТА № 5

Анализ проекта

Цель: изучить принципы использования карт пользовательских историй, освоить методику построения карт пользовательских историй для инициации работ по проекту и составление графиков для анализа хода выполнения задач.

Краткие теоретические сведения

Карта пользовательских историй (user story map) – это инструмент, позволяющий грамотно пользовательских историй разбить продукт показывает на приоритет поставки.

Карта пользовательских историй, а также как они связаны между собой и с более крупными задачами пользователей. Карта пользовательских историй помогает команде, работающей над проектом, понять, как можно скомпоновать истории, чтобы получить продукт, готовый к релизу.

Карта пользовательских историй обычно составляется с помощью доски и стикеров. Для составления карты пользовательских историй необходимо определить роль пользователя, для которого составляется карта, и цель, которую он будет преследовать, пользуясь разрабатываемым программным продуктом.



Рисунок 1 – Роль и цель пользователя на карте пользовательских историй

Далее необходимо выделить ряд последовательных действий, которые необходимо совершить пользователю для достижения цели.



Рисунок 2 – Действия пользователя на карте пользовательских историй

После этого под каждым действием необходимо разместить варианты реализации этого действия в разрабатываемом продукте (задачи).

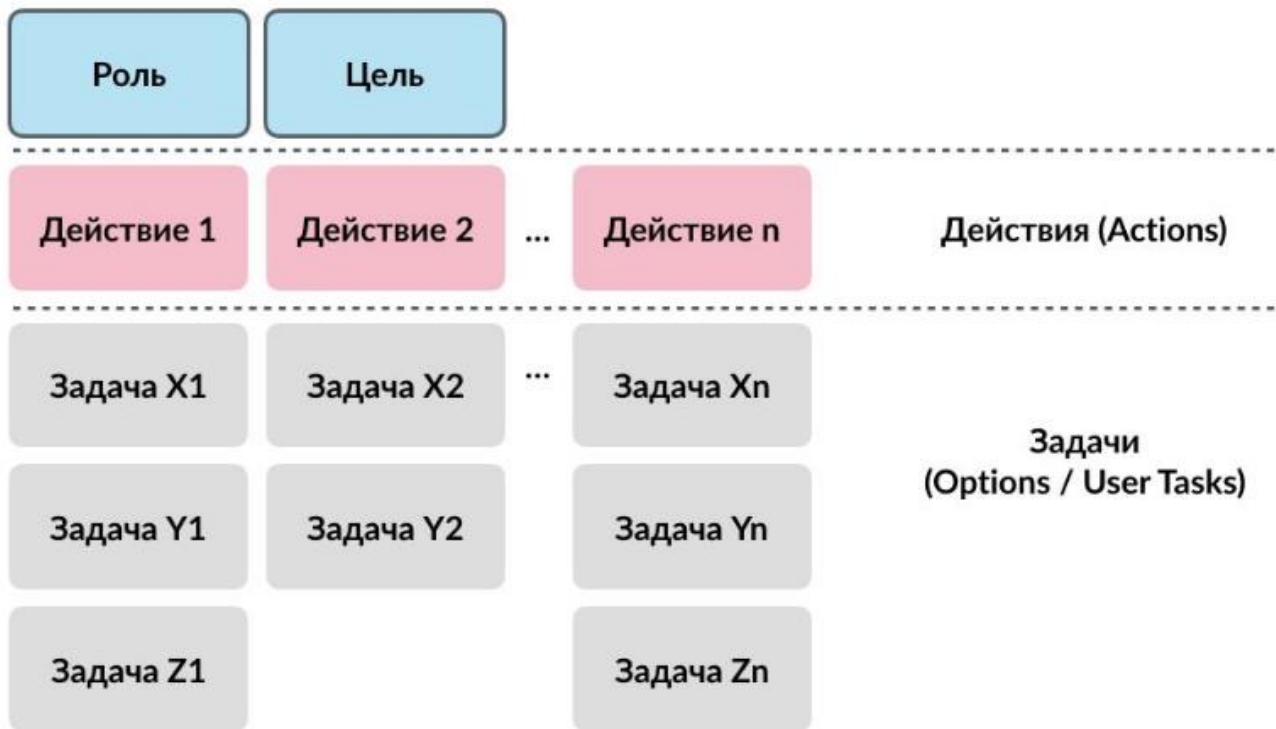


Рисунок 3 – Задачи на карте пользовательских историй

Следующий шаг – сортировка задач по сложности реализации и определение релизов по результатам сортировки. Продукт, получаемый по результатам релиза должен так или иначе позволять пользователю достичь его цели в определенной роли.

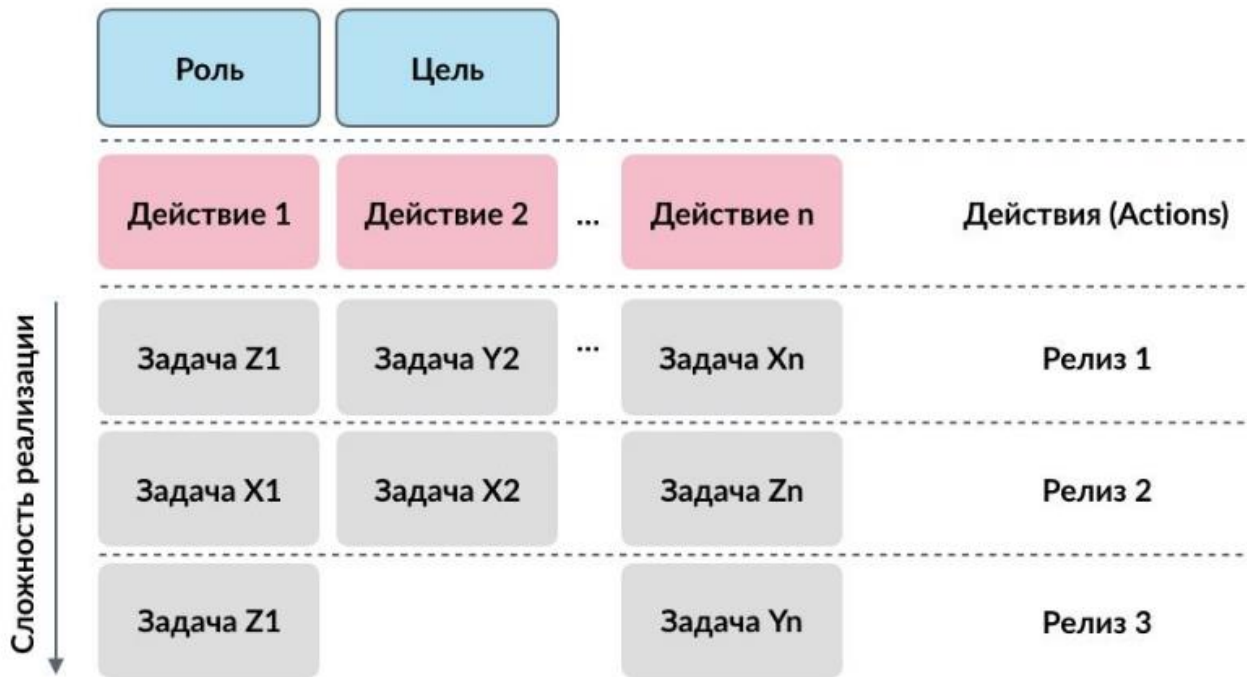


Рисунок 4 – Выделение релизов на карте пользовательских историй

На рисунке 5 показан пример полученной карты пользовательских историй для пассажира в приложении вызова такси.

