



РАБОЧАЯ ПРОГРАММА
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ
/вид практики/

Преддипломная практика
/тип практики/

Направление подготовки:
38.03.05 Бизнес-информатика

Направленность (профиль) подготовки:
Цифровая экономика

уровень высшего образования
Бакалавриат

2022 г.

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	3
1.1 Вид и типы, способ и формы ее проведения	3
1.2 Цель, задачи и нормативно-правовая база практики	3
1.3 Перечень планируемых результатов обучения при прохождении Практики, соотнесенных с планируемыми результатами освоения образовательной программы	4
1.4 Место Практики в структуре образовательной программы	5
1.5 Объем Практики в зачетных единицах и ее продолжительности в неделях	5
2. СОДЕРЖАНИЕ ПРАКТИКИ	7
2.1 Этапы практики и виды выполняемых работ	7
2.2 Индивидуальное задание на практику	7
2.3 Особенности прохождения практики инвалидами и лицами с ограниченными возможностями здоровья	7
3. ФОРМЫ ОТЧЕТНОСТИ ПО РЕЗУЛЬТАТАМ ПРАКТИКИ	8
4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ПРАКТИКЕ	11
5. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ И РЕСУРСОВ СЕТИ ИНТЕРНЕТ, НЕОБХОДИМЫХ ДЛЯ ПРОВЕДЕНИЯ ПРАКТИКИ	11
6. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ПРОВЕДЕНИИ ПРАКТИКИ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ	13
7. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ПРОХОЖДЕНИЯ ПРАКТИКИ	13
8. СВЕДЕНИЯ ОБ УТВЕРЖДЕНИИ ПРОГРАММЫ ПРАКТИКИ И ВНЕСЕНИИ ИЗМЕНЕНИЙ	13
9. ПРИЛОЖЕНИЯ	14
Приложение 1	14
Приложение 2.1	15
Приложение 2.2	33
Приложение 3	51
Приложение 4	54
Приложение 5	60

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Вид и типы, способ и формы ее проведения

В соответствии с ФГОС ВО, видом практики является – производственная практика (далее – Практика).

Типом данного вида практики является: *Преддипломная практика*

Аттестация по итогам практики осуществляется на основе достижения обучающимися планируемых результатов обучения при прохождении Практики.

Способ проведения практики – *стационарный и выездной*.

Форма проведения Практики – *дискретно по видам практик*.

1.2 Цель, задачи и нормативно-правовая база практики

Цели, задачи и содержание Практики определяются требованиями к результатам Практики, установленными ФГОС ВО в части формирования профессиональных компетенций.

Цель: получение студентами практического опыта использования информационных технологий для решения прикладных задач, а также на выработку практических навыков по их анализу, выбору и применению информационных технологий.

Задачи:

- Закрепление приобретенных теоретических знаний.
- Изучение опыта создания и применения информационных технологий в структурных подразделениях организации
- Использование нормативно-правовой документации при решении прикладных задач
- Составление требований к информационной системе при анализе заявок от пользователей информационной системы в организации
- Разработка и адаптации модулей прикладного программного обеспечения ИТ-инфраструктуры в организации
- Составление технического задания на создание компонентов автоматизированной системы
- Анализ собранной информации от пользователей и формирования требований к информационно-коммуникационной среде в организации
- Формулирование предложений по автоматизации бизнес-процессов
- Приобретение навыков практического решения задач на конкретных рабочих местах в качестве исполнителей или стажеров.
- Получение навыков самостоятельной работы
- Получение навыков подготовки отчетной документации

Нормативную правовую базу разработки программы Практики составляют:

- Федеральный закон РФ от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 38.03.05 Бизнес-информатика (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации № 838 от 29.07.2020;
- Приказ Минобрнауки России от 06.04.2021 г. № 245 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры»;
- Приказ Минобрнауки России N 885, Минпросвещения России N 390 от 05.08.2020 "О практической подготовке обучающихся"(вместе с "Положением о практической подготовке обучающихся");
- Локальные нормативные акты Университета.

1.3 Перечень планируемых результатов обучения при прохождении Практики, соотнесенных с планируемыми результатами освоения образовательной программы

В результате прохождения практики у обучающихся должны быть сформированы компетенции в соответствии с ФГОС ВО по направлению подготовки 38.03.05 Бизнес-информатика и с учетом обобщенной трудовой функции и трудовой функции профессионального стандарта 06.013 «Специалист по информационным ресурсам» и 06.015 «Специалист по информационным системам», к выполнению которых в ходе обучения готовится обучающийся.

Соотношение обобщённых трудовых функций (ОТФ) и трудовых функций, имеющих отношение к будущей профессиональной деятельности обучающегося (ТФ):

Наличие и соответствие ПК профессиональному стандарту

Код и наименование профессионального стандарта	Код и наименование ОТФ	Код и наименование ТФ
06.013 Специалист по информационным ресурсам	С6 Управление информационными ресурсами	С/01.6 Организация работ по созданию и редактированию контента сайтов
		С/03.6 Контроль за наполнением сайта
		С/04.6 Организация работ по изменению структуры сайта
		С/05.6 Анализ информационных потребностей посетителей сайта
		С/07.6 Поддержка процессов модернизации и продвижения сайта
06.015 Специалист по информационным системам	С6 Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	С/08.6 Разработка модели бизнес-процессов заказчика в рамках проекта создания (модификации) ИС
		С/09.6 Адаптация бизнес-процессов заказчика ИС к возможностям ИС в рамках проекта создания (модификации) ИС
		С/16.6 Проектирование и дизайн ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС
		С/40.6 Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию

Процесс прохождения практики направлен на формирование у обучающихся следующих компетенций (результатов освоения образовательной программы):

Коды компетенций	Содержание компетенции
ПК-1	Способность создавать, редактировать и наполнять контентом информационные ресурсы
ПК-2	Способность разработки прикладного программного обеспечения для управления бизнесом
ПК-3	Способность выполнять анализ бизнес-информации
ПК-4	Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами
ПК-7	Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования

Планируемые результаты практики, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование профессиональной компетенции выпускника программы бакалавриата	Код и наименование индикатора достижения профессиональной компетенции
---	---

ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы	ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента.
ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом	ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владет навыками автоматизации решения типовых задач для управления бизнесом.
ПК-3. Способность выполнять анализ бизнес-информации	ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2. Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владет методами извлечения, анализа и обработки информации.
ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владет навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
ПК-7. Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования	ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владет методами разработки дизайна ИТ сервисов

1.4 Место Практики в структуре образовательной программы

Практика является обязательным видом учебной работы обучающихся, входит в Блок 2 Практика в соответствии с ФГОС ВО по направлению подготовки 38.03.05 Бизнес-информатика, Преддипломная практика относится к части ОПОП ВО, формируемой участниками образовательных отношений.

1.5 Объем Практики в зачетных единицах и ее продолжительности в неделях

В соответствии с ОПОП по направлению 38.03.05 Бизнес-информатика, профиль «Цифровая экономика» на Практику отводится следующее количество зачетных единиц и времени:

Тип/вид практики	Трудоемкость в часах	Трудоемкость в зачетных единицах	Продолжительность в неделях
<i>Производственная практика: Преддипломная практика</i>	216	6	4

2. СОДЕРЖАНИЕ ПРАКТИКИ

2.1. Этапы практики и виды выполняемых работ

Практика является частью ОПОП ВО, формируемой участниками образовательных отношений и представляет собой вид учебных занятий, непосредственно ориентированных на практическую подготовку обучающихся в соответствии с видами деятельности, на которые направлена программа бакалавриата.

Практика осуществляется **в три этапа**:

- *Подготовительный этап* (проведение инструктивного совещания с приглашением руководителей практики, ознакомление обучающихся с содержанием и спецификой практики, доведение до сведения обучающегося содержания индивидуального задания на практику, видов отчетности по практике).

- *Организационно-аналитический этап реализуется как практическая подготовка* (участие обучающихся в различных видах профессиональной деятельности в соответствии с индивидуальным заданием).

- *Результативно-аналитический этап* (анализ проделанной работы и подведение её итогов, оформление отчета о практике и представление его руководителю).

Практика может быть организована на основе договоров с организациями, осуществляющих деятельность по профилю направления подготовки (далее - профильная организация) и может быть пройдена в:

- информационно-технологических, программно-обеспечивающих службах организаций различных отраслей, сфер и форм собственности;

- подразделения и учреждения, занимающихся информационной безопасностью, веб-программированием, созданием и обслуживанием визуально-коммуникативной среды, программного обеспечения и т.п.

Обучающиеся, совмещающие обучение с трудовой деятельностью, вправе проходить производственную практику по месту трудовой деятельности в случаях, если профессиональная деятельность, осуществляемая ими, соответствует требованиям образовательной программы к проведению практики.

Также производственная практика может проводиться в структурных подразделениях ЧОУВО «МУ им. С.Ю. Витте».

Обучающиеся направляются на практику на основании заявления (Приложение 1).

Обучающиеся в период прохождения практики:

- выполняют индивидуальные задания, предусмотренные программами практики;

- соблюдают правила внутреннего трудового распорядка;

- соблюдают требования охраны труда и пожарной безопасности

2.2. Индивидуальное задание на практику

В процессе прохождения практики обучающийся должен руководствоваться индивидуальным заданием. Индивидуальное задание может быть получено от руководителя практики от Университета или обучающийся может взять типовое задание из настоящей программы практики. При прохождении практики в профильной организации, индивидуальное задание должно быть согласовано с руководителем практики от профильной организации.

Индивидуальное задание представляет собой планирование работы обучающихся во время практики, направленной на формирование указанных компетенций, выполняемой во внеаудиторное время по заданию и при методическом руководстве преподавателя (Приложение 2).

2.3. Особенности прохождения практики инвалидами и лицами с ограниченными возможностями здоровья

Особенности прохождения практики инвалидами и лицами с ОВЗ определены в Порядке организации и проведения практики обучающихся, осваивающих основные

профессиональные образовательные программы высшего образования, утвержденным приказом ректора.

Практика для обучающихся с ограниченными возможностями здоровья и инвалидов проводится с учетом особенностей их психофизического развития, индивидуальных возможностей и состояния здоровья.

Для лиц с ограниченными возможностями здоровья и инвалидов, создаются специально оборудованные рабочие места с учетом их особенностей, физиологии, а также психофизического развития, индивидуальных возможностей, состояния здоровья, профессионального вида деятельности, характера труда, выполняемых трудовых функций.

Материально-технические условия прохождения Практики, должны обеспечивать возможность беспрепятственного доступа практикантов из числа лиц с ограниченными возможностями здоровья и инвалидов к специально оборудованным рабочим местам, а также в туалетные комнаты и другие помещения, а также их пребывания в указанных помещениях Организации (в том числе наличие пандусов, подъемников, поручней, расширенных дверных проемов, лифтов; при отсутствии лифтов рабочее место должно располагаться на первом этаже здания).

Не допускается использование практиканта на должностях и работах, противопоказанных лицам с ограниченными возможностями и инвалидам.

3. ФОРМЫ ОТЧЕТНОСТИ ПО РЕЗУЛЬТАТАМ ПРАКТИКИ

Полный пакет отчетных документов по результатам практики включает:

- заявление на практику с подписью студента;
- договор на практику (если практика проводится на базе профильного предприятия) со всеми подписями и печатями;
- индивидуальное задание на практику, со всеми подписями;
- отчет о прохождении практики. **Не допускается выгрузка отчета внутри архивного файла!**

В том числе:

- **файл программного кода *.ру** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях), если работа выполнялась с использованием языка программирования Python
- **файл с выгрузкой конфигурации 1С *.dt** (возможно приложить архив всей папки с конфигурацией, с комментариями о версии платформы 1С, для которой выполнялся проект), если работа выполнялась с использованием платформы 1С
- **файлы веб-страниц в формате *.html, *.css, *.php, *.js** (при необходимости архивируется вся папка проекта, с комментариями об используемых версиях), если работа велась по разработке интернет-сайта
- в случае, если работа велась на программном обеспечении организации, в которой проходила преддипломная практика и предоставить исходные файлы невозможно, **скринкаст запись работы в программном обеспечении в форматах .mkv, .mp4, .avi**. При этом допускается применение эффекта замыливания конфиденциальных данных или использование тестовых данных («рыбы»)
- **другие исходные файлы и файлы запуска ПО**, если работа велась на иных языках программирования/средах разработки
- **файлы данных и дополнительные файлы.**

В случае отсутствия файлов программного кода/конфигурации отчет по практике не может быть проверен и оценен.

Объем заимствований и порядок использования для составления отчетной документации технологии искусственной генерации текста определяются «Положением об использовании системы «Антиплагиат» в образовательной и научной деятельности» Университета.

Выполнение задач по практике и оформление отчетной документации производится каждым студентом самостоятельно. Особое внимание обращается на самостоятельность выполнения программного кода. Самостоятельность выполнения работы проверяется руководителем практики с помощью системы «Антиплагиат» согласно Положению об использовании системы «Антиплагиат» в образовательной и научной деятельности, размещенному на официальном сайте университета. Не самостоятельно выполненная работа не может быть принята и не может оцениваться положительной оценкой.

При использовании обучающимся в учебной работе систем искусственной генерации текста и/или программного кода, и при полном соответствии требованиям Положения об использовании системы "Антиплагиат" в образовательной и научной деятельности, и настоящей программы практики максимально возможная оценка за работу не может превышать 51 балла.

Руководитель практики в "отзыве в виде комментария" ставит отметку об использовании обучающимся в своей работе систем искусственной генерации текста.

Не допускается неоднократная выгрузка работы выполненной несамостоятельно, не соответствующей требованиям настоящей программы практики и нормативных документов университета, а также по которой невозможно определить сформированность у обучающихся компетенций, указанных в разделе 1.3 настоящей программы практики. Попытки неоднократной выгрузки подобных работ могут быть заблокированы руководителем практики в Электронной университете.

По результатам Практики, в сроки, установленные Регламентом практической подготовки, обучающийся обязан загрузить в личном кабинете обучающегося вышеуказанные документы и программные файлы, свидетельствующие о закреплении знаний, умений, приобретении практического опыта, сформированности компетенций.

Порядок проверки и оценивания отчетной документации по практике регулируется Регламентом практической подготовки.

Обучающиеся, не выполнившие программу практики без уважительной причины или получившие по ее итогам неудовлетворительную оценку, и не ликвидировавшие возникшую академическую задолженность, подлежат отчислению из вуза в установленном порядке как имеющие академическую задолженность.

Отчет о прохождении Практики, составляется по утвержденной форме:

1. Содержание
2. Введение;
3. Основная часть;
4. Заключение;
5. Список литературы;
6. Приложения.

Пример структуры отчета по практике приведен в Приложении 3.

Основными требованиями, предъявляемыми к содержанию отчета по практике, являются:

Титульный лист отчета по практике генерируется из личного кабинет электронного университета при нажатии на ссылку со сроками практики. Титульный лист помимо основной информации содержит в правом верхнем углу персонализированный штрих-код, который является уникальным и позволяет идентифицировать как ФИО самого студента, так и всю сопроводительную документацию по практике. Титульный лист вставляется в качестве первого листа в отчет. Подписывать титульный лист не нужно. **Запрещается распознавать титульный лист и вставлять его в виде текста.**

Содержание формируется как Автособираемое оглавление из заголовков отчета.

Во введении указываются: цель, задачи, место прохождения практики, источники информации и данных, которые планировалось использовать при выполнения индивидуального задания практики.

В основной части отчета дается описание предметной области и основных результатов в период прохождения практики в соответствии с индивидуальным заданием, ведется разработка программного продукта, автоматизация бизнес-процессов. После каждого раздела приводятся промежуточные выводы в соответствии с выполненными этапами индивидуального задания, соотнесенными с формируемыми компетенциями. Основная часть содержит три раздела: анализ предприятия, составление технической документации, проектирование.

В первом разделе необходимо провести анализ организации, на базе которой проходит практика. Анализируется структура организации, документационное обеспечение ее деятельности. Проводится исследование ИТ-инфраструктуры. Формализуются требования пользователей. Формализуется перечень служебных поручений для прохождения практики. Уточняются требования к информационно-коммуникационной среде. Создается техническое задание на разработку, модификацию или развертывание составляющих компонентов информационно-коммуникационной среды.

Во втором разделе выполняется проектирование информационной системы или программного модуля. Проектируется архитектура информационной системы. Разрабатывается прототип и дизайн интерфейса. Создается структура базы данных. Выполняется написание программного кода и его верификация. Проводится тестирование и отладка разработанного ПО.

В третьем разделе составляется план развертывания и интеграции системы на предприятии. Формируется модель обновлений и технической поддержки.

В заключении необходимо описать навыки и умения, приобретенные за время практики; сделать выводы по достигнутой цели и решенным задачам практики.

В списке использованной литературы необходимо отразить ссылки на учебно-методические материалы, в том числе и электронные библиотеки, техническую документацию, нормативно-правовую информацию, интернет-источники, которые были использованы при прохождении практики и формировании отчетной документации для выполнения программы практики. Также допускается указание специализированных Интернет-ресурсов, посвященных ИТ-тематике. Список литературы следует оформлять по ГОСТ Р 7.0.100-2018.

Основными требованиями, предъявляемыми к оформлению отчета

Отчет выполняется в электронной форме с использованием текстового редактора, поддерживающим сложное форматирование. Отчет должен быть представлен в формате редактируемого файла, например: .odt, .docx, .rtf. **Нельзя предоставлять отчет в виде неотредактируемого pdf-файла.** Имя файла отчета должно иметь следующий вид:

«ФИО_ПД_Отчет.odt»

Например: «Иванов_Иван_Иванович_ПД_Отчет.docx» или «Иванов_Иван_Иванович_ПД_Отчет.rtf»

Используемые шрифты должны быть внедрены в документ для корректного отображения на других устройствах.

В документ отчета, допускается вставка таблиц, картинок, схем. Также допускается вставка фрагментов программного кода, который должен быть выделен от основного текста рамкой. Текст на листе должен иметь книжную ориентацию, альбомная ориентация допускается только для таблиц и блок-схем алгоритмов. Поля страницы должны иметь следующие размеры: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее – 2 см. Текст печатается через полтора интервала шрифтом Times New Roman, 14 кегль (для сносок 12 кегль), выравнивание по ширине. Абзац – 1,5 см. Абзацы отступы в таблицах не допускаются. Формулы необходимо представлять с помощью редактора формул. Блок-схемы алгоритмов должны быть выполнены в векторном редакторе, либо с использованием стандартных векторных средств рисования текстового редактора. Использование растровых картинок для блок-схем не допускается. При создании блок-схем и диаграмм

запрещается пользоваться устаревшим программным обеспечением, таким как BPWin и ERWin.

Таблицы, схемы, рисунки должны иметь собственные наименования и нумерацию. Нумерация таблиц, схем, рисунок может быть как сквозной, так и включать номер раздела. На все эти объекты должны быть ссылки по тексту.

Номера страниц размещаются в нижнем правом углу. Применяется сквозная нумерация листов, начиная с титульного листа и включая приложения (если есть). Номер листа на титульном листе не проставляют. Второй лист отчета – содержание.

Объем отчета зависит от индивидуального задания и может быть от 30 до 50 страниц.

Титульный лист отчета по Эксплуатационной практике генерируется из личного кабинета электронного университета при нажатии на ссылку со сроками практики. Титульный лист помимо основной информации содержит в правом верхнем углу персонализированный штрих-код, который является уникальным и позволяет идентифицировать как ФИО самого студента, так и всю сопроводительную документацию по производственной (эксплуатационной) практике. Титульный лист вставляется в качестве первого листа в отчет. Подписывать титульный лист не нужно.

Отчет проверяется на объем заимствований через систему «Антиплагиат» согласно Положению об использовании системы «Антиплагиат» в образовательной и научной деятельности, размещенному на официальном сайте университета.

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ПРАКТИКЕ

Оценочные средства для проведения промежуточной аттестации обучающихся по итогам Практики представлены в Приложении 4 и является неотъемлемой частью настоящей программы.

5. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ И РЕСУРСОВ СЕТИ ИНТЕРНЕТ, НЕОБХОДИМЫХ ДЛЯ ПРОВЕДЕНИЯ ПРАКТИКИ

Основная литература:

1. Гладких, Т. В. Информационные системы учета и контроля ресурсов предприятия: учебное пособие: [16+] / Т. В. Гладких, Л. А. Коробова, М. Н. Ивлиев; науч. ред. Д. С. Сайко; Воронежский государственный университет инженерных технологий. – Воронеж : Воронежский государственный университет инженерных технологий, 2020. – 89 с. : ил., табл., схем., граф. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=612378>
2. Зайцева, О. С. Технологии разработки web-ресурсов: учебное пособие: [16+] / О. С. Зайцева; Тюменский индустриальный университет. – Тюмень : Тюменский индустриальный университет, 2020. – 75 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=611103>
3. Ипатова, Э. Р. Методологии и технологии системного проектирования информационных систем: учебник / Э. Р. Ипатова, Ю. В. Ипатов. – 3-е изд., стер. – Москва : ФЛИНТА, 2021. – 256 с. : табл., схем. – (Информационные технологии). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=79551>
4. Программирование, тестирование, проектирование, нейросети, технологии аппаратно-программных средств (практические задания и способы их решения): учебник: [16+] / С. В. Веретехина, К. С. Кармицкий, Д. Д. Лукашин [и др.]. – Москва : Директ-Медиа, 2022. – 144 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=694782>
5. Булычев А.А. Система управления версиями GIT и российский сервис хранения исходного кода GitFlic: учеб. пособие - М.: Московский университет им. С.Ю. Витте, 2022. - 143 с. – [Электронный ресурс] - https://e.muiv.ru/local/library/library.php?book_id=148809

6. И.А. Королькова, М.В. Преображенский, С.А. Зайцев. Разработка и проектирование дизайна web-интерфейсов UI/UX : учебное пособие / И.А. Королькова, М.В. Преображенский, С.А. Зайцев; Моск. ун-т им. С.Ю. Витте; каф. ИС [Электронное издание]. – М.: изд. ЧОУВО «МУ им. С.Ю. Витте», 2022. – 9,02 Мб. URL: https://e.muiv.ru/local/library/library.php?book_id=148770

7. И.П. Романова, П.С. Романов Базы данных: работа с PostgreSQL: учебное пособие - М.: Московский университет имени С.Ю. Витте, 2023. - 193 с. – [Электронный ресурс] - https://e.muiv.ru/local/library/library.php?book_id=148859

Дополнительная литература:

1. Лисяк, В. В. Разработка информационных систем : учебное пособие : [16+] / В. В. Лисяк ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 97 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=577875>

2. Лауферман, О. В. Разработка программного продукта: профессиональные стандарты, жизненный цикл, командная работа : [16+] / О. В. Лауферман, Н. И. Лыгина ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2019. – 75 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=576397>

3. Математические и инструментальные методы в экономике, бизнесе и менеджменте / Е. А. Березовская, А. М. Галицына, А. Т. Калмакова и др. ; отв. ред. С. В. Крюков ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 164 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=598563>

4. Черных, В. В. Управление разработкой и внедрением инновационного продукта : учебное пособие : [16+] / В. В. Черных ; Поволжский государственный технологический университет. – Йошкар-Ола : Поволжский государственный технологический университет, 2019. – 122 с. : табл., граф., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=570613>

5. Пирская, Л. В. Разработка мобильных приложений в среде Android Studio: учебное пособие: [16+] / Л. В. Пирская; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 125 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=598634>

6. Скороход, С. В. Программирование на платформе 1С:Предприятие 8.3 : учебное пособие : [16+] / С. В. Скороход ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 136 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=577921>

7. Курчеева, Г. И. Информационное и программное обеспечение электронного бизнеса: учебное пособие: [16+] / Г. И. Курчеева, М. А. Бакаев, В. А. Хворостов. – Новосибирск : Новосибирский государственный технический университет, 2018. – 107 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=576386>

Информационное обеспечение и Интернет-ресурсы

1. Русскоязычный веб-сайт в формате коллективного блога с элементами новостного сайта, созданный для публикации новостей, аналитических статей, мыслей, связанных с информационными технологиями, бизнесом и интернетом <https://habr.com/ru/>

2. Интернет-издание в сфере высоких технологий в России и странах СНГ – <http://www.cnews.ru>

3. Библиотека федерального портала «РОССИЙСКОЕ ОБРАЗОВАНИЕ» <http://www.edu.ru>

Справочно-правовые системы

- Гарант <http://www.garant.ru>;
- Консультант ПЛЮС <http://www.consultant.ru>

6. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ПРОВЕДЕНИИ ПРАКТИКИ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

В целях формирования навыков использования современных информационных технологий, при прохождении практики обучающиеся обеспечиваются:

- мультимедийными технологиями (проекторы, ноутбуки, персональные компьютеры, комплекты презентаций, видеоматериалы);
- дистанционной формой консультаций во время прохождения конкретных этапов производственной практики и подготовки отчета, которая обеспечивается: выходом в глобальную сеть Интернет, поисковыми системами Яндекс, Мейл, Гугл, системами электронной почты;
- лицензионными программными продуктами, являющимися частью электронной информационно-образовательной среды университета и базирующимися на телекоммуникационных технологиях, которые представлены:
 1. Информационной справочной правовой системой: КонсультантПлюс;
 2. Электронной библиотечной системой МУ им. С.Ю. Витте - <https://online.muiv.ru/lib/>;
 3. Электронной библиотечной системой «Университетская библиотека онлайн» - <http://biblioclub.ru>.

Кроме того, при прохождении практики в профильных организациях, обучающиеся обеспечиваются доступом к пользованию программными продуктами, имеющимися в распоряжении организации.

7. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ПРОХОЖДЕНИЯ ПРАКТИКИ

В соответствии с содержанием договора, каждая организация обеспечивает обучающихся-практикантов автоматизированным рабочим местом, доступом к необходимой информации для решения задач практики.

Перечень материально-технического обеспечения прохождения практики в профильной организации может меняться, в зависимости от особенностей деятельности.

При прохождении Практики в вузе обучающиеся обеспечиваются материально-техническим оборудованием и библиотечными фондами вуза. Каждый практикант имеет доступ компьютерными технологиями, мультимедийному оборудованию.

8. СВЕДЕНИЯ ОБ УТВЕРЖДЕНИИ ПРОГРАММЫ ПРАКТИКИ И ВНЕСЕНИИ ИЗМЕНЕНИЙ

Программа разработана на основе Федерального государственного образовательного стандарта высшего образования по направлению подготовки 38.03.05 Бизнес-информатика, рассмотрена на заседании кафедры информационных систем протокол №7 от 16.02.2024г.

Автор-разработчик:

Руководитель образовательной программы факультета информационных технологий,
Королькова И.А.

9. ПРИЛОЖЕНИЯ

Приложение 1

Руководителю центра карьеры,
практики и трудоустройства / Директору
филиала / Директору колледжа

(фамилия, инициалы)

от обучающегося _____ курса

форма обучения _____

направление подготовки (специальность) _____

профиль (специализация, направленность)* _____

фамилия _____

имя _____

отчество _____

№ договора/студ. билета _____

контактный телефон (обучающегося) _____

e.mail: _____

Заявление

Прошу разрешить мне прохождение практики

(вид и тип)

В _____,
(наименование организации)

расположенной по адресу: г. _____, ул. _____ д. _____ к. _____

в период с « ____ » _____ 202__ г. по « ____ » _____ 202__ г.

Прошу назначить ответственным лицом от организации

/ФИО ответственного лица, указывается обучающимся/

В оплату расходов, связанных с проездом к месту проведения практической подготовки и обратно, а также расходов по проживанию в период практической подготовки не нуждаюсь, так как практическая подготовка проводится по месту моего жительства.

Приложение: Договор о практике обучающихся Частного образовательного учреждения высшего образования «Московский университет имени С.Ю. Витте».

Дата _____

Подпись _____

* указывается при наличии

Приложение 2.1

Содержание типового индивидуального задания, в случае прохождения практики в
Университете



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ С.Ю. ВИТТЕ

Руководитель практики*
от профильной организации

_____/...../

«__» _____ 20__ г.

Руководитель практики от Университета
(филиала Университета)

_____/...../

«__» _____ 20__ г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Студент _____

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль) Цифровая экономика

Вид практики производственная практика

Тип практики Преддипломная практика

Срок прохождения практики _____

Место проведения практики (профильное подразделение Университета) _____

С требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка на предприятии ознакомлен.

_____/...../ 20__ г.

Подпись

ФИО обучающегося

* для практики на базе Университета не вносится

Период практики	Содержание работ	Планируемые результаты практики (формируемые компетенции)	Отметка о выполнении и*
	Виды проделанной работы, результаты которой необходимо отразить в отчете:		
1 неделя	- оформить документы на практику, ознакомиться с требованиями по прохождению практики и правилами оформления документов по практике. - Получить у руководителя практики от Университета и ознакомиться с индивидуальным заданием прохождения преддипломной практики.	ПК-3, ПК-4	вып.
	- ознакомиться со структурой Университета, основными учредительными документами, внутренней организационно-распорядительной документацией, нормативно-правовыми документами и внутренними регламентами Университета в области ИТ-инфраструктуры на основе данных раздела официального сайта «Сведения об образовательной организации» и корпоративного портала - выбрать один и/или несколько бизнес-процессов согласно тематике ВКР - выбрать подразделение, ответственное за выбранный ранее бизнес-процесс		

Период практики	Содержание работ	Планируемые результаты практики (формируемые компетенции)	Отметка о выполнении и*
	– провести анализ ИТ-инфраструктуры выбранного подразделения. – провести анализ требований пользователей к разрабатываемой корпоративной информационной системе – выполнить формализацию и систематизацию требований пользователей. – Разработать модель «КАК ЕСТЬ» – составить техническое задание на разработку / модификацию / развертывание составляющих компонентов корпоративной информационной системы. – сформировать материалы для аналитической частей выпускной квалификационной работы		
2-3 неделя	– разработать предложения по архитектуре корпоративной информационной системы в соответствии с требованиями к разрабатываемому ПО из Приложения к данному индивидуальному заданию – Разработать модель «КАК ДОЛЖНО БЫТЬ» – составить архитектурную спецификацию компонентов корпоративной информационной системы – разработать прототип корпоративной информационной системы / программного модуля / библиотеки. – провести тестирование и согласование прототипа / макета пользовательского интерфейса корпоративной информационной системы / программного модуля / библиотеки – выполнить проектирование и разработку структуру базы данных / системы хранения данных – разработать / Адаптировать / модифицировать / разработать программный модуль / приложение в соответствии с требованиями к разрабатываемому ПО из Приложения к данному индивидуальному заданию – провести верификацию и тестирование программного кода – провести корректировку обнаруженных ошибок в программном коде	ПК-1, ПК-2, ПК-3, ПК-4, ПК-7	
4 неделя	– сформировать модель доступа к данным и определить права пользователей. – выполнить разграничение прав пользователей с целью обеспечения информационной безопасности. – составить план инсталляции и развертывания компонентов корпоративной информационной системы / программного модуля / библиотеки. – разработать план интеграции информационной системы / программного модуля / библиотеки с существующими ИС у заказчика; предложить интерфейсы взаимодействия и обмена данными / разработать программные модули для реализации обмена данными между разработанной информационной системы /	ПК-1, ПК-2, ПК-3, ПК-4	вып.

Период практики	Содержание работ	Планируемые результаты практики (формируемые компетенции)	Отметка о выполнении и*
	программного модуля / библиотеки с существующими ИС – разработать модуль обновлений для предлагаемой корпоративной информационной системы / программного модуля / библиотеки; разработать модель обновлений и технической поддержки. – сформировать материалы для проектной части выпускной квалификационной работы – Подготовить отчет по результатам прохождения производственной практики (преддипломной практики). Обсудить с руководителем выполненную работу. – Оформить отчет по практике. Выгрузить отчетную документацию в ЭИОС «Электронный университет».		

Планируемые результаты практики

Коды компетенций	Содержание компетенции
ПК-1	Способность создавать, редактировать и наполнять контентом информационные ресурсы
ПК-2	Способность разработки прикладного программного обеспечения для управления бизнесом
ПК-3	Способность выполнять анализ бизнес-информации
ПК-4	Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами
ПК-7	Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования

Планируемые результаты практики, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование профессиональной компетенции выпускника программы бакалавриата	Код и наименование индикатора достижения профессиональной компетенции
ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы	ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента.
ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом	ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владеет навыками автоматизации решения типовых задач для управления бизнесом.
ПК-3. Способность выполнять анализ бизнес-информации	ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2.

	Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации.
ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
ПК-7 Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования	ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владеть методами разработки дизайна ИТ сервисов

Прим.: Перечень представленных заданий может быть изменен или дополнен руководителем практики от Университета.

Приложение к индивидуальному заданию

1. Требования к разрабатываемому программному обеспечению на базе платформы 1С: Предприятие

- Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github **файла с выгрузкой конфигурации 1С *.dt** (возможно приложить архив всей папки с конфигурацией, с комментариями о версии платформы 1С, для которой выполнялся проект).
- Конфигурация должна содержать следующий минимальный набор объектов:
 - Справочники – **не менее 15**, из них **не менее 3 иерархических**.
 - Документы – **не менее 10**.
 - Константы – **не менее 2**.
 - Перечисления – **не менее 3**.
 - Общие модули – **не менее 1**.
 - Обработки – **не менее 1**.
 - Отчеты – **не менее 5**, в том числе сложные – **не менее 3**. Сложным отчетом считается отчет, для построения которого следует обратиться к данным **не менее 3-х объектов конфигурации**. Для решения задач учета следует использовать специализированные объекты (планы и регистры).
 - Подсистемы - **не менее 4**.
 - Регистры накопления – **не менее 3**.
- Отчеты, представленные в конфигурации должны иметь **печатные формы**.

4. Проанализировать необходимость использования **регистров сведений и регистров расчета**. Разработать необходимые регистры сведений и регистры накопления. Анализ отразить в ВКР.
5. Разработать механизм проведения документов по регистрам накопления, а также по иным регистрам конфигурации, при наличии
6. Создать **не менее 3-х ролей пользователей** конфигурации.
7. Объем собственноручно разработанного программного кода должен быть **не менее 500 логических строк кода**. Пустые строки, строки комментариев, строки только с управляющими символами подсчету не подлежат. Программный код, созданный автоматически с использованием платформы 1С:Предприятие не учитывается.
8. Для основных сценариев использования конфигурации должны быть созданы бизнес-процессы.
9. Заполнить конфигурацию тестовыми данными, соответствующими предметной области и выбранной организации. **Не менее 10 примеров для каждого объекта конфигурации**.

2. Требования к разрабатываемому ПО на языке высокого уровня

1. Программное обеспечение разрабатывается на языке **Python** с использованием сред разработки **PyCharm / VS Code / Google Colab / Yandex DataSphere**.
2. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github **исполняемого *.exe файла и файлов программного кода *.py** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях), либо **файла программного кода *.ipynb**, если работа выполнялась с использованием языка программирования Python и сервисов Google Colab и Yandex DataSphere.
3. Объем собственноручно разработанного программного кода должен быть **не менее 2 000 логических строк кода**. Пустые строки, строки комментариев, строки только с управляющими символами подсчету не подлежат.
4. Обязательно должен быть сформирован **исполняемый *.exe файл**
5. В качестве приоритетного решения следует рассматривать программное обеспечение, построенное с использованием клиент-серверной архитектуры. Разработка программного решения, не предполагающего сетевое взаимодействие должна быть обоснована.
6. Обеспечить хранение данных приложения в СУБД, поддерживающей SQL-запросы. Рекомендуются PostgreSQL.
7. В приложении должно быть **не менее 10 оконных форм**.
8. Реализовать **не менее одного диалогового окна** для получения данных от пользователя со специальной формы.
9. Реализовать функционал доступа к файловой системе.
10. Реализовать панель управления / настройки в приложении, где добавить **не менее 5 пунктов меню**.
11. Реализовать возможность формирования документов в форматах **.docx, .xlsx** и т.п. из разрабатываемого приложения.
12. Предусмотреть **не менее 3-х ролей пользователей**. Предоставить учетные данные всех созданных пользователей.
13. Разработать панель администратора.
14. Разработать личный кабинет пользователя.

15. Предусмотреть функционал «Справка по системе».

В случае разработки ПО для микроконтроллеров и программируемых логических интегральных схем:

16. Разработка кода должна выполняться с использованием аппаратного обеспечения.
17. Результатом выполнения выпускной квалификационной работы должен быть работающий программно-аппаратный комплекс.

3. Требования к разрабатываемому программному обеспечению для мобильных платформ

1. Приложение должно быть разработано под операционную систему Android. Язык написания приложения Java или Kotlin.
2. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github исполняемого ***.apk файла-приложения для Android** и исходных файлов с программным кодом.
3. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента, происходящих в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
4. **Использование готовых универсальных кроссплатформенных фреймворков запрещается.** Используемые средства разработки не регламентируются, однако рекомендуется использовать Android Studio для уменьшения проблем со сборкой приложения в аудитории.
5. Определить не менее **5 взаимосвязанных сущностей** в рамках предметной области.
6. Для организации сущностей использовать модель MVC или MVVM.
7. Для организации интерфейса в файлах представлений использовать не менее одного макета ConstraintLayout.
8. **Не менее одного макета** представления сформировать как для портретного, так и для альбомного режима.
9. Ресурсы, в том числе строковые, вынести из кода и расположить в отдельных предусмотренных для этого файлах и папках в структуре проекта.
10. Контроллеры представлений выполнять с помощью класса Fragment и библиотеки поддержки androidx. Разработать взаимодействие фрагментов на разных этапах жизненного цикла.
11. У каждой сущности должна быть форма редактирования объекта, а также форма списка объектов, построенная на базе класса RecyclerView.
12. Обеспечить хранение данных приложения в СУБД, поддерживающей SQL-запросы. Допустимо использование библиотек ORM для упрощения доступа к данным. Рекомендуется использование SQLite.
13. Реализовать панель управления в приложении, где добавить **не менее 5 пунктов меню**.
14. Реализовать **не менее одного диалогового окна** для получения данных от пользователя со специальной формы.

15. Добавить в функционал возможность вызова стандартных приложений через неявные намерения (Intent).
16. В основной сущности должен быть реализован функционал доступа к файловой системе и получение изображений с камеры.
17. Реализовать взаимодействие основной сущности с внешним веб-сервисом.

4. Требования к разрабатываемому сетевому ресурсу (веб-сайта, порталу и т.п.)

1. Сетевой ресурс должен быть разработан с применением языков программирования – PHP, JavaScript или Python с использованием фреймворков Django / Flask.
2. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github **файлов веб-страниц в формате *.html, *.css, *.php, *.js** (при необходимости архивируется вся папка проекта, с комментариями об используемых версиях) или **файлами программного кода *.py** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях), если работа выполнялась с использованием языка программирования Python.
3. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента, происходящих в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
4. Дополнительно сетевой ресурс должен быть размещен на бесплатном хостинге.
5. При работе над сетевым ресурсом **недопустимо подменять разработку ресурса разворачиванием готового программного обеспечения** и установку готовых дополнительных программных модулей для работы с ним. Допускается разработка собственных программных модулей для работы с имеющимися программными системами и/или модификация таковых модулей. **Запрещено использование CMS и готовых плагинов к ней**. Исключением для данного пункта могут быть работы, тема которых явно указывает на данный тип деятельности.
6. При разработке сетевых ресурсов студентом должно быть предложено программное и/или аппаратное решение, готовое к внедрению в выбранный бизнес-процесс.
7. Объем программного кода должен быть **не менее 2 000 логических строк кода**, в том числе и строк, написанных с применением языка разметки. Пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
8. Разработать структуру Web-страниц. Сетевой ресурс должен содержать **не менее 10 страниц**.
9. Разработать макет каждой страницы сетевого ресурса.
10. Предусмотреть форму обратной связи.
11. Разработать **не менее 10 пунктов бокового и/или верхнего меню, «хлебные крошки»**.
12. Обеспечить хранение данных приложения в СУБД, поддерживающей SQL-запросы.
13. Предусмотреть **не менее 3-х ролей пользователей**, например, администратор, менеджер, клиент. Предоставить учетные данные всех созданных пользователей.
14. Разработать панель администратора сетевого ресурса.
15. Разработать личный кабинет пользователя сетевого ресурса.

16. Обеспечить корректное отображение сетевого ресурса в основных браузерах (Mozilla Firefox, Chrome, Яндекс.Браузер, Microsoft Edge)
17. Обеспечить корректное отображение сетевого ресурса на разных устройствах (ПК, ноутбук, смартфон)
18. Текст статей в сетевом ресурсе должен быть читабельным, грамотно сформулированным, корректным и соответствовать предметной области ВКР. Не должно быть пустых страниц в сетевом ресурсе.

5. Требования к разрабатываемому интеллектуальному сервису (с использованием машинного обучения, нейронных сетей и интеллектуального анализа данных)

1. Интеллектуальный сервис разрабатывается на языке Python с использованием сервисов Google Colab или Yandex DataSphere.
2. Обязательна загрузка в ЭУ вместе с ВКР **файла/ов программного кода *.ipynb** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях, а также с используемыми датасетами). Обязательна выгрузка в репозиторий Gitflic или Github **файла/ов программного кода *.ipynb** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях, а также с используемыми датасетами)
3. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента**, происходящих **в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
4. Объем собственноручно разработанного программного кода должен быть **не менее 2 000 логических строк кода**. Считаются только строчки алгоритмического кода, строки кода разметки типа HTML или JSON, пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
5. Создаваемое приложение для демонстрации работы приложения обязательно должно позволять:
 - загружать и сохранять данные для обучения моделей из локальных файлов или из внешних хранилищ данных (типа kaggle или коннекторов для подключения хранилищ на облачных сервисах);
 - проводить обучение моделей, сохранять обученную модель в файл по результатам обучения;
 - загружать обученную модель из файла.
 - применять обученную или загруженную модель по назначению
 - оценивать и сохранять результаты работы модели для обучения новых версий модели
6. Дизайн приложения должен быть отражён в отчёте. Загрузка данных в приложение отражается в отчёте в **разделе загрузки данных**, обучение моделей в приложении - в **разделе обучения моделей**, процесс применения модели и оценки её результатов – в **разделе тестирования модели**.

7. В git-репозитории должен присутствовать файл **requirements.txt**, описывающий список необходимых для работы приложения и модели библиотек, которые надо будет установить проверяющему в пустое виртуальное окружение. Файл необходимо положить в корневой каталог репозитория.
8. В git-репозитории должен присутствовать в корневом каталоге файл **main.py**, который запускает приложение для работы с моделью.
9. Все демонстрируемые в отчёте модели должны быть сохранены в сдаваемом git-репозитории в файле с названием **model** и расширением, которое удобно для конкретной выбранной архитектуры (например, модели на PyTorch сохраняются с расширением **.pt**, то есть будет **model.pt**). Если в отчёте демонстрируется больше 1 модели, то к названию файла добавляется порядковый номер (напр., **model1**, **model2** и т.д.). При проверке будет ожидаться, что работать эти модели будут так же, как описано в отчёте.
10. Для всех демонстрируемых моделей, в отчёте должны быть сохранены данные для их обучения. При проверке будет ожидаться, что модель, обученная на этих данных, будет совпадать по метрикам своей эффективности с моделью в файле и описанием этих метрик в отчёте.
11. При работе моделей, ожидается достижений уровня эффективности, сопоставимого с уровнем **Accuracy** ≥ 70 . Если к конкретной модели в конкретной предметной области этот уровень не применим, то можно согласовать с руководителем работы другую метрику эффективности, отразив её в отчёте.
12. Реализация GUI интеллектуального сервиса может быть в виде «оконных» или web форм, а также с использованием возможностей чат-ботов по отрисовке интерфейсных компонентов. Конкретный формат GUI согласовывается с руководителем ВКР.
13. В ВКР обязательно должно быть математическое описание выбранной модели.

Методологические требования к разработке интеллектуального сервиса

1. Основной формат демонстрации навыков – **создание, обучение, тестирование и улучшение модели машинного обучения** в выбранной предметной области, сопровождаемое **визуальным анализом** получаемых результатов. Формат сохраняется не зависимо от сложности модели.
2. Создание модели предполагает, что студент умеет применять основные способы создания моделей в программном коде, как в плане написания алгоритмов «с нуля», так и в плане использование стандартных готовых библиотек – от базовых библиотек типа NumPy и Pandas, до библиотек глубокого обучения типа PyTorch и Tensorflow.
3. Создание модели предполагает, что студент владеет не только типовыми способами создания моделей, но и способен адаптировать модель под работу в конкретной предметной области, осуществляя математически грамотный подбор гиперпараметров моделей, структуру ансамблевых моделей или топологий нейронных сетей.
4. Уверенное владение моделями предполагает знакомство студентов с алгоритмами оптимизации и функциями подсчёта потерь, способность объяснить математический смысл этих алгоритмов и функций и их отношение к целевой функции, выражаемой моделью.

5. Обучение модели предполагает получение данных, на основе которых проводится обучение, подготовку данных к работе, а так же проведение самого процесса обучения модели с визуализацией промежуточных и итоговых результатов.
6. Получение данных предполагает, что студент знает откуда вообще можно брать данные для обучения моделей. Знаком как с публичными ресурсами типа kaggle, так и с подходами к самостоятельной добыче данных через web-скрейпинг, публичные API или через коннекторы для подключения хранилищ на облачных сервисах.
7. **Генерация случайных данных не допускается**, поскольку случайные данные автоматически лишают результаты работы конкретной практической ценности.
8. Получение данных предполагает, что студент понимает, какие форматы данных необходимы разным типам моделей для успешной работы в выбранной предметной области, способен привести исходные данные в нужный формат, производя их очистку и необходимые трансформации, такие как нормализация, векторизация, приведение к тензорам, в том числе с использованием готовых конструкций загрузки данных в профессиональных библиотеках.
9. Получение данных предполагает способность студента определить, какие данные полезны, а какие нет, применяя для этого разные способы отбора и фильтрации признаков, в том числе на основе подходов визуального анализа.
10. Получение данных предполагает способность студента выделять из данных тренировочные и тестовые выборки, знать стандартные способы и инструменты такого выделения, уметь применять эти способы с учётом нужд предметной области и задачи анализа.
11. Обучение модели предполагает, что студент знает, как запустить такое обучение в рамках выбранного им способа создания модели. На минимальном уровне – умеет применять стандартные инструменты обучения моделей, представленных в рабочих библиотеках. На максимальном уровне – способен самостоятельно реализовать алгоритм обучения модели методом градиентного спуска.
12. Обучение многослойных моделей предполагает способность студента запускать автоматические фазы обучения (т.н. «эпохи»), отслеживать и визуализировать изменение эффективности модели на каждом этапе обучения.
13. По результатам обучения, в некоторых случаях - и в процессе обучения, необходимо проводить тестирование моделей. Для этого надо знать метрики оценки эффективности моделей, уметь производить тестирование и применять метрики к результатам тестирования, владеть инструментами визуализации метрик.
14. Тестирование моделей предполагает от студента способность не только произвести тестирование, но и сделать содержательные выводы из его результатов. Содержательными считаются выводы о работоспособности модели, полезности результатов обучения, может быть возможных способах улучшения модели.
15. Улучшение модели машинного обучения предполагает внесение изменений в модель с целью получить улучшенные результаты. От студента ожидается способность сформулировать предложение об улучшении модели по результатам первичного тестирования, обосновать его с помощью математического и визуального анализа, выполнить это улучшение (изменив параметры модели или пересоздав модель заново с новой топологией) и заново провести обучение и тестирование, оценив результаты улучшения (то есть фактически повторить весь путь с улучшенной моделью). Это должно быть отражено в ВКР.

Студенты должны демонстрировать все вышеперечисленные пункты по отношению к любому типу модели машинного обучения, представленному в стандартных рабочих библиотеках.

Таким образом, студент предоставляет не только работающую модель, но и приложение как среду, в которой он тестировал работу своей модели и в которой её работу может протестировать руководитель ВКР или другой проверяющий. Отсутствие среды, в которой можно проверить работу модели, является нарушением правил сдачи работы.

6. Требования к разрабатываемому интерактивному графическому приложению (компьютерной игре)

1. Интерактивное графическое приложение разрабатывается на языке **Python** с использованием интерпретатора **PyCharm** или **Google Colab**.
2. Обязательна загрузка в ЭУ вместе с ВКР **исполняемого *.exe файла** и выгрузка в репозиторий Gitflic или Github **файлов программного кода *.py** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях) если работа выполнялась с использованием языка программирования Python, либо **файла программного кода *.ipynb**, если работа выполнялась с использованием языка программирования Python и сервисов Google Colab и Yandex DataSphere.
3. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента**, происходящих **в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
4. Объем собственноручно разработанного программного кода должен быть **не менее 2 000 логических строк кода**. Считаются только строки алгоритмического кода, строки кода разметки типа HTML или JSON, пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
5. При описании внутренней архитектуры приложения, студент должен описать **не менее 10 сущностей игровой системы** (из которых **минимум 5** должны быть реализованы самостоятельно) и **не менее 5 игровых систем** (из которых **минимум 3** должны быть реализованы самостоятельно).
6. При описании интерфейса приложения, студент должен описать **не менее 4 игровых экранов и не менее 15 элементов интерфейса** на этих экранах (из которых **не менее 8** взяты не в готовом виде, а реализованы самостоятельно). Элементом интерфейса считается любой интерфейсный объект, реагирующий на события в игровой системе (например, шкала здоровья или индикатор возможности действия) или сам инициирующий событийную цепочку по факту ввода от игрока (например, кнопка).
7. Архитектура приложения обязательно должна содержать локально сохраняемые данные, такие как пользовательские настройки, сохранения, пароли и т.д.
8. Для реализации приложения можно использовать любые игровые движки и middleware, при этом имея ввиду требование о реализации достаточного объема собственного функционала поверх заимствованного.

9. В сдаваемом git-репозитории должен в корневом каталоге присутствовать **файл readme.md**, описывающий способы управления игровым процессом, а также содержащий ссылки на скачивание запускаемых сборок игрового приложения.
10. Хранение сборок в самом git-репозитории не разрешается. Можно выгрузить сборки на любой внешний ресурс для скачивания или, что предпочтительнее, осуществить релиз своего приложения на специализированной площадке типа itch.io.
11. Необходимым условием является наличие сборки, которую можно запустить под Linux, для запуска на компьютерах Университета для проверки. **Сборки только под Windows недостаточно.** Если проект нельзя собрать для Linux или хотя бы для Web-запуска из браузера под Linux, то запускаемую сборку необходимо установить в соответствующий докер-контейнер.

Методологические требования к разработке интерактивного графического приложения

12. **Основной формат демонстрации навыков** – выполнение полного комплекса работ по разработке интерактивного графического приложения (далее «игрового приложения»), в частности: определение требований к игровому приложению, планирование отвечающего этим требованиям функционала, как в плане системной архитектуры, так и в плане графического интерфейса, с последующей реализацией этого функционала, его тестированием на предмет соответствия требованиям и улучшение приложения по результатам тестирования.
13. Формат демонстрации навыков в целом сохраняется не зависимо от природы конкретного игрового приложения. Планирование, реализация, тестирование и улучшения являются неотъемлемыми частями процесса создания любого такого приложения.
14. На этапе определения требований к игровому приложению возможна вариативность, в том числе и в плане демонстрируемых студентом навыков. Рассматривается три основных подхода к определению требований к игровому приложению: требования научной модели, требования дизайн-фокуса и требования коммерческого продукта. Эти три способа требуют демонстрирования разных наборов навыков на этапе определения требований к игровому приложению.
15. Определение требований к игровому приложению через требования научной модели используется когда целью работы ставится создание игровой модели какого-то сложного процесса. Примерами приложений, моделирующих подобные сложные процессы являются игровые приложения в стиле Microsoft Flight Simulator, Kerbal Space Program, Civilisation. Требования к игровому приложению в этом случае происходят из требований к точности реализации и визуализации работы научной модели и описанных в этой модели процессов.
16. При выборе этого способа определения требований к игровому приложению, наличие готовой научной модели, выраженной в соответствующем наборе научных публикаций, является необходимым граничным условием. Студент может взять реализацию игрового приложения на основе научной модели как задание только если предъявит руководителю научные публикации, описывающие соответствующую научную модель.
17. Если студент решает в своей работе определять требования к своему игровому приложению через призму требований научной модели, то от него будут ожидать

- a. Знание о научных моделях в выбранной им предметной области, описывающих моделируемый процесс. Способность адресовать к научным публикациям, обосновывать ими элементы дизайна игрового приложения.
 - b. Умение реализовывать выбранную научную модель в программном виде, как в плане внутренней рабочей логики этой модели, так и в плане визуализации её работы и демонстрации возникающей в этой модели интересных процессов.
18. Определение требований к игровому приложению на основании дизайн-фокуса предполагает наличие внешнего бизнес-процесса или группы бизнес-процессов, в котором существует проблема. Решение этой проблемы и становится дизайн-фокусом игрового приложения. Требования к игровому приложению в этом случае определяются на основании бизнес-анализа исходной группы бизнес-процессов и определения точек применения игрового приложения для разрешения проблем в этих процессах.
19. При выборе этого способа определения требований к игровому приложению, необходимо наличие у студента доступа к специалисту, который поможет ему описать конкретные бизнес-процессы и создаст тем самым возможность для их анализа с целью последующего выделения дизайн-фокуса игрового приложения. Студент может использовать в своей работе бизнес-процессы Университета (и тогда специалистом может выступать руководитель работы) или внешней организации (и тогда специалист должен быть из внешней организации). Отсутствие доступа к специалисту делает невозможным определение требований к игровому приложению через дизайн-фокус.
20. Если студент решает в своей работе определять требования к своему игровому приложению через дизайн-фокус, то от него будут ожидать:
- a. Навыки анализа бизнес-процессов и выделения в них тех точек, которые могут быть улучшены с помощью игрового приложения.
 - b. Знание технологий и подходов геймификации бизнес-процессов и способность применить эти знания для предложения улучшения в бизнес-процессах за счёт игрового приложения.
21. Определение требований к игровому приложению на основании требований коммерческого продукта предполагает, что студент в состоянии определить провести маркетинговое профилирование целевой аудитории продукта, может сформулировать потребности этой целевой аудитории и тем самым обосновать требования к игровому приложению как потребительским характеристикам, интересующих целевую аудиторию.

Если студент решает в своей работе определять требования к своему игровому приложению как к коммерческому продукту, то от него будут ожидать:

- a. Навыки поиска других продуктов в выбранной им категории игровых приложений, определения тех их ключевых качеств, на которые реагирует аудитория.
 - b. Способность профилирования аудитории выбранной категории игровых приложений с точки зрения интересующего конкретных представителей конкретного профиля набора потребительских качеств.
 - c. Способность выделить целевую аудиторию для своего продукта и спланировать потребительские качества собственного продукта, соответствующие маркетинговому профилю целевой аудитории.
22. Сформулировав требования к своему игровому приложению одним из трёх вышеописанных способов, студент переходит к планированию реализации игрового

приложения. Необходимо спланировать первую версию приложения, удовлетворяющего заданным требованиям, как в плане программной архитектуры, так и в плане интерфейсов и игровых взаимодействий.

23. На этапе планирования от студента ожидаются:

- a. Знания о типовых способах составления проектной документации для игрового приложения (далее «дизайн-документа»).
- b. Умение правильно структурировать дизайн-документ, отразив в нём как требования к игровому продукту, так и способы их реализации.
- c. Владение способами визуализации схем игровых взаимодействий и необходимых алгоритмов, видов игровых объектов и отношений между ними, способов управления игровым процессом.

24. Спланировав первую версию приложения, студент переходит к его реализации. На этом этапе, от студента ожидаются:

- a. Знания общих архитектурных схем, стандартных алгоритмов и типовых структур данных, позволяющих реализовать запланированное в дизайн-документе.
- b. Умение преобразовывать требования продукта (будь то потребительские качества, требования научной модели или схемы геймификации) в конкретный функционал игрового приложения.
- c. Владение инструментарием создания игровых приложений, будь то промежуточные библиотеки типа rpgame или Ren'Py, или интегрированные решения типа Unity.

25. Реализовав первую версию приложения, студент должен протестировать её относительно требований и сделать выводы о необходимых улучшениях. Конкретный характер тестов будет зависеть от способа формирования требований.

- a. В случае научной модели, необходимо реализовать архитектуру автоматического тестирования корректного выполнения научной модели во всех игровых сценариях, показать работу этой архитектуры.
- b. В случае коммерческого продукта, необходимо предложить игровой продукт целевой аудитории, собрать с неё обратную связь. После чего на основании этой обратной связи сделать выводы о том, насколько продукт отвечает потребностям аудитории в рамках сформулированного профиля, предложить и реализовать улучшения продукта.
- c. В случае дизайн-фокуса, необходимо согласовать конкретную схему проверки со специалистом, владеющим улучшаемым с помощью игрового приложения бизнес-процессом. Согласованную схему проверки будет необходимо выполнить, отразить в отчёте работы и после чего, на основании результатов, сделать выводы о работоспособности предложенной геймификации и внести улучшения.

В итоге, от студентов ожидается внятное объяснение требований, которые они сформулировали к реализуемому игровому приложению, способов выполнения этих требований, способов верификации выполнения этих требований, а также улучшений, которые были сделаны в тех местах, где требования с первого раза не выполнялись.

7. Требования к тестированию программного продукта

1. Для разрабатываемого приложения необходимо выбрать **минимум 4 метода тестирования** из нижеприведенных видов тестирования:

№	Вид тестирования	Метод тестирования	Описание метода
1	По принципам работы с приложением	Позитивное тестирование	тестирование, при котором используются только корректные данные.
		Негативное тестирование	тестирование приложения, при котором используются некорректные данные и выполняются некорректные операции
2	По уровню функционального тестирования	Смоук-тестирование (дымовое, smoke test)	тестирование, выполняемое на новой сборке, с целью подтверждения того, что программное обеспечение запускается и выполняет основные для бизнеса функции.
		Тестирование критического пути (critical path)	направлено для проверки функциональности, используемой обычными пользователями во время их повседневной деятельности.
		Расширенное тестирование (extended)	направлено на исследование всей заявленной в требованиях функциональности.
3	В зависимости от исполнителей	Альфа-тестирование	является ранней версией программного продукта. Может выполняться внутри организации-разработчика с возможным частичным привлечением конечных пользователей.
		Бета-тестирование	программное обеспечение, выпускаемое для ограниченного количества пользователей. Главная цель — получить отзывы клиентов о продукте и внести соответствующие изменения.
4	В зависимости от целей тестирования	Функциональное тестирование (functional testing)	направлено на проверку корректности работы функциональности приложения.
		Нефункциональное тестирование(non-functional testing):	тестирование атрибутов компонента или системы, не относящихся к функциональности.
		– Тестирование производительности (performance testing)	определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок.
		– Нагрузочное тестирование (load testing)	определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).
		– Тестирование масштабируемости (scalability testing)	тестирование, которое измеряет производительность сети или системы, когда количество пользовательских запросов увеличивается или уменьшается.
		– Объемное тестирование (volume testing)	это тип тестирования программного обеспечения, которое проводится для

			тестирования программного приложения с определенным объемом данных.
		– Стрессовое тестирование (stress testing)	тип тестирования направленный для проверки, как система обращается с нарастающей нагрузкой (количеством одновременных пользователей).
		– Инсталляционное тестирование (installation testing)	тестирование, направленное на проверку успешной установки и настройки, обновления или удаления приложения.
		– Тестирование интерфейса (GUI/UI testing)	проверка требований к пользовательскому интерфейсу.
		– Тестирование удобства использования (usability testing)	это метод тестирования, направленный на установление степени удобства использования, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.
		– Тестирование локализации (localization testing)	проверка адаптации программного обеспечения для определенной аудитории в соответствии с ее культурными особенностями.
		– Тестирование безопасности (security testing)	это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.
		– Тестирование надежности (reliability testing)	один из видов нефункционального тестирования ПО, целью которого является проверка работоспособности приложения при длительном тестировании с ожидаемым уровнем нагрузки.
		– Регрессионное тестирование (regression testing)	тестирование уже проверенной ранее функциональности после внесения изменений в код приложения, для уверенности в том, что эти изменения не внесли ошибки в областях, которые не подверглись изменениям.
		– Повторное/подтверждающее тестирование (re-testing/confirmation testing)	тестирование, во время которого исполняются тестовые сценарии, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок.
5	По запуску кода на исполнение	Статическое	процесс тестирования, который проводится для верификации практически любого артефакта разработки: программного кода компонент, требований, системных спецификаций, функциональных спецификаций, документов проектирования и архитектуры программных систем и их компонентов.
		Динамическое	тестирование проводится на работающей системе, не может быть

			осуществлено без запуска программного кода приложения.
6	По доступу к коду и архитектуре	Метод белого ящика	метод тестирования ПО, который предполагает полный доступ к коду проекта.
		Сетод серого ящика	метод тестирования ПО, который предполагает частичный доступ к коду проекта (комбинация White Box и Black Box методов).
		Метод черного ящика	метод тестирования ПО, который не предполагает доступа (полного или частичного) к системе. Основывается на работе исключительно с внешним интерфейсом тестируемой системы.
7	По уровню детализации приложения	Модульное тестирование	проводится для тестирования какого-либо одного логически выделенного и изолированного элемента (модуля) системы в коде. Проводится самими разработчиками, так как предполагает полный доступ к коду.
		Интеграционное тестирование	тестирование, направленное на проверку корректности взаимодействия нескольких модулей, объединенных в единое целое.
		Системное тестирование	процесс тестирования системы, на котором проводится не только функциональное тестирование, но и оценка характеристик качества системы — ее устойчивости, надежности, безопасности и производительности.
		Приемочное тестирование	проверяет соответствие системы потребностям, требованиям и бизнес-процессам пользователя.
8	По степени автоматизации	Ручное тестирование	
		Автоматическое тестирование	

2. **Сформировать тест план.** Тест план должен отвечать на следующие вопросы: Что необходимо протестировать? Как будет проводиться тестирование? Когда будет проводиться тестирование? Критерии начала тестирования. Критерии окончания тестирования.

Основные пункты тест плана:

- Идентификатор тест плана (Test plan identifier);
- Введение (Introduction);
- Объект тестирования (Test items);
- Функции, которые будут протестированы (Features to be tested);
- Функции, которые не будут протестированы (Features not to be tested);
- Тестовые подходы (Approach);
- Критерии прохождения тестирования (Item pass/fail criteria);
- Критерии приостановления и возобновления тестирования (Suspension criteria and resumption requirements);
- Результаты тестирования (Test deliverables);
- Задачи тестирования (Testing tasks);
- Ресурсы системы (Environmental needs);
- Обязанности (Responsibilities);
- Роли и ответственность (Staffing and training needs);

- Расписание (Schedule);
 - Оценка рисков (Risks and contingencies);
 - Согласования (Approvals).
3. **Сформировать тест-кейсы** (тестовые сценарии). Обязательные атрибуты тест кейса:
- **Предусловия** (PreConditions) — список действий, которые приводят систему к состоянию пригодному для проведения основной проверки. Либо список условий, выполнение которых говорит о том, что система находится в пригодном для проведения основного теста состоянии.
 - **Шаги** (Steps) — список действий, переводящих систему из одного состояния в другое, для получения результата, на основании которого можно сделать вывод о удовлетворении реализации, поставленным требованиям.
 - **Ожидаемый результат** (Expected result) — что по факту должны получить.
4. На все найденные ошибки **сформировать баг-репорт**. Баг-репорт должен обязательно содержать атрибуты: приоритет и серьезность.
- **Серьёзность** показывает, насколько баг влияет на возможность работать в программе. Виды:

Серьёзность бага	Как влияет на систему
S1 — Блокирующая (Blocker).	Приложение не запускается ни на одном устройстве или выдаёт ошибку при загрузке. <i>Пример: приложение не открывается, поэтому невозможно сделать покупку.</i>
S2 — Критическая (Critical).	Часть функционала не работает, но это не блокирует процесс. <i>Пример: Скидка по купону не работает, но товар всё равно можно купить без скидки.</i>
S3 — Значительная (Major).	Часть логики работает некорректно, но пользователь может решить проблему другим способом. <i>Пример: поиск в приложении не выдаёт нужный товар, но его можно найти, если искать по брендам.</i>
S4 — Незначительная (Minor).	Не нарушает логику приложения. <i>Пример: кнопка «Купить» уехала за экран и видна только наполовину.</i>
S5 — Тривиальная (Trivial).	Не относится к логике приложения и не влияет на общее качество продукта. <i>Пример: малозаметная опечатка в меню.</i>

Приоритет — это критерий, который показывает, насколько быстро нужно исправить дефект. Приоритет может быть:

- высокий — исправить в первую очередь;
 - средний — исправить, когда разобрались с первой категорией багов;
 - низкий — исправить, когда разобрались с багами других приоритетов.
5. Произвести отладку разработанной системы.

Приложение 2.2

Содержание типового индивидуального задания, при прохождении практики на базе профильной организации



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ С.Ю. ВИТТЕ

Руководитель практики*
от профильной организации
_____/...../
«__» _____ 20__ г.

Руководитель практики от
Университета
(филиала Университета)
_____/...../
«__» _____ 20__ г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Студент _____

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль) Цифровая экономика

Вид практики производственная практика

Тип практики Преддипломная практика

Срок прохождения практики _____

Место проведения практики _____

С требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка на предприятии ознакомлен.