Роль: пассажир	Цель: доехать из точки А в точку В
-------------------	---

Заказать такси	Выйти на улицу, когда машина приехала	Опреде- лить машину	Назвать адрес В	Доехать в точку В	Оплатить поездку
-------------------	---	---------------------------	--------------------	----------------------	---------------------

Ввести текстом адреса А и В	Получить звонок/ смс от водителя о том, что машина приехала	Позвонить водителю		Видеть оставшее- ся время в поездке	Оплатить наличны- ми	Релиз 1
Выбрать адреса из списка	Получить оповеще- ние о том, что машина приехала	Посмот- реть марку, цвет и номер машины			Оплатить картой	Релиз 2
Опреде- лить адрес А с помощью геолока- ции	Посмот- реть положение машины на карте		Изменить маршрут	Видеть маршрут на карте		Релиз 3
Отметить точки А и В на карте						Релиз 4
				Играть в игру		Релиз 5

Рисунок 5 – Пример карты пользовательских историй

Выполнение работы

1. Лабораторная работа выполняется индивидуально.
2. Для выполнения лабораторной работы рекомендуется использовать приложение Google Jamboard (<https://jamboard.google.com/>, авторизация с помощью аккаунта Google) или любой другой графический/текстовый редактор, поддерживающий элементы карты пользовательских историй.

3. Необходимо выбрать вариант задания для выполнения в рамках лабораторной работы. Предлагаемые варианты:

- a. CRM-система для SMM-агентства;
- b. CRM-система компании-производителя одежды;
- c. Интернет-магазин крупного производителя игрушек;
- d. Youtube-like видеосервис: мобильное приложение + веб-приложение + админ панель;
- e. Система бронирования переговорных комнат: клиентское приложение + админ панель для офиса сети продуктовых магазинов;
- f. Корпоративная социальная сеть для нефтедобывающей компании;
- g. Сервис по продаже авиа и ж/д билетов.

Возможно также использовать свою дипломную работу или любой другой ИТ-проект, который вы знаете.

Если вы выбираете собственный вариант проекта, его необходимо согласовать с преподавателем.

4. Для выбранной системы необходимо определить минимум две роли пользователей. Для каждой роли пользователей необходимо определить минимум две цели, которые будут достигаться с помощью использования разрабатываемого программного продукта.

5. Необходимо составить две карты пользовательских историй для двух любых выявленных ролей пользователей. Для карты нужно выбрать любую из определенных в предыдущем пункте целей пользователя.

6. Составленную карту пользовательских историй необходимо разбить на релизы. Для успешного выполнения лабораторной работы на карте должно получиться 4-6 релизов.

7. Синхронизировать карту пользовательских историй и scrum-доску проекта.

Рассмотрим специфические термины современного подхода к управлению ИТ-проектами, которые помогут нам в выполнении работы.

Спринт — итерация в скрам, в ходе которой создаётся функциональный рост программного обеспечения. Жёстко фиксирован по времени. Длительность одного спринта от 2 до 4 недель. В отдельных случаях, к примеру согласно Scrum стандарту Nokia, длительность спринта должна быть не более 6 недель. Тем не менее, считается, что чем короче спринт, тем более гибким является процесс разработки, релизы выходят чаще, быстрее поступают отзывы от потребителя, меньше времени тратится на работу в неправильном направлении. С другой стороны, при более длительных спринтах команда имеет больше времени на решение возникших в процессе проблем, а владелец проекта уменьшает издержки на совещания, демонстрации продукта и т. п. Разные

команды подбирают длину спринта согласно специфике своей работы, составу команд и требований, часто методом проб и ошибок. Для оценки объема работ в спринте можно использовать предварительную оценку, измеряемую в очках истории. Предварительная оценка фиксируется в резерве проекта. На протяжении спринта никто не имеет права менять список требований к работе, внесенном в резерв проекта.

Резерв проекта (Product backlog) — это список требований к функциональности, упорядоченный по их степени важности, подлежащих реализации. Элементы этого списка называются «пожеланиями пользователя» (user story) или элементами резерва (backlog items). Резерв проекта открыт для редактирования для всех участников скрам процесса.