_____/...../
Подпись ФИО обучающегося

* для практики на базе Университета не вносится

Период практики	Содержание работ	Планируемые результаты практики (формируемые компетенции)	Отметка о выполнении и*
	Виды проделанной работы, результаты которой необходимо отразить в отчете:		
1 неделя	- оформить документы на практику, ознакомиться с требованиями по прохождению практики и правилами оформления документов по практике. - Получить у руководителя практики от Университета и ознакомиться с индивидуальным заданием прохождения преддипломной практики.	ПК-3, ПК-4	вып.
	- ознакомиться со структурой Организации, основными учредительными документами, внутренней организационно-распорядительной документацией, нормативно-правовыми документами и внутренними регламентами Организации в области ИТ-инфраструктуры - выбрать один и/или несколько бизнес-процессов согласно тематике ВКР - выбрать подразделение, ответственное за выбранный ранее бизнес-процесс		

Период практики	Содержание работ	Планируемые результаты практики (формируемые компетенции)	Отметка о выполнении и*
	– провести анализ ИТ-инфраструктуры выбранного подразделения. – провести анализ требований пользователей к разрабатываемой корпоративной информационной системе – выполнить формализацию и систематизацию требований пользователей. – Разработать модель «КАК ЕСТЬ» – составить техническое задание на разработку / модификацию / развертывание составляющих компонентов корпоративной информационной системы. – сформировать материалы для аналитической частей выпускной квалификационной работы		
2-3 неделя	– разработать предложения по архитектуре корпоративной информационной системы в соответствии с требованиями к разрабатываемому ПО из Приложения к данному индивидуальному заданию – Разработать модель «КАК ДОЛЖНО БЫТЬ» – составить архитектурную спецификацию компонентов корпоративной информационной системы – разработать прототип корпоративной информационной системы / программного модуля / библиотеки. – провести тестирование и согласование прототипа / макета пользовательского интерфейса корпоративной информационной системы / программного модуля / библиотеки – выполнить проектирование и разработку структуру базы данных / системы хранения данных – разработать / Адаптировать / модифицировать / разработать программный модуль / приложение в соответствии с требованиями к разрабатываемому ПО из Приложения к данному индивидуальному заданию – провести верификацию и тестирование программного кода – провести корректировку обнаруженных ошибок в программном коде	ПК-1, ПК-2, ПК-3, ПК-4, ПК-7	
4 неделя	– сформировать модель доступа к данным и определить права пользователей. – выполнить разграничение прав пользователей с целью обеспечения информационной безопасности. – составить план инсталляции и развертывания компонентов корпоративной информационной системы / программного модуля / библиотеки. – разработать план интеграции информационной системы / программного модуля / библиотеки с существующими ИС у заказчика; предложить интерфейсы взаимодействия и обмена данными / разработать программные модули для реализации обмена данными между разработанной информационной системы /	ПК-1, ПК-2, ПК-3, ПК-4	вып.

Период практики	Содержание работ	Планируемые результаты практики (формируемые компетенции)	Отметка о выполнении и*
	программного модуля / библиотеки с существующими ИС – разработать модуль обновлений для предлагаемой корпоративной информационной системы / программного модуля / библиотеки; разработать модель обновлений и технической поддержки. – сформировать материалы для проектной части выпускной квалификационной работы – Подготовить отчет по результатам прохождения производственной практики (преддипломной практики). Обсудить с руководителем выполненную работу. – Оформить отчет по практике. Выгрузить отчетную документацию в ЭИОС «Электронный университет».		

Планируемые результаты практики

Коды компетенций	Содержание компетенции
ПК-1	Способность создавать, редактировать и наполнять контентом информационные ресурсы
ПК-2	Способность разработки прикладного программного обеспечения для управления бизнесом
ПК-3	Способность выполнять анализ бизнес-информации
ПК-4	Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами
ПК-7	Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования

Планируемые результаты практики, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование профессиональной компетенции выпускника программы бакалавриата	Код и наименование индикатора достижения профессиональной компетенции
ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы	ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента.
ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом	ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владеет навыками автоматизации решения типовых задач для управления бизнесом.
ПК-3. Способность выполнять анализ бизнес-информации	ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2.

	Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации.
ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
ПК-7 Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования	ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владеть методами разработки дизайна ИТ сервисов

Прим.: Перечень представленных заданий может быть изменен или дополнен в соответствии с профилем организации и фактическими служебными поручениями. Индивидуальное задание составляется руководителем практики от университета и согласовывается с руководителем практики от организации.

Приложение к индивидуальному заданию

8. Требования к разрабатываемому программному обеспечению на базе платформы 1С: Предприятие

10. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github **файла с выгрузкой конфигурации 1С *.dt** (возможно приложить архив всей папки с конфигурацией, с комментариями о версии платформы 1С, для которой выполнялся проект).
11. Конфигурация должна содержать следующий минимальный набор объектов:
 - Справочники – **не менее 15**, из них **не менее 3 иерархических**.
 - Документы – **не менее 10**.
 - Константы – **не менее 2**.
 - Перечисления – **не менее 3**.
 - Общие модули – **не менее 1**.
 - Обработки – **не менее 1**.
 - Отчеты – **не менее 5**, в том числе сложные – **не менее 3**. Сложным отчетом считается отчет, для построения которого следует обратиться к данным **не менее 3-х объектов конфигурации**. Для решения задач учета следует использовать специализированные объекты (планы и регистры).
 - Подсистемы - **не менее 4**.
 - Регистры накопления – **не менее 3**.
12. Отчеты, представленные в конфигурации должны иметь **печатные формы**.
13. Проанализировать необходимость использования **регистров сведений** и **регистров расчета**. Разработать необходимые регистры сведений и регистры накопления. Анализ отразить в ВКР.

14. Разработать механизм проведения документов по регистрам накопления, а также по иным регистрам конфигурации, при наличии
15. Создать **не менее 3-х ролей пользователей** конфигурации.
16. Объем собственноручно разработанного программного кода должен быть **не менее 500 логических строк кода**. Пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат. Программный код, созданный автоматически с использованием платформы 1С:Предприятие не учитывается.
17. Для основных сценариев использования конфигурации должны быть созданы бизнес-процессы.
18. Заполнить конфигурацию тестовыми данными, соответствующими предметной области и выбранной организации. **Не менее 10 примеров для каждого объекта конфигурации.**

9. Требования к разрабатываемому ПО на языке высокого уровня

18. Программное обеспечение разрабатывается на языке **Python** с использованием сред разработки **PyCharm / VS Code / Google Colab / Yandex DataSphere**.
19. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github **исполняемого *.exe файла и файлов программного кода *.py** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях), либо **файла программного кода *.ipynb**, если работа выполнялась с использованием языка программирования Python и сервисов Google Colab и Yandex DataSphere.
20. Объем собственноручно разработанного программного кода должен быть **не менее 2 000 логических строк кода**. Пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
21. Обязательно должен быть сформирован **исполняемый *.exe файл**
22. В качестве приоритетного решения следует рассматривать программное обеспечение, построенное с использованием клиент-серверной архитектуры. Разработка программного решения, не предполагающего сетевое взаимодействие должна быть обоснована.
23. Обеспечить хранение данных приложения в СУБД, поддерживающей SQL-запросы. Рекомендуется PostgreSQL.
24. В приложении должно быть **не менее 10 оконных форм**.
25. Реализовать **не менее одного диалогового окна** для получения данных от пользователя со специальной формы.
26. Реализовать функционал доступа к файловой системе.
27. Реализовать панель управления / настройки в приложении, где добавить **не менее 5 пунктов меню**.
28. Реализовать возможность формирования документов в форматах **.docx, .xlsx** и т.п. из разрабатываемого приложения.
29. Предусмотреть **не менее 3-х ролей пользователей**. Предоставить учетные данные всех созданных пользователей.
30. Разработать панель администратора.
31. Разработать личный кабинет пользователя.
32. Предусмотреть функционал «Справка по системе».

В случае разработки ПО для микроконтроллеров и программируемых логических интегральных схем:

- 33. Разработка кода должна выполняться с использованием аппаратного обеспечения.
- 34. Результатом выполнения выпускной квалификационной работы должен быть работающий программно-аппаратный комплекс.

10. Требования к разрабатываемому программному обеспечению для мобильных платформ

- 18. Приложение должно быть разработано под операционную систему Android. Язык написания приложения Java или Kotlin.
- 19. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github исполняемого ***.apk файла-приложения для Android** и исходных файлов с программным кодом.
- 20. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента**, происходящих **в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
- 21. **Использование готовых универсальных кроссплатформенных фреймворков запрещается.** Используемые средства разработки не регламентируются, однако рекомендуется использовать Android Studio для уменьшения проблем со сборкой приложения в аудитории.
- 22. Определить не менее **5 взаимосвязанных сущностей** в рамках предметной области.
- 23. Для организации сущностей использовать модель MVC или MVVM.
- 24. Для организации интерфейса в файлах представлений использовать не менее одного макета ConstraintLayout.
- 25. **Не менее одного макета** представления сформировать как для портретного, так и для альбомного режима.
- 26. Ресурсы, в том числе строковые, вынести из кода и расположить в отдельных предусмотренных для этого файлах и папках в структуре проекта.
- 27. Контроллеры представлений выполнять с помощью класса Fragment и библиотеки поддержки androidx. Разработать взаимодействие фрагментов на разных этапах жизненного цикла.
- 28. У каждой сущности должна быть форма редактирования объекта, а также форма списка объектов, построенная на базе класса RecyclerView.
- 29. Обеспечить хранение данных приложения в СУБД, поддерживающей SQL-запросы. Допустимо использование библиотек ORM для упрощения доступа к данным. Рекомендуется использование SQLite.
- 30. Реализовать панель управления в приложении, где добавить **не менее 5 пунктов меню**.
- 31. Реализовать **не менее одного диалогового окна** для получения данных от пользователя со специальной формы.
- 32. Добавить в функционал возможность вызова стандартных приложений через неявные намерения (Intent).

33. В основной сущности должен быть реализован функционал доступа к файловой системе и получение изображений с камеры.
34. Реализовать взаимодействие основной сущности с внешним веб-сервисом.

11. Требования к разрабатываемому сетевому ресурсу (веб-сайта, порталу и т.п.)

19. Сетевой ресурс должен быть разработан с применением языков программирования – PHP, JavaScript или Python с использованием фреймворков Django / Flask.
20. Обязательна загрузка в ЭУ вместе с ВКР / выгрузка в репозиторий Gitflic или Github **файлов веб-страниц в формате *.html, *.css, *.php, *.js** (при необходимости архивируется вся папка проекта, с комментариями об используемых версиях) или **файлами программного кода *.py** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях), если работа выполнялась с использованием языка программирования Python.
21. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента, происходящих в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
22. Дополнительно сетевой ресурс должен быть размещен на бесплатном хостинге.
23. При работе над сетевым ресурсом **недопустимо подменять разработку ресурса разворачиванием готового программного обеспечения** и установку готовых дополнительных программных модулей для работы с ним. Допускается разработка собственных программных модулей для работы с имеющимися программными системами и/или модификация таковых модулей. **Запрещено использование CMS и готовых плагинов к ней**. Исключением для данного пункта могут быть работы, тема которых явно указывает на данный тип деятельности.
24. При разработке сетевых ресурсов студентом должно быть предложено программное и/или аппаратное решение, готовое к внедрению в выбранный бизнес-процесс.
25. Объем программного кода должен быть **не менее 2 000 логических строк кода**, в том числе и строк, написанных с применением языка разметки. Пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
26. Разработать структуру Web-страниц. Сетевой ресурс должен содержать **не менее 10 страниц**.
27. Разработать макет каждой страницы сетевого ресурса.
28. Предусмотреть форму обратной связи.
29. Разработать **не менее 10 пунктов бокового и/или верхнего меню, «хлебные крошки»**.
30. Обеспечить хранение данных приложения в СУБД, поддерживающей SQL-запросы.
31. Предусмотреть **не менее 3-х ролей пользователей**, например, администратор, менеджер, клиент. Предоставить учетные данные всех созданных пользователей.
32. Разработать панель администратора сетевого ресурса.
33. Разработать личный кабинет пользователя сетевого ресурса.
34. Обеспечить корректное отображение сетевого ресурса в основных браузерах (Mozilla Firefox, Chrome, Яндекс.Браузер, Microsoft Edge)

35. Обеспечить корректное отображение сетевого ресурса на разных устройствах (ПК, ноутбук, смартфон)
36. Текст статей в сетевом ресурсе должен быть читабельным, грамотно сформулированным, корректным и соответствовать предметной области ВКР. Не должно быть пустых страниц в сетевом ресурсе.

12. Требования к разрабатываемому интеллектуальному сервису (с использованием машинного обучения, нейронных сетей и интеллектуального анализа данных)

14. Интеллектуальный сервис разрабатывается на языке Python с использованием сервисов Google Colab или Yandex DataSphere.
15. Обязательна загрузка в ЭУ вместе с ВКР **файла/ов программного кода *.ipynb** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях, а также с используемыми датасетами). Обязательна выгрузка в репозиторий Gitflic или Github **файла/ов программного кода *.ipynb** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях, а также с используемыми датасетами)
16. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента, происходящих в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
17. Объем собственноручно разработанного программного кода должен быть **не менее 2 000 логических строк кода**. Считаются только строки алгоритмического кода, строки кода разметки типа HTML или JSON, пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
18. Создаваемое приложение для демонстрации работы приложения обязательно должно позволять:
 - загружать и сохранять данные для обучения моделей из локальных файлов или из внешних хранилищ данных (типа kaggle или коннекторов для подключения хранилищ на облачных сервисах);
 - проводить обучение моделей, сохранять обученную модель в файл по результатам обучения;
 - загружать обученную модель из файла.
 - применять обученную или загруженную модель по назначению
 - оценивать и сохранять результаты работы модели для обучения новых версий модели
19. Дизайн приложения должен быть отражён в отчёте. Загрузка данных в приложение отражается в отчёте в **разделе загрузки данных**, обучение моделей в приложении - в **разделе обучения моделей**, процесс применения модели и оценки её результатов – в **разделе тестирования модели**.
20. В git-репозитории должен присутствовать файл **requirements.txt**, описывающий список необходимых для работы приложения и модели библиотек, которые надо

- будет установить проверяющему в пустое виртуальное окружение. Файл необходимо положить в корневой каталог репозитория.
21. В git-репозитории должен присутствовать в корневом каталоге файл **main.py**, который запускает приложение для работы с моделью.
 22. Все демонстрируемые в отчёте модели должны быть сохранены в сдаваемом git-репозитории в файле с названием **model** и расширением, которое удобно для конкретной выбранной архитектуры (например, модели на PyTorch сохраняются с расширением **.pt**, то есть будет **model.pt**). Если в отчёте демонстрируется больше 1 модели, то к названию файла добавляется порядковый номер (напр., **model1**, **model2** и т.д.). При проверке будет ожидаться, что работать эти модели будут так же, как описано в отчёте.
 23. Для всех демонстрируемых моделей, в отчёте должны быть сохранены данные для их обучения. При проверке будет ожидаться, что модель, обученная на этих данных, будет совпадать по метрикам своей эффективности с моделью в файле и описанием этих метрик в отчёте.
 24. При работе моделей, ожидается достижений уровня эффективности, сопоставимого с уровнем **Accuracy ≥ 70** . Если к конкретной модели в конкретной предметной области этот уровень не применим, то можно согласовать с руководителем работы другую метрику эффективности, отразив её в отчёте.
 25. Реализация GUI интеллектуального сервиса может быть в виде «оконных» или web форм, а также с использованием возможностей чат-ботов по отрисовке интерфейсных компонентов. Конкретный формат GUI согласовывается с руководителем ВКР.
 26. В ВКР обязательно должно быть математическое описание выбранной модели.

Методологические требования к разработке интеллектуального сервиса

16. Основной формат демонстрации навыков – **создание, обучение, тестирование и улучшение модели машинного обучения** в выбранной предметной области, сопровождаемое **визуальным анализом** получаемых результатов. Формат сохраняется не зависимо от сложности модели.
17. Создание модели предполагает, что студент умеет применять основные способы создания моделей в программном коде, как в плане написания алгоритмов «с нуля», так и в плане использование стандартных готовых библиотек – от базовых библиотек типа NumPy и Pandas, до библиотек глубокого обучения типа PyTorch и Tensorflow.
18. Создание модели предполагает, что студент владеет не только типовыми способами создания моделей, но и способен адаптировать модель под работу в конкретной предметной области, осуществляя математически грамотный подбор гиперпараметров моделей, структуру ансамблевых моделей или топологий нейронных сетей.
19. Уверенное владение моделями предполагает знакомство студентов с алгоритмами оптимизации и функциями подсчёта потерь, способность объяснить математический смысл этих алгоритмов и функций и их отношение к целевой функции, выражаемой моделью.
20. Обучение модели предполагает получение данных, на основе которых проводится обучение, подготовку данных к работе, а так же проведение самого процесса обучения модели с визуализацией промежуточных и итоговых результатов.

21. Получение данных предполагает, что студент знает откуда вообще можно брать данные для обучения моделей. Знаком как с публичными ресурсами типа kaggle, так и с подходами к самостоятельной добыче данных через web-скрейпинг, публичные API или через коннекторы для подключения хранилищ на облачных сервисах.
22. **Генерация случайных данных не допускается**, поскольку случайные данные автоматически лишают результаты работы конкретной практической ценности.
23. Получение данных предполагает, что студент понимает, какие форматы данных необходимы разным типам моделей для успешной работы в выбранной предметной области, способен привести исходные данные в нужный формат, произведя их очистку и необходимые трансформации, такие как нормализация, векторизация, приведение к тензорам, в том числе с использованием готовых конструкций загрузки данных в профессиональных библиотеках.
24. Получение данных предполагает способность студента определить, какие данные полезны, а какие нет, применяя для этого разные способы отбора и фильтрации признаков, в том числе на основе подходов визуального анализа.
25. Получение данных предполагает способность студента выделять из данных тренировочные и тестовые выборки, знать стандартные способы и инструменты такого выделения, уметь применять эти способы с учётом нужд предметной области и задачи анализа.
26. Обучение модели предполагает, что студент знает, как запустить такое обучение в рамках выбранного им способа создания модели. На минимальном уровне – умеет применять стандартные инструменты обучения моделей, представленных в рабочих библиотеках. На максимальном уровне – способен самостоятельно реализовать алгоритм обучения модели методом градиентного спуска.
27. Обучение многослойных моделей предполагает способность студента запускать автоматические фазы обучения (т.н. «эпохи»), отслеживать и визуализировать изменение эффективности модели на каждом этапе обучения.
28. По результатам обучения, в некоторых случаях - и в процессе обучения, необходимо проводить тестирование моделей. Для этого надо знать метрики оценки эффективности моделей, уметь производить тестирование и применять метрики к результатам тестирования, владеть инструментами визуализации метрик.
29. Тестирование моделей предполагает от студента способность не только произвести тестирование, но и сделать содержательные выводы из его результатов. Содержательными считаются выводы о работоспособности модели, полезности результатов обучения, может быть возможных способах улучшения модели.
30. Улучшение модели машинного обучения предполагает внесение изменений в модель с целью получить улучшенные результаты. От студента ожидается способность сформулировать предложение об улучшении модели по результатам первичного тестирования, обосновать его с помощью математического и визуального анализа, выполнить это улучшение (изменив параметры модели или пересоздав модель заново с новой топологией) и заново провести обучение и тестирование, оценив результаты улучшения (то есть фактически повторить весь путь с улучшенной моделью). Это должно быть отражено в ВКР.

Студенты должны демонстрировать все вышеперечисленные пункты по отношению к любому типу модели машинного обучения, представленному в стандартных рабочих библиотеках.

Таким образом, студент предоставляет не только работающую модель, но и приложение как среду, в которой он тестировал работу своей модели и в которой её работу

может протестировать руководитель ВКР или другой проверяющий. Отсутствие среды, в которой можно проверить работу модели, является нарушением правил сдачи работы.

13. Требования к разрабатываемому интерактивному графическому приложению (компьютерной игре)

26. Интерактивное графическое приложение разрабатывается на языке **Python** с использованием интерпретатора **PyCharm** или **Google Colab**.
27. Обязательна загрузка в ЭУ вместе с ВКР исполняемого ***.exe файла** и выгрузка в репозиторий Gitflic или Github **файлов программного кода *.py** (при необходимости архивируется вся папка проекта, с комментариями об используемых дополнительных библиотеках и версиях) если работа выполнялась с использованием языка программирования Python, либо **файла программного кода *.ipynb**, если работа выполнялась с использованием языка программирования Python и сервисов Google Colab и Yandex DataSphere.
28. Работа над проектом должна вести в git-репозитории в системе Gitflic или Github. Работа над ВКР должна представлять собой **не менее 50 коммитов от имени студента**, происходящих **в течение периодов прохождения преддипломной практики и подготовки ВКР**. Коммиты должны отражать итеративное развитие проекта, то есть не просто создание целиком готовых файлов, а их изменение по мере развития функционала проекта.
29. Объем собственноручно разработанного программного кода должен быть **не менее 2 000 логических строк кода**. Считаются только строки алгоритмического кода, строки кода разметки типа HTML или JSON, пустые строки, строки комментариев, строки только с управляющими символами подсчета не подлежат.
30. При описании внутренней архитектуры приложения, студент должен описать **не менее 10 сущностей игровой системы** (из которых **минимум 5** должны быть реализованы самостоятельно) и **не менее 5 игровых систем** (из которых **минимум 3** должны быть реализованы самостоятельно).
31. При описании интерфейса приложения, студент должен описать **не менее 4 игровых экранов и не менее 15 элементов интерфейса** на этих экранах (из которых **не менее 8** взяты не в готовом виде, а реализованы самостоятельно). Элементом интерфейса считается любой интерфейсный объект, реагирующий на события в игровой системе (например, шкала здоровья или индикатор возможности действия) или сам инициирующий событийную цепочку по факту ввода от игрока (например, кнопка).
32. Архитектура приложения обязательно должна содержать локально сохраняемые данные, такие как пользовательские настройки, сохранения, пароли и т.д.
33. Для реализации приложения можно использовать любые игровые движки и middleware, при этом имея ввиду требование о реализации достаточного объема собственного функционала поверх заимствованного.
34. В сдаваемом git-репозитории должен в корневом каталоге присутствовать **файл readme.md**, описывающий способы управления игровым процессом, а также содержащий ссылки на скачивание запускаемых сборок игрового приложения.
35. Хранение сборок в самом git-репозитории не разрешается. Можно выгрузить сборки на любой внешний ресурс для скачивания или, что предпочтительнее, осуществить релиз своего приложения на специализированной площадке типа itch.io.
36. Необходимым условием является наличие сборки, которую можно запустить под Linux, для запуска на компьютерах Университета для проверки. **Сборки только под**

Windows недостаточно. Если проект нельзя собрать для Linux или хотя бы для Web-запуска из браузера под Linux, то запускаемую сборку необходимо установить в соответствующий докер-контейнер.