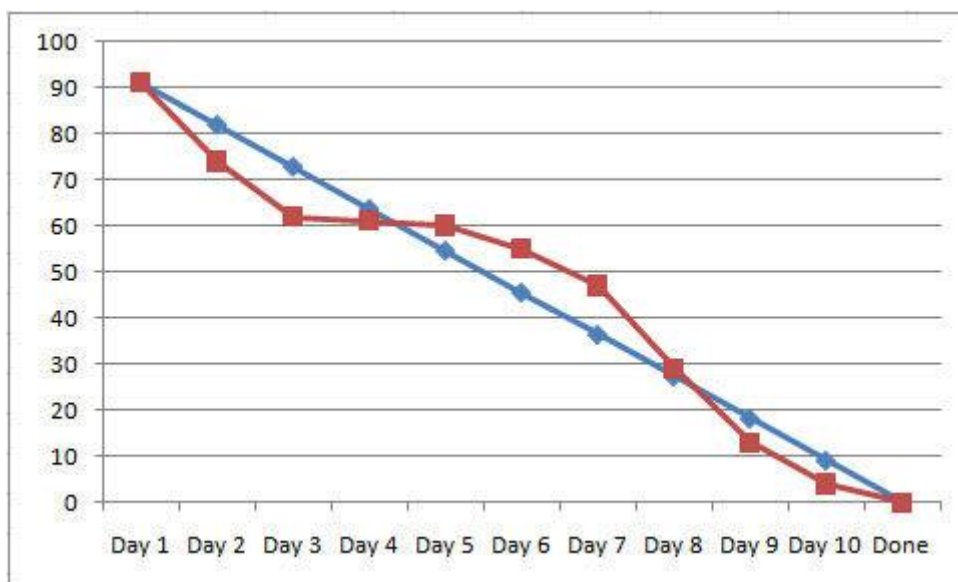
Резерв спринта (Sprint backlog) — содержит функциональность, выбранную владельцем проекта из резерва проекта. Все функции разбиты по задачам, каждая из которых оценивается скрам-командой. Каждый день команда оценивает объем работы, который нужно проделать для завершения спринта.

Пожелание заказчика (User Story) — требуемая функциональность, которую добавляют в резерв часто называют историей. Зачастую история имеет следующую структуру: «Будучи пользователем <тип пользователя> я хочу сделать <действие>, чтобы получить <результат>». Такая структура удобна тем, что понятна как разработчикам так и заказчикам.

Очки истории (Story Points) — абстрактная метрика оценки сложности истории, которая не учитывает затраты в человекочасах. Обычно используют одну из следующих шкал: ряд Фибоначчи (1,2,3,5,8,13,21,34,55); линейную шкалу (1,2,3,4 ... n); степень двойки (1,2,4,8 ... 2n); размеры одежды (XS, S, M, L, XL).

График сгорания задач (Burndown chart) — Диаграмма, показывающая количество сделанной и оставшейся работы. Обновляется ежедневно с тем, чтобы в простой форме показать подвижки в работе над спринтом. График должен быть общедоступен. Существуют разные виды диаграммы:

- диаграмма сгорания работ для спринта — показывает, сколько уже задач сделано и сколько ещё остаётся сделать в текущем спринте.
- диаграмма сгорания работ для выпуска проекта — показывает, сколько уже задач сделано и сколько ещё остаётся сделать до выпуска продукта (обычно строится на базе нескольких спринтов).



Соответственно, для выполнения этой части задания нам необходимо сформировать спринты из задач, выделенных на предыдущих этапах работы, определить состав спринтов и сформировать scrum-доску.

Сымитировать движение задач по доске в течении спринта.

После расположения задач на доске, требуется сформировать отчеты в виде диаграмм сгорания и выполнить их анализ, то есть определить, насколько эффективно решаются задачи.

8. Результаты работы необходимо оформить в отчет в формате .doc или .pdf. Отчет необходимо оформить по ГОСТам (вспомните свою бакалаврскую работу и все получится). В отчет необходимо:

a. ответить своими словами и мыслями на вопрос “Когда и для чего необходимо составлять карты пользовательских историй?”;

b. описать процесс создания своих карт.