Методологические требования к разработке интерактивного графического приложения

37. **Основной формат демонстрации навыков** – выполнение полного комплекса работ по разработке интерактивного графического приложения (далее «игрового приложения»), в частности: определение требований к игровому приложению, планирование отвечающего этим требованиям функционала, как в плане системной архитектуры, так и в плане графического интерфейса, с последующей реализацией этого функционала, его тестированием на предмет соответствия требованиям и улучшение приложения по результатам тестирования.
38. Формат демонстрации навыков в целом сохраняется не зависимо от природы конкретного игрового приложения. Планирование, реализация, тестирование и улучшения являются неотъемлемыми частями процесса создания любого такого приложения.
39. На этапе определения требований к игровому приложению возможна вариативность, в том числе и в плане демонстрируемых студентом навыков. Рассматривается три основных подхода к определению требований к игровому приложению: требования научной модели, требования дизайн-фокуса и требования коммерческого продукта. Эти три способа требуют демонстрирования разных наборов навыков на этапе определения требований к игровому приложению.
40. Определение требований к игровому приложению через требования научной модели используется когда целью работы ставится создание игровой модели какого-то сложного процесса. Примерами приложений, моделирующих подобные сложные процессы являются игровые приложения в стиле Microsoft Flight Simulator, Kerbal Space Program, Civilisation. Требования к игровому приложению в этом случае происходят из требований к точности реализации и визуализации работы научной модели и описанных в этой модели процессов.
41. При выборе этого способа определения требований к игровому приложению, наличие готовой научной модели, выраженной в соответствующем наборе научных публикаций, является необходимым граничным условием. Студент может взять реализацию игрового приложения на основе научной модели как задание только если предъявит руководителю научные публикации, описывающие соответствующую научную модель.
42. Если студент решает в своей работе определять требования к своему игровому приложению через призму требований научной модели, то от него будут ожидать:
 - a. Знание о научных моделях в выбранной им предметной области, описывающих моделируемый процесс. Способность адресовать к научным публикациям, обосновывать ими элементы дизайна игрового приложения.
 - b. Умение реализовывать выбранную научную модель в программном виде, как в плане внутренней рабочей логики этой модели, так и в плане визуализации её работы и демонстрации возникающей в этой модели интересных процессов.
43. Определение требований к игровому приложению на основании дизайн-фокуса предполагает наличие внешнего бизнес-процесса или группы бизнес-процессов, в

котором существует проблема. Решение этой проблемы и становится дизайн-фокусом игрового приложения. Требования к игровому приложению в этом случае определяются на основании бизнес-анализа исходной группы бизнес-процессов и определения точек применения игрового приложения для разрешения проблем в этих процессах.

44. При выборе этого способа определения требований к игровому приложению, необходимо наличие у студента доступа к специалисту, который поможет ему описать конкретные бизнес-процессы и создаст тем самым возможность для их анализа с целью последующего выделения дизайн-фокуса игрового приложения. Студент может использовать в своей работе бизнес-процессы Университета (и тогда специалистом может выступать руководитель работы) или внешней организации (и тогда специалист должен быть из внешней организации). Отсутствие доступа к специалисту делает невозможным определение требований к игровому приложению через дизайн-фокус.
45. Если студент решает в своей работе определять требования к своему игровому приложению через дизайн-фокус, то от него будут ожидать:
 - a. Навыки анализа бизнес-процессов и выделения в них тех точек, которые могут быть улучшены с помощью игрового приложения.
 - b. Знание технологий и подходов геймификации бизнес-процессов и способность применить эти знания для предложения улучшения в бизнес-процессах за счёт игрового приложения.
46. Определение требований к игровому приложению на основании требований коммерческого продукта предполагает, что студент в состоянии определить провести маркетинговое профилирование целевой аудитории продукта, может сформулировать потребности этой целевой аудитории и тем самым обосновать требования к игровому приложению как потребительским характеристикам, интересующих целевую аудиторию.

Если студент решает в своей работе определять требования к своему игровому приложению как к коммерческому продукту, то от него будут ожидать:

- d. Навыки поиска других продуктов в выбранной им категории игровых приложений, определения тех их ключевых качеств, на которые реагирует аудитория.
 - e. Способность профилирования аудитории выбранной категории игровых приложений с точки зрения интересующего конкретных представителей конкретного профиля набора потребительских качеств.
 - f. Способность выделить целевую аудиторию для своего продукта и спланировать потребительские качества собственного продукта, соответствующие маркетинговому профилю целевой аудитории.
47. Сформулировав требования к своему игровому приложению одним из трёх вышеописанных способов, студент переходит к планированию реализации игрового приложения. Необходимо спланировать первую версию приложения, удовлетворяющего заданным требованиям, как в плане программной архитектуры, так и в плане интерфейсов и игровых взаимодействий.
48. На этапе планирования от студента ожидаются:
 - a. Знания о типовых способах составления проектной документации для игрового приложения (далее «дизайн-документа»).
 - b. Умение правильно структурировать дизайн-документ, отразив в нём как требования к игровому продукту, так и способы их реализации.

- с. Владение способами визуализации схем игровых взаимодействий и необходимых алгоритмов, видов игровых объектов и отношений между ними, способов управления игровым процессом.
49. Спланировав первую версию приложения, студент переходит к его реализации. На этом этапе, от студента ожидаются:
- d. Знания общих архитектурных схем, стандартных алгоритмов и типовых структур данных, позволяющих реализовать запланированное в дизайн-документе.
 - e. Умение преобразовывать требования продукта (будь то потребительские качества, требования научной модели или схемы геймификации) в конкретный функционал игрового приложения.
 - f. Владение инструментарием создания игровых приложений, будь то промежуточные библиотеки типа rpgame или Ren'Py, или интегрированные решения типа Unity.
50. Реализовав первую версию приложения, студент должен протестировать её относительно требований и сделать выводы о необходимых улучшениях. Конкретный характер тестов будет зависеть от способа формирования требований.
- a. В случае научной модели, необходимо реализовать архитектуру автоматического тестирования корректного выполнения научной модели во всех игровых сценариях, показать работу этой архитектуры.
 - b. В случае коммерческого продукта, необходимо предложить игровой продукт целевой аудитории, собрать с неё обратную связь. После чего на основании этой обратной связи сделать выводы о том, насколько продукт отвечает потребностям аудитории в рамках сформулированного профиля, предложить и реализовать улучшения продукта.
 - с. В случае дизайн-фокуса, необходимо согласовать конкретную схему проверки со специалистом, владеющим улучшаемым с помощью игрового приложения бизнес-процессом. Согласованную схему проверки будет необходимо выполнить, отразить в отчёте работы и после чего, на основании результатов, сделать выводы о работоспособности предложенной геймификации и внести улучшения.

В итоге, от студентов ожидается внятное объяснение требований, которые они сформулировали к реализуемому игровому приложению, способов выполнения этих требований, способов верификации выполнения этих требований, а также улучшений, которые были сделаны в тех местах, где требования с первого раза не выполнялись.

14. Требования к тестированию программного продукта

6. Для разрабатываемого приложения необходимо выбрать **минимум 4 метода тестирования** из нижеприведенных видов тестирования:

№	Вид тестирования	Метод тестирования	Описание метода
1	По принципам работы с приложением	Позитивное тестирование	тестирование, при котором используются только корректные данные.
		Негативное тестирование	тестирование приложения, при котором используются некорректные данные и выполняются некорректные операции

2	По уровню функционального тестирования	Смоук-тестирование (дымовое, smoke test)	тестирование, выполняемое на новой сборке, с целью подтверждения того, что программное обеспечение стартует и выполняет основные для бизнеса функции.
		Тестирование критического пути (critical path)	направлено для проверки функциональности, используемой обычными пользователями во время их повседневной деятельности.
		Расширенное тестирование (extended)	направлено на исследование всей заявленной в требованиях функциональности.
3	В зависимости от исполнителей	Альфа-тестирование	является ранней версией программного продукта. Может выполняться внутри организации-разработчика с возможным частичным привлечением конечных пользователей.
		Бета-тестирование	программное обеспечение, выпускаемое для ограниченного количества пользователей. Главная цель — получить отзывы клиентов о продукте и внести соответствующие изменения.
4	В зависимости от целей тестирования	Функциональное тестирование (functional testing)	направлено на проверку корректности работы функциональности приложения.
		Нефункциональное тестирование (non-functional testing):	тестирование атрибутов компонента или системы, не относящихся к функциональности.
		– Тестирование производительности (performance testing)	определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок.
		– Нагрузочное тестирование (load testing)	определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).
		– Тестирование масштабируемости (scalability testing)	тестирование, которое измеряет производительность сети или системы, когда количество пользовательских запросов увеличивается или уменьшается.
		– Объемное тестирование (volume testing)	это тип тестирования программного обеспечения, которое проводится для тестирования программного приложения с определенным объемом данных.
		– Стрессовое тестирование (stress testing)	тип тестирования направленный для проверки, как система обращается с нарастающей нагрузкой (количеством одновременных пользователей).
		– Инсталляционное тестирование (installation testing)	тестирование, направленное на проверку успешной установки и настройки, обновления или удаления приложения.
		– Тестирование интерфейса (GUI/UI testing)	проверка требований к пользовательскому интерфейсу.

		– Тестирование удобства использования (usability testing)	это метод тестирования, направленный на установление степени удобства использования, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.
		– Тестирование локализации (localization testing)	проверка адаптации программного обеспечения для определенной аудитории в соответствии с ее культурными особенностями.
		– Тестирование безопасности (security testing)	это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.
		– Тестирование надежности (reliability testing)	один из видов нефункционального тестирования ПО, целью которого является проверка работоспособности приложения при длительном тестировании с ожидаемым уровнем нагрузки.
		– Регрессионное тестирование (regression testing)	тестирование уже проверенной ранее функциональности после внесения изменений в код приложения, для уверенности в том, что эти изменения не внесли ошибки в областях, которые не подверглись изменениям.
		– Повторное/подтверждающее тестирование (re-testing/confirmation testing)	тестирование, во время которого исполняются тестовые сценарии, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок.
5	По запуску кода на исполнение	Статическое	процесс тестирования, который проводится для верификации практически любого артефакта разработки: программного кода компонент, требований, системных спецификаций, функциональных спецификаций, документов проектирования и архитектуры программных систем и их компонентов.
		Динамическое	тестирование проводится на работающей системе, не может быть осуществлено без запуска программного кода приложения.
6	По доступу к коду и архитектуре	Метод белого ящика	метод тестирования ПО, который предполагает полный доступ к коду проекта.
		Сетод серого ящика	метод тестирования ПО, который предполагает частичный доступ к коду проекта (комбинация White Box и Black Box методов).
		Метод черного ящика	метод тестирования ПО, который не предполагает доступа (полного или частичного) к системе. Основывается

			на работе исключительно с внешним интерфейсом тестируемой системы.
7	По уровню детализации приложения	Модульное тестирование	проводится для тестирования какого-либо одного логически выделенного и изолированного элемента (модуля) системы в коде. Проводится самими разработчиками, так как предполагает полный доступ к коду.
		Интеграционное тестирование	тестирование, направленное на проверку корректности взаимодействия нескольких модулей, объединенных в единое целое.
		Системное тестирование	процесс тестирования системы, на котором проводится не только функциональное тестирование, но и оценка характеристик качества системы — ее устойчивости, надежности, безопасности и производительности.
		Приемочное тестирование	проверяет соответствие системы потребностям, требованиям и бизнес-процессам пользователя.
8	По степени автоматизации	Ручное тестирование	
		Автоматическое тестирование	

7. **Сформировать тест план.** Тест план должен отвечать на следующие вопросы: Что необходимо протестировать? Как будет проводиться тестирование? Когда будет проводиться тестирование? Критерии начала тестирования. Критерии окончания тестирования.

Основные пункты тест плана:

- Идентификатор тест плана (Test plan identifier);
 - Введение (Introduction);
 - Объект тестирования (Test items);
 - Функции, которые будут протестированы (Features to be tested);
 - Функции, которые не будут протестированы (Features not to be tested);
 - Тестовые подходы (Approach);
 - Критерии прохождения тестирования (Item pass/fail criteria);
 - Критерии приостановления и возобновления тестирования (Suspension criteria and resumption requirements);
 - Результаты тестирования (Test deliverables);
 - Задачи тестирования (Testing tasks);
 - Ресурсы системы (Environmental needs);
 - Обязанности (Responsibilities);
 - Роли и ответственность (Staffing and training needs);
 - Расписание (Schedule);
 - Оценка рисков (Risks and contingencies);
 - Согласования (Approvals).
8. **Сформировать тест-кейсы** (тестовые сценарии). Обязательные атрибуты тест кейса:
- **Предусловия** (PreConditions) — список действий, которые приводят систему к состоянию пригодному для проведения основной проверки. Либо список условий, выполнение которых говорит о том, что система находится в пригодном для проведения основного теста состоянии.

- **Шаги (Steps)** — список действий, переводящих систему из одного состояния в другое, для получения результата, на основании которого можно сделать вывод о удовлетворении реализации, поставленным требованиям.
 - **Ожидаемый результат (Expected result)** — что по факту должны получить.
9. На все найденные ошибки **сформировать баг-репорт**. Баг-репорт должен обязательно содержать атрибуты: приоритет и серьезность.
- **Серьезность** показывает, насколько баг влияет на возможность работать в программе. Виды:

Серьезность бага	Как влияет на систему
S1 — Блокирующая (Blocker).	Приложение не запускается ни на одном устройстве или выдаёт ошибку при загрузке. <i>Пример: приложение не открывается, поэтому невозможно сделать покупку.</i>
S2 — Критическая (Critical).	Часть функционала не работает, но это не блокирует процесс. <i>Пример: Скидка по купону не работает, но товар всё равно можно купить без скидки.</i>
S3 — Значительная (Major).	Часть логики работает некорректно, но пользователь может решить проблему другим способом. <i>Пример: поиск в приложении не выдаёт нужный товар, но его можно найти, если искать по брендам.</i>
S4 — Незначительная (Minor).	Не нарушает логику приложения. <i>Пример: кнопка «Купить» уехала за экран и видна только наполовину.</i>
S5 — Тривиальная (Trivial).	Не относится к логике приложения и не влияет на общее качество продукта. <i>Пример: малозаметная опечатка в меню.</i>

Приоритет — это критерий, который показывает, насколько быстро нужно исправить дефект. Приоритет может быть:

- высокий — исправить в первую очередь;
- средний — исправить, когда разобрались с первой категорией багов;
- низкий — исправить, когда разобрались с багами других приоритетов.

10. Произвести отладку разработанной системы.



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ С.Ю. ВИТТЕ

Факультет Информационных технологий

Кафедра Кафедра информационных систем

Направление подготовки/Специальность Бизнес-информатика

ОТЧЕТ

о прохождении

Производственная практика

/вид практики/

Преддипломная практика

/тип практики/

Студента Ф.И.О. обучающегося
(фамилия, имя, отчество)

Место прохождения практики ЧОУВО "Московский университет имени С.Ю. Витте"

Период прохождения практики с 10.05.2021 г. по 23.05.2021 г.

г.Москва 2021 г.

Примерная структура отчета о прохождении практики

Титульный лист

Содержание

ВВЕДЕНИЕ

ОСНОВНАЯ ЧАСТЬ

1 АНАЛИЗ ОРГАНИЗАЦИИ

- 1.1 Исходные данные для отчета
- 1.2 Анализ структуры и нормативной документации, регламентов подразделения университета и регулирующих выполнение бизнес-процесса
- 1.3 Обоснование выбора бизнес-процесса для автоматизации (модель AS IS)
- 1.4 Анализ требований пользователей к разрабатываемой корпоративной информационной системе
- 1.5 Анализ ИТ-инфраструктуры выбранного структурного подразделения для решения задачи разработки корпоративной информационной системы
 - 1.5.1 Анализ материально-технического обеспечения
 - 1.5.2 Анализ программного обеспечения
- 1.6 Перечень служебных поручений и задач при прохождении производственной практики
- 1.7 Техническое задание на разработку корпоративной информационной системы
 - 1.7.1 Общие сведения
 - 1.7.2 Цели и назначение создания автоматизированной системы
 - 1.7.3 Характеристика объектов автоматизации
 - 1.7.4 Требования к автоматизированной системе
 - 1.7.5 Состав и содержание работ по созданию автоматизированной системы
 - 1.7.6 Порядок разработки автоматизированной системы
 - 1.7.7 Порядок контроля и приемки автоматизированной системы
 - 1.7.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие
 - 1.7.9 Требования к документированию
 - 1.7.10 Источники разработки
- 1.8 Выводы по разделу

2 ПРОЕКТИРОВАНИЕ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

- 2.1 Разработка предложений по оптимизации выбранного бизнес-процесса (модель TO BE)
- 2.2 Разработка предложений по архитектуре корпоративной информационной системы
- 2.3 Составление плана разработки
- 2.4 Настройка репозитория
- 2.5 Проектирование базы данных
 - 2.5.1 Инфологическое проектирование БД (ER-диаграмма)
 - 2.5.2 Логическое проектирование БД (Уточненная ER-диаграмма)
 - 2.5.3 Разработка схемы данных
 - 2.5.4 Физическое проектирование БД
- 2.6 Разработка интерфейса корпоративной информационной системы
 - 2.6.1 Разработка прототипа корпоративной ИС
 - 2.6.2 Разработка прототип корпоративной ИС
- 2.7 Разработка корпоративной информационной системы

2.8 Тестирование и отладка разработанного ПО

2.8.1 Тестирование по методу 1

2.8.2 Тестирование по методу 2

2.8.3 Тестирование по методу 3

2.8.4 Тестирование по методу 4

2.8.5 Отладка

2.9 Выводы по разделу

3 РАЗРАБОТКА МОДЕЛИ РАЗВЕРТЫВАНИЯ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Составление плана развертывания ИС

3.2 Разработка плана интеграции корпоративной ИС с существующими ИС у заказчика

3.3 Формирование модели обновлений и технической поддержки

3.4 Выводы по разделу

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

ОЦЕНОЧНЫЕ СРЕДСТВА
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ
/вид практики/

Преддипломная практика
/тип практики/

Направление подготовки:
38.03.05 Бизнес-информатика

Направленность (профиль) подготовки:
Цифровая экономика

уровень высшего образования
Бакалавриат

2022 г.

1. Общие положения

Оценочные средства для проведения текущего контроля и промежуточной аттестации по практике включают:

- перечень формируемых компетенций в результате прохождения практики, характеризующих результаты освоения образовательной программы;
- описание показателей и критериев оценивания компетенций, описание шкал оценивания;
- процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие формирование компетенций.

Текущий контроль производственной практики проводится в форме собеседования с руководителями производственной практики от Университета и от предприятия – базы практики в ходе составления, согласования и выполнения индивидуального задания на практику обучающегося, а также в процессе составления отчета о прохождении практики.

Промежуточная аттестация по производственной практике осуществляется в форме зачета с оценкой (далее – зачета).

Для получения зачета обучающийся представляет в электронном виде все необходимые документы по практике, отчет и разработанное ПО, которые выполняются по результатам прохождения практики с учетом результатов проведенных работ.

Страницы документов, на которых должны быть проставлены подписи и печать, представляются в отсканированном виде.

Все документы по практике размещаются в личном кабинете обучающегося через Электронный университет.

2. Перечень формируемых компетенций в результате прохождения практики, характеризующих результаты освоения образовательной программы

Основными этапами формирования компетенций при прохождении производственной практики являются последовательное прохождение содержательно связанных между собой разделов практики. Изучение каждого раздела предполагает овладение обучающимися необходимыми элементами компетенций на уровне знаний, навыков и умений. Итоговый результат, полученный с учетом оценивания формирования компетенций, показывает успешность освоения обучающимися образовательной программы.

При выставлении оценки учитывается алгоритмическая реализация поставленных задач, работоспособность программного кода и используемые библиотеки, также учитывается качество представленных практикантом материалов в период практики.

Планируемые результаты практики

Коды компетенций	Содержание компетенции
ПК-1	Способность создавать, редактировать и наполнять контентом информационные ресурсы
ПК-2	Способность разработки прикладного программного обеспечения для управления бизнесом
ПК-3	Способность выполнять анализ бизнес-информации
ПК-4	Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами
ПК-7	Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования

Планируемые результаты практики, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование профессиональной компетенции выпускника программы бакалавриата	Код и наименование индикатора достижения профессиональной компетенции
ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы	ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента.
ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом	ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владет навыками автоматизации решения типовых задач для управления бизнесом.
ПК-3. Способность выполнять анализ бизнес-информации	ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2. Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации.
ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
ПК-7. Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования	ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владеть методами разработки дизайна ИТ сервисов

3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Промежуточная аттестация по практике проводится с целью выявления соответствия уровня теоретических знаний, практических умений и навыков в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 «Прикладная информатика» в форме зачета с оценкой.

Оценка по результатам зачета по практике выставляется в ведомость, электронную зачетную книжку обучающегося, а в последующем в приложение к диплому.

Оценки выставляются по столбальной шкале, с последующим переводом в четырехбалльную следующим образом:

- 100-85 баллов – зачтено с оценкой отлично;
- 84-66 баллов – зачтено с оценкой хорошо;
- 65-50 баллов – зачтено с оценкой удовлетворительно;
- 49 и менее – не зачтено с оценкой неудовлетворительно.

Шкалы и показатели оценивания представлены в таблице 2.