Важно : копировать и вставлять теорию из сети Интернет не нужно, требуется описать процесс создания конкретных ваших карт, объяснить выбор ролей пользователей, их целей, действий и задач;

c. оформить и прикрепить полученные карты.

Важно: полученные карты необходимо оформить в отчете так, чтобы можно было прочитать текст на стикерах.

Отчет необходимо прикрепить к заданию “Практическая работа N 5” в LMS (<https://lms.sfedu.ru/>).

ПРАКТИЧЕСКАЯ РАБОТА № 6

Отслеживание исполнения проекта

Цель: изучить механику создания Scrum-досок, изучить принципы работы со Scrum-досками, освоить базовые принципы работы с цифровыми инструментами, имитирующими Scrum-доски.

Краткие теоретические сведения

Scrum-board (task board, доска задач, Scrum-доска или скрам-борд) – это пространство (физическое или цифровое) разделенное на колонки со следующими значениями:

- Backlog/Бэклог – что предстоит сделать;
- To Do/Ещё не начато – что нужно сделать в заданный промежуток времени - спринт;
- In Progress/В работе – что уже кем-либо делается в текущем спринте;
- Returned/Возвращённые – то, что не прошло тестирование;
- Testing/Для тестирования – то, что нужно проверить или протестировать;
- Done/Готово – то, что уже выполнено.

В этих колонках принято располагать стикеры/карточки (или их подобие) с заданиями, делами или таскам. Изначально все стикеры попадают в первую колонку, затем, после планирования спринта или итерации, часть стикеров переходит во вторую колонку. Как только кто-то начинает делать задачу, стикер переносится в колонку “В работе”.

Колонка проверки является необязательной, потому что не всегда есть необходимость в подобной деятельности. Целиком выполненные таски попадают в последний столбик и являются гарантией того, что обращаться к ним больше не понадобится.

Scrum-доска – это динамическая структура, где постоянно происходит движение элементов, поэтому ее нельзя запечатлеть на фотографии или просто нарисовать на листке бумаги. Доски существуют как в физическом формате в виде пластиковых или деревянных досок со стикерами, так и в цифровом формате, например Trello, MS Planner, Jira и др.

Scrum-доска может понадобиться в следующих случаях:

- Для ведения IT-проектов – организация слаженной работы команды, планирование и приоритизация задач, обеспечение прозрачности прогресса для заказчика или руководства.
- Для ведения других проектов, например, ремонт, подготовка к мероприятию и т.д.
- В сфере образования, например, список всех тем за семестр – Бэклог, темы на следующий раз и домашнее задание – Ещё не начато, пройденные темы –

Готово. Таким образом студенты и преподаватели знают что делать, видят прикрепленные документы, понимают приоритеты заданий.

- Для планирования личных задач – расписание дня в семье, планирование путешествия, составление личного расписания и др.

Выполнение работы

1. Лабораторная работа выполняется индивидуально.

2. Для выполнения лабораторной работы рекомендуется использовать приложение Trello (<https://trello.com/>). Если у вас нет аккаунта, зарегистрируйтесь в сервисе Trello.

3. Для выполнения текущей лабораторной работы необходимо использовать результаты лабораторной работы №4 – пользовательские истории любой из составленных карт пользовательских историй. 4. В Trello необходимо создать доску “Проект <название проекта>”. На доске должно быть 6 колонок:

- a. Бэклог;
- b. Ещё не начато;
- c. В работе;
- d. Возвращённые;
- e. Для тестирования;
- f. Готово.

5. Колонку “Бэклог” необходимо заполнить карточками в формате пользовательских историй.

Для этого необходимо использовать пользовательские истории любой карты пользовательских историй.

6. В колонку “Ещё не начато” необходимо перенести карточки из колонки “Бэклог” – пользовательские истории, определенные для первого релиза на карте пользовательских историй – это пользовательские истории, которые будут реализованы в первом спринте.

7. На доске необходимо разместить цель спринта. Цель спринта должна соотноситься с пользовательскими историями, выбранными для реализации в первом спринте. Карточку с целью спринта необходимо выделить с помощью элементов интерфейса Trello.