Таблица 2 - Шкалы оценивания/показатели оценивания

Показатели оценивания	«Зачтено (85-100)», «отлично»	«Зачтено (66-84)» «хорошо»	«Зачтено (50-65)» «удовлетворительно»	«Не зачтено (менее 50)» «неудовлетворительно»
Оценивание выполнения разделов 1,2 разделов индивидуального задания	- проведено обследование организации и области ее деятельности - формализованы требования пользователей к информационно-коммуникационной среде - проведено обоснование выбора программной платформы и необходимых библиотеки для решения заданий - разработано техническое задание по ГОСТ 19.201-78 - приведено технико-экономическое обоснование - выполнен функциональный анализ бизнес-процессов организации - разработана, модифицирована или внедрена система для автоматизации бизнес-процессов - составлен план развертывания и внедрения системы - составлена техническая документация - своевременно, качественно выполнил весь объем работы, требуемый программой практики; - показал глубокую теоретическую, методическую подготовку, демонстрирует достаточно полные знания всех профессионально-прикладных и методических вопросов в объеме программы практики.	- в проведенном обследовании организации учтены не все области ее деятельности, необходимы для выполнения работы по практике - анализ требований пользователей проведен, но нет формализации требования пользователей к информационно-коммуникационной среде - слабое обоснование выбора программной платформы и необходимых библиотеки для решения заданий - разработано техническое задание на разработку сайта с небольшими отклонениями от ГОСТ 19.201-78 - приведено технико-экономическое обоснование - выполнен функциональный анализ бизнес-процессов организации - разработана, модифицирована или внедрена система для автоматизации бизнес-процессов - составлен план развертывания и внедрения системы - составлена техническая документация - своевременно, качественно выполнил весь объем работы, требуемый программой практики; - показал необходимую для прохождения практики теоретическую и методическую подготовку, демонстрирует знания большинства профессионально-	- не проведены необходимые анализы данных/бизнес-процессов и обоснование - разработанное техническое задание на разработку сайта не соответствует согласно ГОСТ 19.201-78 - не приведено технико-экономическое обоснование - разработанная, модифицированная или внедренная система автоматизирует не все описанные в отчете бизнес-процессов - составлен план развертывания и внедрения системы - составлена слабая техническая документация - объем работы, требуемый программой практики, выполнен с задержкой - план развертывания и внедрения системы составлен некорректно - имеет слабую, но минимально достаточную теоретическую и методическую подготовку	- владеет фрагментарными знаниями и не умеет применить их на практике, не способен самостоятельно продемонстрировать работу программного обеспечения; - не выполнил программу практики в полном объеме - не приведено технико-экономическое обоснование - не выполнен функциональный анализ бизнес-процессов организации - не составлен план развертывания и внедрения системы - не составлена техническая документация - представленные результаты выполнения индивидуального задания в части программной разработки не формируют требуемых компетенций.

		прикладных и методических вопросов в объеме программы практики.		
Оценивание содержания и оформления отчета по практике (раздел 1-3 индивидуального задания)	Отчет по практике выполнен в полном объеме и в соответствии с требованиями. Результативность практики представлена в виде работающего приложения. Студент соотносит выполненные задания с формированием компетенций. Грамотно использует профессиональную терминологию при оформлении отчетной документации по практике. Программный код содержит подробные комментарии. Представлено описание всех сторонних библиотек, используемых при выполнении индивидуального задания. Полностью соответствует ГОСТ Р 7.0.100-2018 и требованиям ВУЗа	Отчет по практике выполнен в полном объеме и в соответствии с ГОСТ Р 7.0.100-2018 и требованиями ВУЗа. Студент соотносит выполненные задания с формированием компетенций. В целом владеет профессиональной терминологией при оформлении отчетной документации по практике. Программный код содержит недостаточно комментариев.	Отчет по практике выполнен в полном объеме, с незначительными отклонениями от ГОСТ Р 7.0.100-2018 и требований ВУЗа. Представленные отчетные документы, результаты выполнения индивидуального задания не формируют требуемых компетенций. Отчет носит описательный характер, в отчете нет подробных пояснений к программному коду. Затрудняется при использовании профессиональной терминологии.	Документы по практике не оформлены в соответствии с требованиями программы практики. Представленные отчетные документы, результаты выполнения индивидуального задания не формируют требуемых компетенций.

3. Методические материалы, определяющие процедуры оценивания знаний, умений и навыков.

Процедура проведения промежуточной аттестации происходит в 4 этапа:

1. Проверка отчета по практике в ЭИОС Электронный университет.
2. Проверка работоспособности программного решения (игрового ПО).
3. Опрос обучающегося по программному коду и функционалу разработанного программного решения (для обучающихся с применением дистанционных технологий обучения, возможно проведение опроса посредством оставления комментария к отчетной документации в ЭИОС Электронный университет).
4. Проверка наличия всей корректно оформленной отчетной документации в соответствующих разделах ЭИОС Электронный университет.

Вопросы (задания) для оценки знаний после прохождения практики зависят от содержания практики обучающегося.

Перечень контролируемых вопросов и компетенций (индикаторов достижения компетенций), сформированных в процессе прохождения практики:

Задания, раскрывающие уровень освоения компетенций	компетенции
Дайте характеристику изученной вами внутренней распорядительной документации, а также стандартов и регламентов в области информационных технологий. Какими документами вы руководствовались при разработке технической документации?	ПК-3, ПК-4
Какие информационные технологии были использованы вами в процессе автоматизации бизнес-процесса? Опишите порядок работ по наполнению контентом информационных ресурсов предприятия.	ПК-1, ПК-2, ПК-7

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Общие методические подходы оценивания результатов промежуточной аттестации установлены в Положении о текущем контроле успеваемости и промежуточной аттестации обучающихся, утвержденном ректором Университета.

Основные методические положения по прохождению практики в Университете установлены Порядком организации и проведения практики обучающихся, осваивающих основные профессиональные образовательные программы высшего образования, утвержденном ректором Университета.

Приложение 5
ПРИМЕР оформления отчета по практике



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ С.Ю. ВИТТЕ

Факультет Информационных технологий

Кафедра Кафедра информационных систем

Направление подготовки/Специальность Бизнес-информатика

ОТЧЕТ

о прохождении

Производственная практика

/вид практики/

Преддипломная практика

/тип практики/

Студента Ф.И.О. обучающегося
(фамилия, имя, отчество)

Место прохождения практики ЧОУВО "Московский университет имени С.Ю. Витте"

Период прохождения практики с 10.05.2021 г. по 23.05.2021 г.

г.Москва 2021 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ		63
ОСНОВНАЯ ЧАСТЬ		65
1 АНАЛИЗ ПРЕДПРИЯТИЯ / ОРГАНИЗАЦИИ / ЧОУ ВО «МОСКОВСКИЙ УНИВЕРСИТЕТ ИМ. С.Ю. ВИТТЕ» ЭТО ПРИМЕР НАЗВАНИЯ РАЗДЕЛА. НЕ НУЖНО ВСЕ КОПИРОВАТЬ. НАЗВАНИЕ РАЗДЕЛА ДОЛЖНО ОТРАЖАТЬ СОДЕРЖАТЕЛЬНУЮ ЧАСТЬ РАЗДЕЛА		65
1.1	Исходные данные для отчета	65
1.2	Анализ структуры и нормативной документации, регламентов подразделения «xxxxxxxxxxxx» университета и регулирующих выполнение бизнес-процесса	65
1.3	Обоснование выбора бизнес-процесса для автоматизации (модель AS IS)	66
1.4	Анализ требований пользователей (руководства, заказчика, если проектируется сайт, мобильное приложение и т.п.) к разрабатываемой корпоративной информационной системе xxx xxx xxxxx ЧОУ ВО «Московский университет им. С.Ю. Витте» Это один из вариантов названия раздела. Не нужно всё копировать	72
1.5	Анализ ИТ-инфраструктуры выбранного структурного подразделения для решения задачи разработки корпоративной информационной системы «Хxxxx xxx х xxxxxxxx» (для решения задачи сетевого администрирования xxxxxxxx, для решения задачи модели автоматизации, для решения задачи разработки модуля корпоративной....) Это один из вариантов названия раздела. Не нужно всё копировать	73
1.5.1	Анализ материально-технического обеспечения	74
1.5.2	Анализ программного обеспечения	74
1.6	Перечень служебных поручений и задач при прохождении производственной практики	75
1.7	Техническое задание на разработку / модификацию / развертывание составляющих компонентов корпоративной информационной системы xxxxxx Это один из вариантов названия раздела. Не нужно всё копировать	75
1.7.1	Общие сведения	76
1.7.2	Цели и назначение создания автоматизированной системы	77
1.7.3	Характеристика объектов автоматизации	77
1.7.4	Требования к автоматизированной системе	78
1.7.5	Состав и содержание работ по созданию автоматизированной системы	78
1.7.6	Порядок разработки автоматизированной системы	78
1.7.7	Порядок контроля и приемки автоматизированной системы	79
1.7.8	Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие	79
1.7.9	Требования к документированию	80
1.7.10	Источники разработки	80
1.8	Выводы по разделу	81
2 ПРОЕКТИРОВАНИЕ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ / ПРОГРАММНОГО МОДУЛЯ / БИБЛИОТЕКИ И Т.П. ПРОЕКТА АВТОМАТИЗАЦИИ ПРИКЛАДНОГО ПРОЦЕССА (МОДИФИКАЦИИ ПРИЛОЖЕНИЯ, СОЗДАНИЯ/АДАПТАЦИИ САЙТА, РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ, РАЗВЕРТЫВАНИЯ ИТ-ИНФРАСТРУКТУРЫ: ПРОГРАММНОЙ/АППАРАТНОЙ И Т.П.) ЭТО ПРИМЕР НАЗВАНИЯ РАЗДЕЛА. НЕ НУЖНО ВСЕ КОПИРОВАТЬ. НАЗВАНИЕ РАЗДЕЛА ДОЛЖНО ОТРАЖАТЬ СОДЕРЖАТЕЛЬНУЮ ЧАСТЬ РАЗДЕЛА		83
2.1	Разработка предложений по оптимизации выбранного бизнес-процесса (модель TO BE)	83

2.2	Разработка предложений по архитектуре корпоративной информационной системы / модуля корпоративной ИС, обеспечивающей автоматизацию процесса xxxx xxxxxxxx в ЧОУ ВО «Московский университет им. С.Ю. Витте» Это один из вариантов названия раздела. Не нужно всё копировать.....	83
2.3	Составление плана разработки	84
2.4	Настройка репозитория	84
2.5	Проектирование базы данных.....	85
2.5.1	Инфологическое проектирование БД (ER-диаграмма)	85
2.5.2	Логическое проектирование БД (Уточненная ER-диаграмма)	87
2.5.3	Разработка схемы данных	89
2.5.4	Физическое проектирование БД	90
2.6	Разработка интерфейса корпоративной информационной системы /сайта/информ.портала/ мобильного приложения) Это один из вариантов названия раздела. Не нужно всё копировать	98
2.6.1	Разработка прототипа корпоративной ИС / программного модуля / библиотеки / дополнения / интерфейса и т.п	98
2.6.2	Разработка макета корпоративной ИС / программного модуля / библиотеки / дополнения / и т.п	99
2.7	Разработка/модификация/разработка программного модуля/приложения (участие в разработке прикладной библиотеки, разработка локальных политик сетевой безопасности и т.п.) – КОНЕЧНЫЙ ВАРИАНТ НАЗВАНИЯ ИС ПОД ВКР	100
2.8	Тестирование и отладка разработанного ПО	101
2.8.1	Тестирование по методу XXXXXXXXXXXX	101
2.8.2	Тестирование по методу 2 XXXXXXXXXXXX	105
2.8.3	Тестирование по методу 3 XXXXXXXXXXXX	106
2.8.4	Тестирование по методу 4 XXXXXXXXXXXX	107
2.8.5	Отладка	107
2.9	Выводы по разделу	108
3	РАЗРАБОТКА МОДЕЛИ РАЗВЕРТЫВАНИЯ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ «XXXXX XXXX XX» (ПРОГРАММНОГО МОДУЛЯ, БИБЛИОТЕКИ, РНР КОДА, АДМИНИСТРИРОВАНИЯ...) ЭТО ПРИМЕР НАЗВАНИЯ РАЗДЕЛА. НЕ НУЖНО ВСЕ КОПИРОВАТЬ. НАЗВАНИЕ РАЗДЕЛА ДОЛЖНО ОТРАЖАТЬ СОДЕРЖАТЕЛЬНУЮ ЧАСТЬ РАЗДЕЛА.....	110
3.1	Составление плана развертывания ИС	110
3.2	Разработка плана интеграции корпоративной ИС xxxx xxxxxxxxxx с существующими ИС у заказчика.....	111
3.3	Формирование модели обновлений и технической поддержки	114
3.4	Выводы по разделу	117
	ЗАКЛЮЧЕНИЕ.....	119
	СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ	128

ВВЕДЕНИЕ

ЭТО ПРИМЕР ВВЕДЕНИЯ Основная цель преддипломной практики состоит в сборе фактического материала для выполнения выпускной квалификационной работы бакалавра. Прохождение преддипломной практики на кафедре информационных систем (либо в другом структурном подразделении университета) позволяет получить ценный практический опыт по проектированию корпоративных информационных систем с точки зрения автоматизации отдельных бизнес-процессов структурного подразделения, выявления недостатков и последующих предложений по их устранению. В большинстве случаев модернизация может заключаться в структурной оптимизации архитектуры ИС или разработке дополнительного функционала для автоматизации бизнес-процессов структурного подразделения университета. В обоих случаях решение задачи состоит в предварительном анализе и, на основе его, последующей разработке эффективных решений.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx. Xxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxx.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxx xxxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx. Xxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

Решение первой задачи позволит найти неэффективные архитектурные схемы реализации отдельных модулей корпоративной информационной системы. Кроме того, это даст возможность разработать модульную структуру программного приложения, автоматизирующего решение задачи xxxx xxx xxxx xxxxxx, а также разработать алгоритмы их функционирования. Хxxxxxx xxxx xxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxxx xxxxx. Xxx xx xx xxxxxxx xxxxxxx xxxxxxx. Хxxxx xxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxx xxxxxxx xxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx xxxxx. Хxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx.

Цель практики

получение практического опыта использования информационных технологий для решения прикладных задач, а также на выработку практических навыков по их анализу, выбору и применению информационных технологий в ЧОУ ВО «Московский университет им. С.Ю. Витте».

Задачи практики

- **ЭТО ПРИМЕР ФОРМУЛИРОВКИ ЗАДАЧ**
- **НЕ СЛЕДУЕТ КОПИРОВАТЬ ВСЕ ПОДРЯД**
- Закрепление приобретенных теоретических знаний.
- Изучение опыта создания и применения информационных технологий в структурных подразделениях ЧОУ ВО «Московский университет им. С.Ю. Витте»
- Использование нормативно-правовой документации при решении прикладных задач
- Составление требований к информационной системе при анализе заявок от пользователей информационной системы в ЧОУ ВО «Московский университет им. С.Ю. Витте»
- Разработка и адаптации модулей прикладного программного обеспечения ИТ-инфраструктуры в ЧОУ ВО «Московский университет им. С.Ю. Витте»
- Составление технического задания на создание компонентов автоматизированной системы
- Анализ собранной информации от пользователей и формирования требований к информационно-коммуникационной среде в ЧОУ ВО «Московский университет им. С.Ю. Витте»
- Приобретение навыков практического решения задач на конкретных рабочих местах в качестве исполнителей или стажеров.
- Получение навыков самостоятельной работы
- Формирование навыков выполнения трудовых действий в соответствие с профессиональными стандартами
- Подготовка отчетов и презентаций по результатам профессиональной деятельности

Источниками информации явились организационно-правовые документы, распорядительные и информационно-справочные документы, используемые в ЧОУ ВО «Московский университет им. С.Ю. Витте».

1. АНАЛИЗ ПРЕДПРИЯТИЯ / ОРГАНИЗАЦИИ / ЧОУ ВО «МОСКОВСКИЙ УНИВЕРСИТЕТ ИМ. С.Ю. ВИТТЕ» ЭТО ПРИМЕР НАЗВАНИЯ РАЗДЕЛА. НЕ НУЖНО ВСЕ КОПИРОВАТЬ. НАЗВАНИЕ РАЗДЕЛА ДОЛЖНО ОТРАЖАТЬ СОДЕРЖАТЕЛЬНУЮ ЧАСТЬ РАЗДЕЛА

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxxxxx xxxxxx. Xxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

1.3. Обоснование выбора бизнес-процесса для автоматизации (модель AS IS)

- IDEF0 (с декомпозицией)
- DFD (возможно использование нотации Гейна-Сарсона или Йордона-Де Марко)
- Диаграмма активностей (Activity diagram, диаграмма деятельности) в нотации UML, BPMN, EPC или IDEF3



Xxxxx xxxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxx xxx xxxxx. Xxxx xxxxx xxxxx
 xxxxxxxx xxxxx xxxxx xx xxxxxxx xxxxxxxx xx x x xxxxxxxx. Xxx xxx xxxxxxx xxxxxxx xxxxxxx
 xxxxxxx xxxxx xxxxxxxxxxx xxxxxxx xxxxxxxxxx xxxxxxx. Xxxxx xxxxxxx xxxxxxx xxxxxxx
 xxxxxxx xxxxxxx xxxxx xxx xxxxx. XXXXXXXXXXX xxxxx xxxxx xxxxxxx xxxxx xxxxxxx xxxxxxx
 xxxxxxx xxx xxx x xxxxx xx. Xxxx xxxxx xxxxx xxxxxxxxxx xxxxx xxxxx xx xxxxxxx xxxxxxxx xx x
 x xxxxxxx. Xxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxx xxx xxxxx. Xxxxx xxxxxxx
 xxxxxxx xxxxxxx xxxxxxxxxx xxxxxxx xxxxx xxx xxxxx.

68

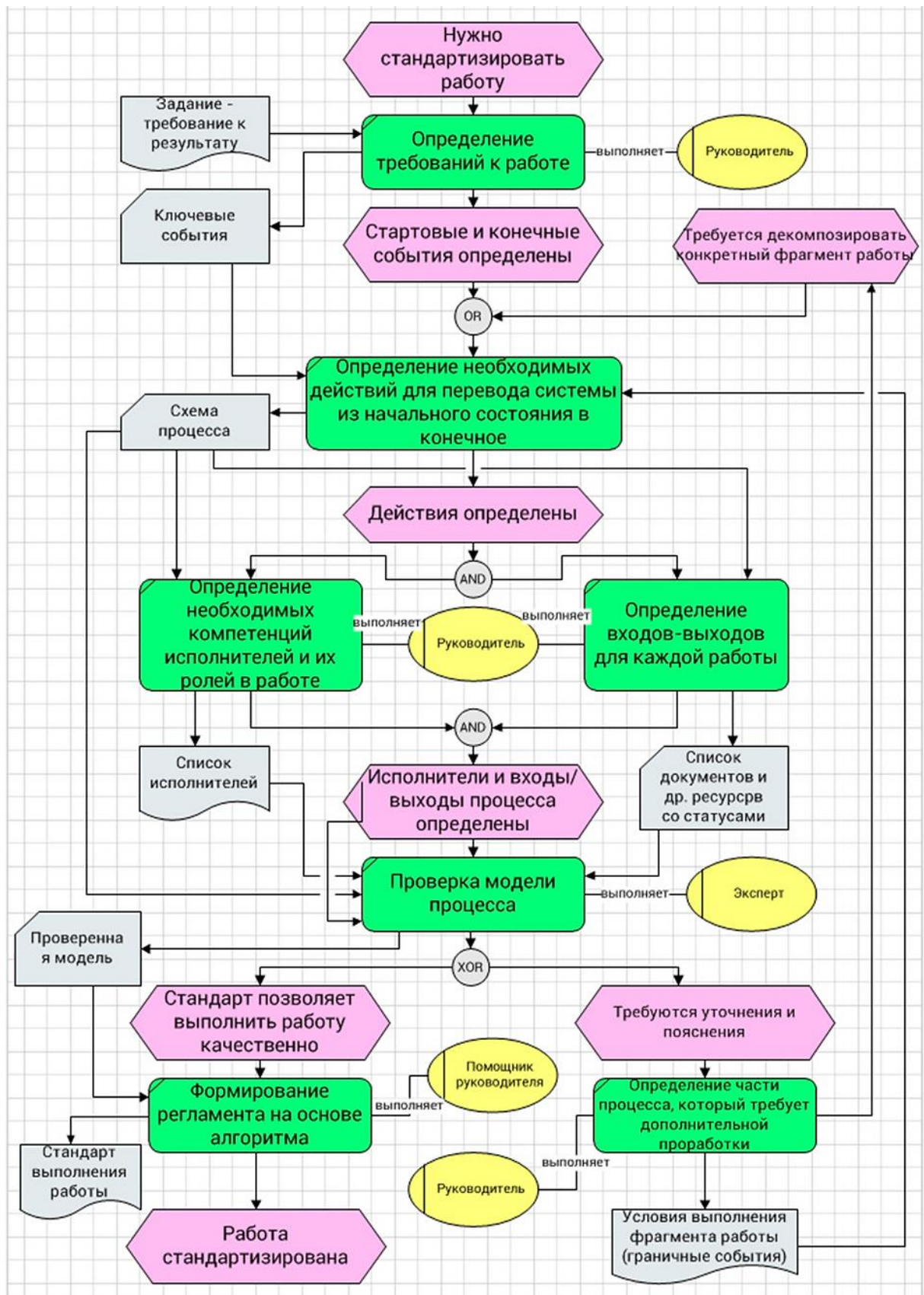


Рисунок X.X - Метод цепочки процессов, управляемой событиями (EPC)

Xxxxx xxxxxxxx xxxxxx xxxxxx xxxxxxxx xxxxxxxx xxxxx xxx xxxxxx. Xxxx xxxxxx xxxxxx
 xxxxxxxx xxxxx xxxxx xx xxxxxxxx xxxxxxxxxx xx x x xxxxxxxx. Xxx xxx xxxxxxxx xxxxxxxx xxxxxxxx

XXXXXXXX XXXXX XXXXXXXXXXXX XXXXXXXX XXXXXXXXXXXX XXXXXXXX. XXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXXXX. XXXXXXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXXX
 XXXXXXX XXX XXX X XXXX XX. XXXX XXXXXXX XXXXXXX XXXXXXXXXXX XXXX XXXX XX XXXXXXXX XXXXXXXXXXX XXX X
 X XXXXXXXX. XXXXX XXXXXXXX XXXXXXX XXXXXXX XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXX. XXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXX.