8. Каждую пользовательскую историю в колонке “Ещё не начато” необходимо разбить на задачи: это могут быть, например, разработка бэкенда для пользовательской истории, разработка фронтенда, разработка дизайна и т.п. По карточке необходимо понять, какой у нее тип (пользовательская история, задача) и к какой пользовательской истории она относится – это можно реализовать с помощью тегов, цветных меток и других элементов интерфейса.

9. Распределите карточки по доске следующим образом:

- a. Пользовательская история 1 завершена полностью;
 - b. Пользовательская история 2: завершена на 40-50%, остальные 50-60% пользовательской истории в тестировании;
 - c. Пользовательская история 3: в работе на 20-30%, одна из задач уже протестирована, но тестировщик не удовлетворен работой программиста, оставшаяся часть пользовательской истории ещё не начата;
 - d. остальные пользовательские истории ещё не начаты.
10. Результаты работы необходимо оформить в отчет в формате .doc или .pdf.

Отчет необходимо оформить по ГОСТам. В отчете необходимо описать все этапы работы и прикрепить скриншоты, иллюстрирующие каждый этап работы. В отчете необходимо разместить ссылку на созданную доску, доступ к доске необходимо предоставить преподавателю. Отчет необходимо прикрепить к заданию “Практическая работа N 6” в LMS (<https://lms.sfedu.ru/>).

ПРАКТИЧЕСКАЯ РАБОТА № 7

Составление отчетности по проекту

Цель: научиться формировать отчетность по проекту на основе проведенных работ, как это выполняется при завершении работ.

Отчёты о статусе проекта не относятся к мониторингу в реальном времени. Эти отчёты представляют собой обобщение сведений, полученных за прошлую неделю, последние две недели или прошедший месяц работ по проекту. Для заинтересованных лиц это возможность узнать, насколько хорошо вы соблюдаете план проекта.

1. Провести анализ выполненных работ.

2. Сформировать отчет по проекту на основе следующего алгоритма:

- a. Подготовить формы отчета по ходу работы на основе отслеживания информации в инструментах управления проектом.
- b. Сформировать заголовки и объективные названия элементам отчета.
- c. Обозначить состояние проекта.
- d. Сформировать содержание отчета.
- e. Сформировать общий обзор для каждой ключевой области работы.
- f. Добавить ссылки на другие документы или ресурсы, которые используются в проекте.
- g. Описать все препятствия, которые были обойдены в ходе ведения проекта.
- h. Описать следующие шаги по проекту.

3. Отчет о статусе проекта можно формировать по следующей структуре:

Название отчёта:

Напишите здесь название отчёта. Для простоты можно указать название проекта и дату составления отчёта.

Состояние проекта:

В проекте всё по плану, он под угрозой или отстаёт?

Сводка:

Добавьте сюда краткое описание наиболее важных выводов из отчёта о статусе проекта. Учитывайте, что занятые заинтересованные лица, возможно, будут просматривать только этот раздел, поэтому включите в него все основные моменты или препятствия, о которых необходимо знать всему коллективу.

Название отчёта:

Напишите здесь название отчёта. Для простоты можно указать название проекта и дату составления отчёта.

Состояние проекта:

В проекте всё по плану, он под угрозой или отстаёт?

Сводка:

Добавьте сюда краткое описание наиболее важных выводов из отчёта о статусе проекта. Учитывайте, что занятые заинтересованные лица, возможно, будут просматривать только этот раздел, поэтому включите в него все основные моменты или препятствия, о которых необходимо знать всему коллективу.

Отчет необходимо оформить по ГОСТам. В отчете необходимо описать все этапы работы и прикрепить скриншоты, иллюстрирующие каждый этап работы. В отчете необходимо разместить ссылку на созданную доску, доступ к доске необходимо предоставить преподавателю. Отчет необходимо прикрепить к заданию “Практическая работа N 7” в LMS (<https://lms.sfedu.ru/>).