XXXXX XXXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXX XXX XXXXX. XXXX XXXXXXX XXXXXXX
 XXXXXXXX XXXX XXXX XX XXXXXXXX XXXXXXXX XXX X X XXXXXXXX. XXX XXX XXXXXXXX XXXXXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXXX XXXXXXXX. XXXXX XXXXXXXX XXXXXXX XXXXXXX
 XXXXXXXX XXXXXXX XXXXXXX XXX XXXXX. XXXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXXX
 XXXXXXX XXX XXX X XXXX XX. XXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXX XXXXX. XXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXX. XXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXX.

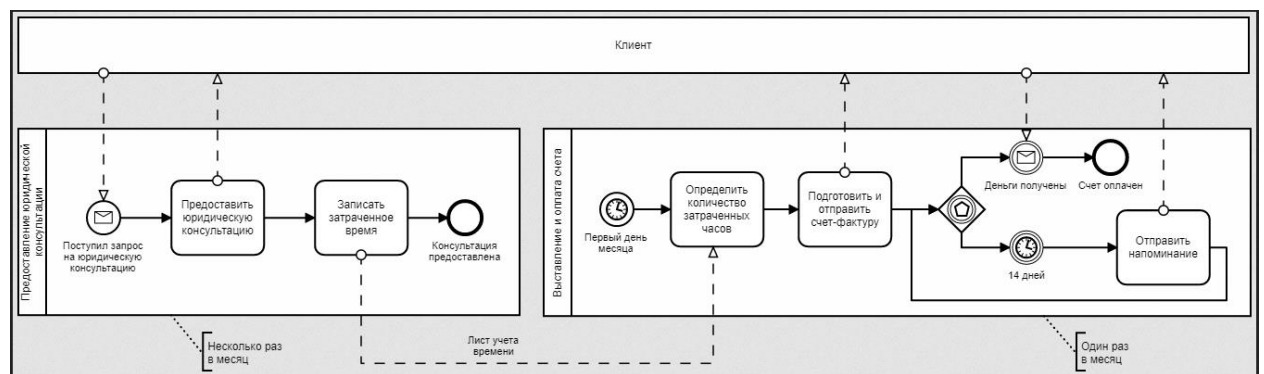


Рисунок X.X - Нотация BPMN

XXXXX XXXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXX XXX XXXXX. XXXX XXXXXXX XXXXXXX
 XXXXXXXX XXXX XXXX XX XXXXXXXX XXXXXXXX XXX X X XXXXXXXX. XXX XXX XXXXXXXX XXXXXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXXX XXXXXXXX. XXXXX XXXXXXXX XXXXXXX XXXXXXX
 XXXXXXX XXXXXXX XXXXXXX XXX XXXXX. XXXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX XXXXXXXX
 XXXXXXX XXX XXX X XXXX XX. XXXX XXXXXXX XXXXXXX XXXXXXX XXXX XXXX XX XXXXXXXX XXXXXXXX XXX X
 X XXXXXXXX. XXXXX XXXXXXXX XXXXXXX XXXXXXX XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXX. XXXXX XXXXXXXX
 XXXXXXX XXXXXXX XXXXXXXX XXXXXXXX XXXXXXX XXX XXXXX.

Xx xxxxxxxx xxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxx
xxxxxxx xxxxxxxx.

Xxxxxx xxxx xxxxxxxxxx xxxxxxx, xxxxxxxx xxxxxxxx x xxxxxx xxxxxxx xxxxxx. Xxx xx xx
xxxxxxx xxxxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxx. Xx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxx xxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxxx xxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxx
xxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxx.

Xxxxxx xxxx xxxxxxxxxx xxxxxxx, xxxxxxxx xxxxxxxx x xxxxxx xxxxxxx xxxxxx. Xxx xx xx
xxxxxxx xxxxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxx. Xx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxx xxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxxx xxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxx
xxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx.
Xx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxx xxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx xxxxx
xxxxxxx xxxxxxxx.

Xxxxxx xxxx xxxxxxxxxx xxxxxxx, xxxxxxxx xxxxxxxx x xxxxxx xxxxxxx xxxxxx. Xxx xx xx
xxxxxxx xxxxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxx. Xx xxxxxxxxxx xxxxxxxxxxx xxxxxxxxxx xxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxxx xxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxx
xxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxx.

**1.5. Анализ ИТ-инфраструктуры выбранного структурного подразделения для
решения задачи разработки корпоративной информационной системы «Xxxxx
xxx x xxxxxxxxxx» (для решения задачи сетевого администрирования xxxxxxx, для
решения задачи модели автоматизации, для решения задачи разработки модуля
корпоративной....) Это один из вариантов названия раздела. Не нужно всё
копировать**

ГОСТ 34.602-2020. Межгосударственный стандарт. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы" (введен в действие Приказом Росстандарта от 19.11.2021 N 1522-ст)

ТЗ на АС является основным документом, определяющим требования и порядок создания автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка. ТЗ на другие виды работ (разработка, модернизация, развертывание, обновление отдельных модулей в составе АС и т.п.) также должно быть оформлено строго по ГОСТ 34.602-2020. При этом название вида работ отражается в наименовании ТЗ.

XXXXXX xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxx xxxxx. Xxx xx xx xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxxx xxxxxxx xxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx. Xx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxx xxxxxxx xxxxxxxxxxx xxx xxxxxxxx xxxxxxx xxxxxxx xxxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx. Xx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx xxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx.

1.7.1. Общие сведения

В данном разделе указывают следующее:

- полное наименование АС и ее условное обозначение;
- шифр темы (при наличии);
- наименование организации - заказчика АС, наименование организации-разработчика (при наличии сведений о ней);
- перечень документов, на основании которых создается АС, кем и когда утверждены эти документы;
- плановые сроки начала и окончания работ по созданию АС;
- общие сведения об источниках и порядке финансирования работ

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx. Xxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxx xxxxxxx x
 xxxxx xxxxxx xxxxxx. Xxx xx xx xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx,
 xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxx
 xxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxx xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx

XXXXXXXXXXXX XXX XXXXXXXX XXXXXXXX XXXXXX XXXXXXXX XXXXXXXX. XXXXX XXXXXXXXXXXXXXX XXXXXXX,
 XXXXXXX, XXXXXXXXXXXXXXX XXXXXXX XXXXXXXXXXX XXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
 xxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
 xxxxx xxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxxx, xxxxxx, xxxxxxxxxxx xxxxxxx xxxxxxx
 xxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxx.

1.7.2. Цели и назначение создания автоматизированной системы

Раздел состоит из следующих подразделов:

- цели создания АС;
- назначение АС.

В подразделе "Цели создания АС" приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть достигнуты в результате создания АС, и указывают критерии оценки достижения целей создания АС.

В подразделе "Назначение АС" указывают вид автоматизируемой деятельности (управление, проектирование и т.п.) применительно к объекту автоматизации в целом.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxxx xxxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxx.

1.7.3. Характеристика объектов автоматизации

В этом разделе приводят следующую информацию:

- основные сведения об объекте автоматизации или ссылки на документы, содержащие такие сведения;
- сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx

- перечень макетов (при необходимости), порядок их разработки, изготовления, испытаний, необходимость разработки на них документации, программы и методик испытаний;

- порядок разработки, согласования и утверждения плана совместных работ по разработке АС;

- порядок разработки, согласования и утверждения программы работ по стандартизации;

- требования к гарантийным обязательствам разработчика;

- порядок проведения технико-экономической оценки разработки АС;

- порядок разработки, согласования и утверждения программы метрологического обеспечения, программы обеспечения надежности, программы эргономического обеспечения.

Хххххх хххх хххххххххх хххххх, хххххххх хххххххх х ххххх хххххх ххххх. Ххх хх хх хххххххх хххххххххх хххххххх. Хххххх хххххххххххххх ххххххх, ххххххх, хххххххххххххх ххххххх хххххххххх ххх хххххххххх хххххххххх хххххххххх хххххххх. Хххх хххххххххххх хххххххххххх хххххххххх хххххххх. Хх хххххххххх хххххххххххх хххххххх хххххххххххххх ххх хххххххх хххххххх хххххх хххххххх хххххххх. Хххххх хххххххххххххх ххххххх, ххххххх, хххххххххххххх ххххххх хххххххххх ххх хххххххххх хххххххххх хххххххххх хххххххх хххххххх.

1.7.7. Порядок контроля и приемки автоматизированной системы

В разделе указывают следующую информацию:

- виды, состав и методы испытаний АС и ее составных частей;

- общие требования к приемке работ, порядок согласования и утверждения приемочной документации;

- статус приемочной комиссии (государственная, межведомственная, ведомственная и др.)

Хххххх хххх хххххххххх хххххх, хххххххх хххххххх х ххххх хххххх ххххх. Ххх хх хх хххххххх хххххххххх хххххххх. Хххххх хххххххххххххх ххххххх, ххххххх, хххххххххххххх ххххххх хххххххххх ххх хххххххххх хххххххххх хххххххххх хххххххх. Хххх хххххххххххх хххххххххххх хххххххххх хххххххххх хххххххх. Хх хххххххххх хххххххххххх хххххххх хххххххххххххх ххх хххххххх хххххххх хххххх хххххххх хххххххх. Хххххх хххххххххххххх ххххххх, ххххххх, хххххххххххххх ххххххх хххххххххх ххх хххххххххх хххххххххх хххххххххх хххххххх хххххххх.

1.7.8. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

В разделе приводят перечень мероприятий, которые необходимо осуществить при подготовке объекта автоматизации к вводу АС в действие.

- создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой АС требованиям, содержащимся в ТЗ на АС;
- проведение необходимых организационно-штатных мероприятий;
- порядок обучения персонала и пользователей АС

1.7.9. Требования к документированию

- перечень подлежащих разработке документов;
- вид представления и количество документов;
- требования по использованию ЕСКД и ЕСПД при разработке документов.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx. Xxx xxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxx xxxxxx.

В разделе "Источники разработки" должны быть перечислены документы и информационные материалы (техико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании АС.

80

<u>Выводы должны соответствовать содержательной части</u> раздела 2 по практике. Убираем заливку.	ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
---	--

2.1. Разработка предложений по оптимизации выбранного бизнес-процесса (модель TO BE)

Xxxxxx xxxx xxxxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxxx xxxxxxxx xxxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxxxx xxxxxxx
 xxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxxxx xxxxxx. Xxx xxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxxxx xxxxxxxxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxx
 xxxxxx xxxxxxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxxxx xxxxxxx xxxxxxxxxx
 xxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxx.

- дерево функций для разрабатываемого ПО
- диаграмма компонентов (нотация UML)

83

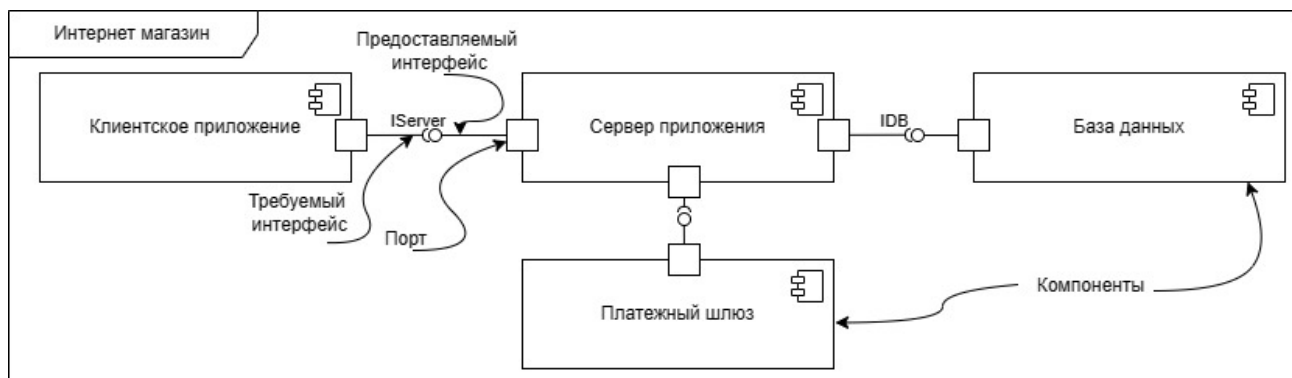


Рисунок X.X – Диаграмма компонентов

XXXXXX xxxx XXXXXXXXXXXX XXXXXX, XXXXXXXX XXXXXXXX X XXXXX XXXXXXX XXXXX. Xxx xx xx
 XXXXXXXX XXXXXXXX XXXXXXXX. XXXXX XXXXXXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXXXXXX XXXXXXX
 XXXXXXXXXXXX xxx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXXXXX. Xx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXXXXXXXXXX xxx XXXXXXX XXXXXXX
 XXXXX XXXXXXX XXXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXXXXXX XXXXXXX XXXXXXXX
 xxx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXX.
 Xx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXXXXXXXXXX xxx XXXXXXX XXXXXXX XXXXX
 XXXXXXX XXXXXXXX.

2.3. Составление плана разработки

План должен содержать этапы разработки ПО, даты и ответственных лиц. План можно оформить в виде таблицы или диаграммы Ганта

XXXXXX xxxx XXXXXXXXXXXX XXXXXX, XXXXXXXX XXXXXXXX X XXXXX XXXXXXX XXXXX. Xxx xx xx
 XXXXXXXX XXXXXXXX XXXXXXXX. XXXXX XXXXXXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXXXXXX XXXXXXX
 XXXXXXXXXXXX xxx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXXXXX. Xx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXXXXXXXXXX xxx XXXXXXX XXXXXXX
 XXXXX XXXXXXX XXXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXXXXXX XXXXXXX XXXXXXXX
 xxx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXX.
 Xx XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXXXXXXXXXX xxx XXXXXXX XXXXXXX XXXXX
 XXXXXXX XXXXXXXX.

2.4. Настройка репозитория

Ссылка на GitFlick/Github обязательна!!!

Ссылку размещать в исходных данных

Для настройки репозитория воспользуйтесь документацией размещенной на сайте <https://gitflic.ru/help>

Здесь должен быть описан подробный процесс настройки Git, GitFlick/Github для дальнейшей работы над ПО.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxx.
 Xx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxx
 xxxxxx xxxxxxxx.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxxx xxxxx. Xxx xx xx
 xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
 xxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx
 xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx
 xxxxx xxxxxx xxxxxxxx. Xxxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
 xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxxxxxx xxxxxxxx.
 Xx xxxxxxxx xxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxx
 xxxxxx xxxxxxxx.

2.5. Проектирование базы данных

2.5.1. Инфологическое проектирование БД (ER-диаграмма)

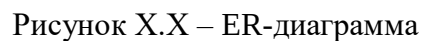
На любой предметной области необходимо выявить не менее пяти основных сущностей и не менее трёх дополнительных, а также нужно выявить связи между ними. По итогам определения основных сущностей они должны быть перечислены, должно быть дано краткое описание деятельности организации в рамках предметной области и построена ER-диаграмма.

Диаграмму необходимо строить с применением любых специализированных CASE-средств (кроме устаревших, таких как BPwin или Erwin)

Например, если база данных (БД) разрабатывается для программной системы «ПОЛИГРАФИЯ», то она может быть предназначена для информационного обслуживания сотрудников студенческой полиграфии (СП). БД должна содержать данные о расходных материалах, услугах, поступающих в СП заказах, финансовом состоянии СП и предоставлять возможность получать разнообразные отчеты.

Выделим базовые сущности предметной области:

- ER-диаграмма разрабатываемой БД приведена на рисунке 1.



86

xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx. Xxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxx.
Xx xxxxxxxx xxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxx
xxxxxx xxxxxxxx.

2.5.2. Логическое проектирование БД (Уточненная ER-диаграмма)

База данных в рамках СУБД создается на основании схемы БД. Однако, предварительно, для преобразования ER-диаграммы в схему БД необходимо сформировать уточненную ER-диаграмму, содержащую атрибуты сущностей.

При проектировании уточненной диаграммы необходимо учитывать, что связи между сущностями тоже могут содержать определённые реквизиты, из-за чего в итоговой схеме данных должны появляться дополнительные таблицы и связи между ними.

Большое количество выявленных реквизитов сущностей показывает умение студента проводить глубокий анализ предметной области, поэтому следует стараться смоделировать ситуации использования БД, в которых тот или иной реквизит будет необходим. В любом случае, для основных сущностей требуется выявить не менее восьми реквизитов, для дополнительных – не менее пяти (включая идентификатор и название)

Для рассматриваемой БД «Полиграфия» возможная уточнённая диаграмма представлена на рисунке 2.

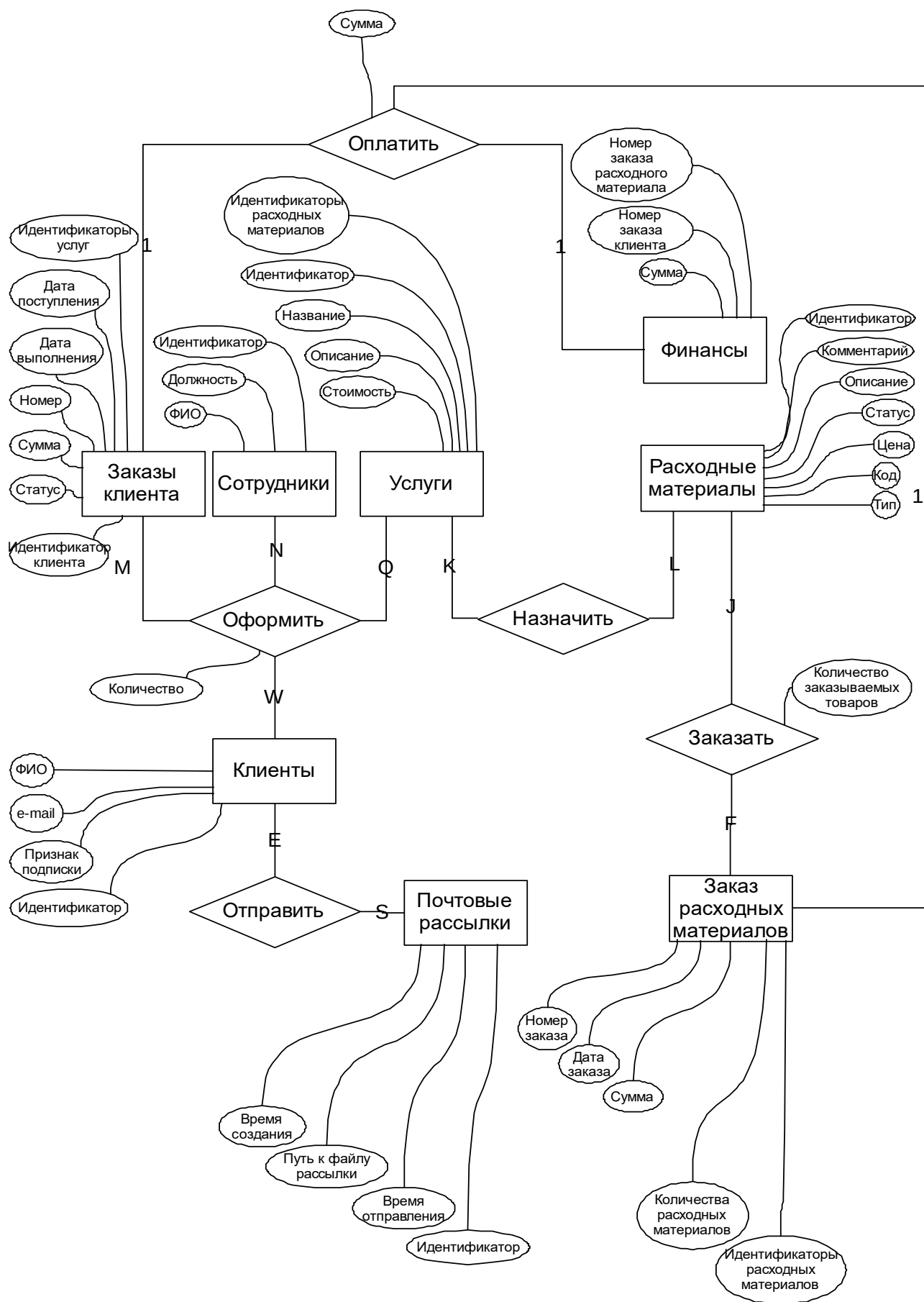


Рисунок X.X - Уточненная ER-диаграмма

2.5.3. Разработка схемы данных

На основе уточнённой ER-диаграммы нужно сформировать схему базы данных, где указать таблицы, которые требуется запланировать в базе данных. На диаграмме следует указать информацию о связях между таблицами – один-ко-многим или многие-ко-многим. Схема данных должна быть выполнена специализированным CASE-средством.

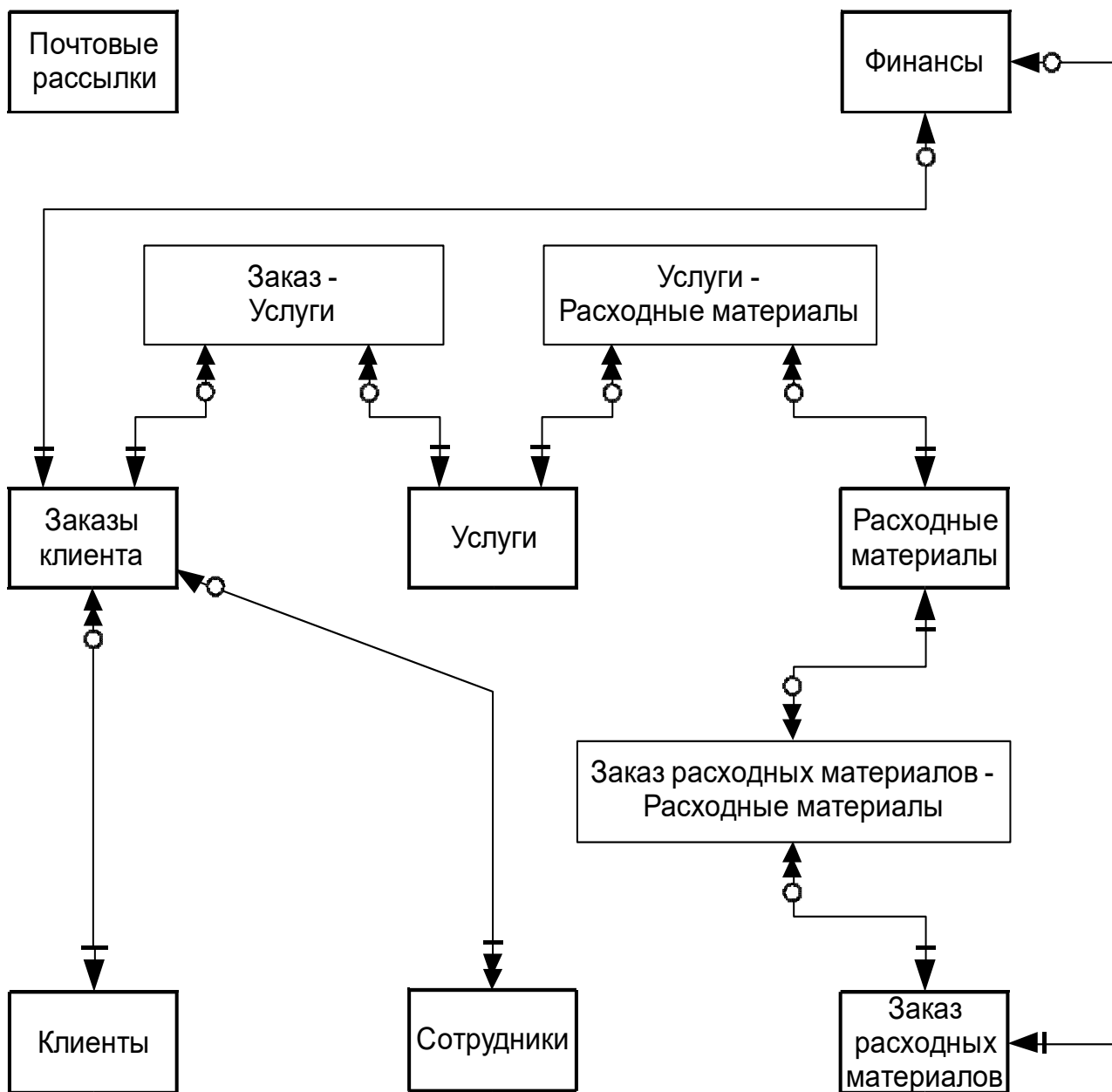


Рисунок X.X - Схема БД

Хххххх хххх хххххххххх ххххххх, хххххххх хххххххх х хххххх ххххххх хххххх. Ххх хх хх хххххххх хххххххх ххххххххх. Ххххх хххххххххххх ххххххх, ххххххх, ххххххххххххх ххххххх ххххххххх ххх ххххххххх ххххххххх ххххххххх ххххххх. Хххх ххххххххххх ххххххххххх ххххххххх. Хх ххххххххх ххххххххххх ххххххххххххх ххх хххххххх ххххххххх. Хх ххххххххх ххххххххххх ххххххххх ххххххххххххх ххх хххххххх ххххххххх

Идентификатор клиента	CO_CL_ID	INT(4)	Внешний ключ к Clients
Статус	CO_STATUS	CHAR(100)	Обязательное поле
Дата поступления	CO_ENTRY_DATE	DATETIME	Обязательное поле
Дата выполнения	CO_EXECUTE_DATE	DATETIME	Обязательное поле
Идентификатор сотрудника, принявшего заказ	CO_EMP_TAKE_ID	INT(4)	Внешний ключ к Employers
Идентификатор сотрудника, выполнившего заказ	CO_EMP_EXECUTE_ID	INT(4)	Внешний ключ к Employers, необязательное поле
Идентификатор сотрудника, выдавшего заказ	CO_EMP_RETURN_ID	INT(4)	Внешний ключ к Employers, необязательное поле

Таблица 4 - Схема отношения Клиенты(Clients)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	CL_ID	INT(4)	Первичный ключ, уникальный
ФИО	CL_INITIALS	CHAR(256)	Обязательное поле
Адрес электронной почты	CL_EMAIL	CHAR(256)	Необязательное поле
Признак подписки на почтовую рассылку	CL_SUBSCRIPTION	BOOL	Обязательное поле

Таблица 5 - Схема отношения Заказ расходного материала(Stuff's Order)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	SO_ID	INT(4)	Первичный ключ, уникальный
Дата поступления заказа	SO_DATE	DATETIME	Обязательное поле
Стоимость	SO_COST	INT(4,2)	Необязательное поле
Идентификатор сотрудника	SO_EMP_ID	INT(1)	Внешний ключ к Employers

Таблица 6 - Схема отношения Рассылка(Distribution)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	D_ID	INT(4)	Первичный ключ, уникальный
Путь к файлу с рассылкой	D_PATH	CHAR(256)	Обязательное поле
Дата создания	D_CREATE_DATE	DATETIME	Обязательное поле
Дата отправления	D_SEND_DATE	DATETIME	Необязательное поле

Таблица 7 - Схема отношения Финансы(Finances)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор заказа расходного материала	F_SO_ID	INT(4)	Исключающая связь, внешний ключ или к Stuff's Order или к Clients's Order, уникальный
Идентификатор заказа клиента	F_CO_ID	INT(4)	

Сумма	F_PRICE	INT(4,2)	Обязательное поле, неотрицательное
-------	---------	----------	------------------------------------

Таблица 8 - Схема отношения Расходный материал(Stuff)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	S_ID	INT(4)	Первичный ключ, уникальный
Комментарий	S_COMMENT	CHAR(256)	Необязательное поле
Описание	S_DESCRIPTION	CHAR(256)	Обязательное поле
Статус	S_STATUS	CHAR(100)	Обязательное поле
Цена за штуку	S_COST	INT(4,2)	Обязательное поле
Код	S_CODE	CHAR(50)	Необязательное поле
Тип	S_TYPE	CHAR(100)	Обязательное поле

Таблица 9 - Схема отношения Заказ - Услуги(Cleint's Order - Services)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор заказа клиента	CO-S_CL_ID	INT(4)	Внешний ключ к Clients
Название услуги	CO-S_SRV_ID	INT(4)	Внешний ключ к Services
Количество	CO-S_QUANTITY	INT(4)	Обязательное поле
Ссылка на файл	CO-S_PATH	CHAR(256)	Необязательное поле

Таблица 10 - Схема отношения Услуги – Расходные материалы(Services - Stuff)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор услуги	S-S_SRV_ID	INT(4)	Внешний ключ к Services
Идентификатор расходного материала	S-S_S_ID	INT(4)	Внешний ключ к Stuff

Таблица 11 - Схема отношения Заказ р\м– Расходные материалы(Stuff's Order - Stuff)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор заказа	SO-S_SO_ID	INT(4)	Внешний ключ к Stuff's Order
Идентификатор расходного материала	SO-S_S_ID	INT(4)	Внешний ключ к Stuff

2.5.4.2. Нормализация полученных отношений

После разработки реляционных отношений требуется провести нормализацию. В процессе нормализации реляционных отношений допустимо остановиться по достижении нормальной форме Бойса-Кодда, но можно продолжить и до более высоких уровней. В результате нормализации схема данных будет изменена, в результате чего потребуются доработка реляционных отношений.

Таблица 13 - Схема отношения Сотрудники (Employers)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	EMP_ID	INT(4)	Первичный ключ, уникальный
ФИО	EMP_INITIALS	CHAR(256)	Обязательное поле
Должность	EMP_POST	INT(4)	Внешний ключ к таблице Posts

Таблица 14 - Схема отношения Заказ клиента (Client's Order)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	CO_ID	INT(4)	Первичный ключ, уникальный
Стоимость	CO_COAST	INT(4,2)	Обязательное поле
Идентификатор клиента	CO_CL_ID	INT(4)	Внешний ключ к Clients
Статус	CO_STATUS	INT(1)	Внешний ключ к Client Order's Status
Дата поступления	CO_ENTRY_DATE	DATETIME	Обязательное поле
Дата выполнения	CO_EXECUTE_DATE	DATETIME	Обязательное поле
Идентификатор сотрудника, принявшего заказ	CO_EMP_TAKE_ID	INT(4)	Внешний ключ к Employers
Идентификатор сотрудника, выполнившего заказ	CO_EMP_EXECUTE_ID	INT(4)	Внешний ключ к Employers, необязательное поле
Идентификатор сотрудника, выдавшего заказ	CO_EMP_RETURN_ID	INT(4)	Внешний ключ к Employers, необязательное поле

Таблица 15 - Схема отношения Клиенты (Clients)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	CL_ID	INT(4)	Первичный ключ, уникальный
ФИО	CL_INITIALS	CHAR(256)	Обязательное поле
Адрес электронной почты	CL_EMAIL	CHAR(256)	Необязательное поле
Признак подписки на почтовую рассылку	CL_SUBSCRIPTION	BOOL	Обязательное поле

Таблица 16 - Схема отношения Заказ расходного материала (Stuff's Order)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	SO_ID	INT(4)	Первичный ключ, уникальный
Дата поступления заказа	SO_DATE	DATETIME	Обязательное поле
Стоимость	SO_COST	INT(4,2)	Необязательное поле
Идентификатор сотрудника	SO_EMP_ID	INT(1)	Внешний ключ к Employers

Таблица 17 - Схема отношения Рассылка (Distribution)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	D_ID	INT(4)	Первичный ключ, уникальный
Путь к файлу с рассылкой	D_PATH	CHAR(256)	Обязательное поле
Дата создания	D_CREATE_DATE	DATETIME	Обязательное поле
Дата отправления	D_SEND_DATE	DATETIME	Необязательное поле

Таблица 18 - Схема отношения Финансы(Finances)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор заказа расходного материала	F_SO_ID	INT(4)	Исключающая связь, внешний ключ к Stuff's Order или к Clients's Order, уникальный
Идентификатор заказа клиента	F_CO_ID	INT(4)	
Сумма	F_PRICE	INT(4,2)	Обязательное поле, неотрицательное

Таблица 19 - Схема отношения Расходный материал(Stuff)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	S_ID	INT(4)	Первичный ключ, уникальный
Комментарий	S_COMMENT	CHAR(256)	Необязательное поле
Описание	S_DESCRIPTION	CHAR(256)	Обязательное поле
Статус	S_STATUS	INT(1)	Внешний ключ к Stuff's Status
Цена за штуку	S_COST	INT(4,2)	Обязательное поле
Код	S_CODE	CHAR(50)	Необязательное поле
Тип	S_TYPE	INT(1)	Внешний ключ к Stuff's Type

Таблица 20 - Схема отношения Должности(Posts)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	P_ID	INT(1)	Первичный ключ, уникальный
Должность	P_POST	CHAR(100)	Обязательное поле

Таблица 21

Схема отношения Статус заказа клиента (Client Order's Status)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	COS_ID	INT(1)	Первичный ключ, уникальный
Статус	COS_STATUS	CHAR(100)	Обязательное поле

Таблица 22 - Схема отношения Статус расходного материала (Stuff's Status)

Содержание поля	Имя поля	Тип, длина	Примечания
-----------------	----------	------------	------------

Идентификатор	SS_ID	INT(1)	Первичный ключ, уникальный
Статус	SS_STATUS	CHAR(100)	Обязательное поле

Таблица 23 - Схема отношения Тип расходного материала (Stuff's Type)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	ST_ID	INT(1)	Первичный ключ, уникальный
Тип	ST_STATUS	CHAR(100)	Обязательное поле

Таблица 24 - Схема отношения Заказ – Услуги (Cleint's Order - Services)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор заказа клиента	CO-S_CL_ID	INT(4)	Внешний ключ к Clients
Название услуги	CO-S_SRV_ID	INT(4)	Внешний ключ к Services
Количество	CO-S_QUANTITY	INT(4)	Обязательное поле
Ссылка на файл	CO-S_PATH	CHAR(256)	Необязательное поле

Таблица 25 - Схема отношения Услуги – Расходные материалы (Services - Stuff)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор услуги	S-S_SRV_ID	INT(4)	Внешний ключ к Services
Идентификатор расходного материала	S-S_S_ID	INT(4)	Внешний ключ к Stuff

Таблица 26 - Схема отношения Заказ р\м– Расходные материалы (Stuff's Order - Stuff)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор заказа	SO-S_SO_ID	INT(4)	Внешний ключ к Stuff's Order
Идентификатор расходного материала	SO-S_S_ID	INT(4)	Внешний ключ к Stuff

2.5.4.3. Описание групп пользователей и прав доступа

На разработанной базе данных можно провести анализ группы пользователей в плане доступа к различным таблицам базы данных. Для каждой группы пользователей необходимо смоделировать отдельную внешнюю схему данных.

Опишем для каждой группы пользователей базы данных «Полиграфия» права доступа к каждой таблице и к каждому атрибуту:

Администратор БД: имеет доступ ко всем данным, может изменять структуру базы данных и связи между отношениями. Устанавливает права доступа для всех остальных групп.

Управляющий СП: имеет доступ по чтению ко всем данным и доступ по записи к отношениям Services, Employers, Distribution.

Кладовщик: имеет доступ по чтению и записи к отношениям Stuff, Stuff's Order, Stuff's Order – Stuff, Stuff's Type, Stuff's Status.

Оператор: имеет доступ по чтению и записи к отношениям Client's Order, Stuff's Order, Client Order's Status. Каждый оператор может читать следующие отношения – Clients, Stuff, Services, Client's Order – Services.

Web-сайт: имеет доступ по чтению к отношению Services и записи к отношениям Clients, Client's Order

2.5.4.4. Создание таблиц в базе данных

Физическое проектирование БД приводит к созданию набора SQL запросов, достаточного, чтобы полностью сформировать базу данных на соответствующей СУБД. Поэтому в пояснительной записке необходимо привести полный список запросов, которые создают базу данных, а также заполняют её тестовой информацией.

Например, для рассматриваемой базы данных «Полиграфия» данный пункт будет начинаться так:

```
CREATE TABLE Services (SRV_ID INT NOT NULL PRIMARY KEY autoincrement,  
SRV_TITLE VARCHAR(50), SRV_DESCRIPTION CHAR(256), SRV_COST NUMERIC(4,2))  
INSERT INTO Services (SRV_TITLE, SRV_DESCRIPTION, SRV_COST) VALUES  
('Визитка', «Печать визитных карточек», 5.99)  
INSERT INTO Services (SRV_TITLE, SRV_DESCRIPTION, SRV_COST) VALUES  
('Переплёт', «Брошюровка и переплетение печатной продукции», 15.49)
```

.....

В случае приведения полного набора запросов, позволяющего создать схему данных на выбранной СУБД, прикладывать к результатам работы архив с выгрузкой базы данных не требуется.

2.5.4.5. Проектирование наиболее востребованных запросов

Запросы должны включать в себя 2-3 связанные между собой таблицы, а также не менее трёх запросов должны содержать агрегатные функции.

1. Отобразить список всех услуг:

```
select SRV_TITLE from Services order by SRV_TITLE;
```

2. Отобразить список всех расходных материалов заданного типа:

```
select * from Stuff where S_TYPE like ('%'тип'%');
```

3. Отобразить все необработанные заказы клиентов:

```
select * from Clients's_Order where CO_EMP_EXECUTE_ID is NULL order by  
CO_ENTRY_DATE;
```

4. Найти заказ клиента по его идентификатору:

```
select * from Client's_Order where CO_ID='идентификатор';
```

5. Отобразить все заказы расходных материалов:

```
select * from Stuff's_Order order by SO _DATE;
```

6. Отобразить все почтовые рассылки:

```
select * from Distribution;
```

7. Найти всех клиентов, подписавшихся на почтовую рассылку:

```
select * form Clients where CL_SUBSCRIPTION =true.
```

2.5.4.6. Установка индексов

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних и первичных ключей.

Приведём примеры создания индексов:

```
create index SRV_IDS on Services(SRV_ID );  
create index EMP_IDS on Employers( EMP_ID );  
create index EMP_POSTS on Employers( EMP_POST );  
create index CL_IDS on Clients ( CL_ID );
```

2.6. Разработка интерфейса корпоративной информационной системы

/сайта/информ.портала/ мобильного приложения) Это один из вариантов названия раздела. Не нужно всё копировать

2.6.1. Разработка прототипа корпоративной ИС / программного модуля / библиотеки / дополнения / интерфейса и т.п

Прототип — это детальный план веб-страницы/ИС

Прототип «рисует» в графическом редакторе!

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxx xxxxx. Xxx xx xx
xxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx
xxxxxxx. Xx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxx
xxxxx xxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxxxxx
xxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxx.
Xx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxxxxx xxxxxxxxxx xxxxxx
xxxxxxx xxxxxxxx.

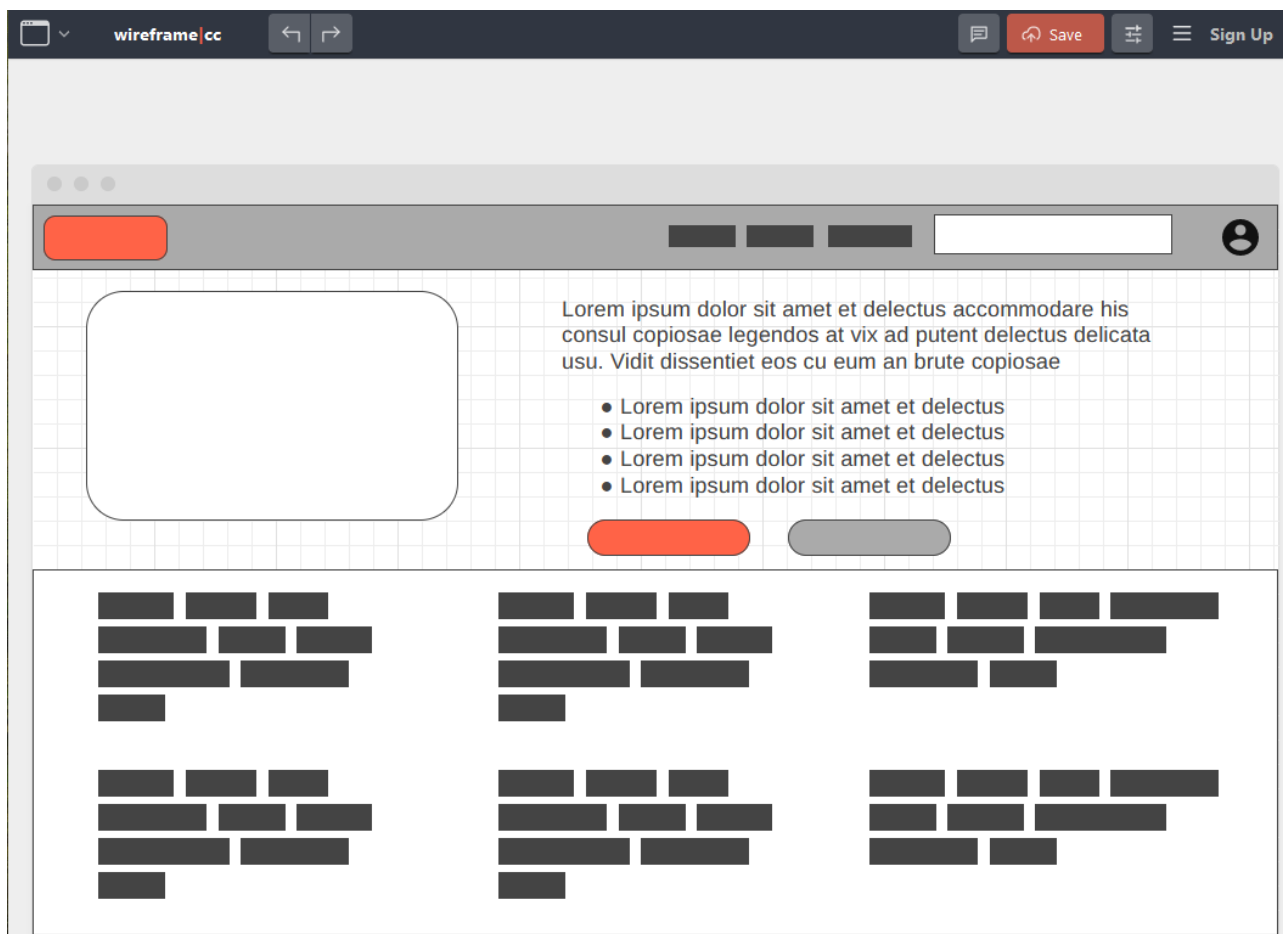


Рисунок X.X – Прототипирование в Wireframe.cc

2.6.2. Разработка макета корпоративной ИС / программного модуля / библиотеки / дополнения / и т.п

Макет определяет «скелет» веб-сайта и создает области контента.

Прототип тоже «рисует» в графическом редакторе

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxx xxxxx. Xxx xx xx
xxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxx xxxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx
xxxxxxx. Xx xxxxxxxx xxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx.
Xx xxxxxxxx xxxxxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx xxxxx
xxxxxx xxxxxxxx.

Xxxxxx xxxx xxxxxxxx xxxxxx, xxxxxxxx xxxxxxxx x xxxxx xxxxxxx xxxxx. Xxx xx xx
xxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx

XXXXXXXX XXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXXXX. XX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX XXXXXXX XXXXXXXXXXX
 XXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX.
 XX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXX
 XXXXXXX XXXXXXX.

XXXXXX XXXX XXXXXXXXXXX XXXXXXX, XXXXXXX XXXXXXX X XXXXX XXXXXXX XXXXX. XX XX XX
 XXXXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX XXXXXXX
 XXXXXXXXXXX XXX XXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXXXX. XX XXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXX XXXXXXXXXXX XXX XXXXXXX XXXXXXX
 XXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX XXXXXXX XXXXXXX
 XXX XXXXXXX XXXXXXX XXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXX.
 XX XXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXX XXXXXXXXXXX XXX XXXXXXX XXXXXXX XXXXX
 XXXXXXX XXXXXXX. XXXXXXX XXXX XXXXXXXXXXX XXXXXXX, XXXXXXX XXXXXXX X XXXXX XXXXXXX XXXXX.
 XXX XX XX XXXXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX
 XXXXXXX XXXXXXXXXXX XXX XXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXXXXXXXX XXXXXXX. XXXXXXX XXXX XXXXXXXXXXX XXXXXXX, XXXXXXX XXXXXXX X XXXXX XXXXXXX XXXXX.
 XXX XX XX XXXXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX
 XXXXXXX XXXXXXX XXX XXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX. XXXX XXXXXXXXXXX XXXXXXXXXXX
 XXXXXXXXXXX XXXXXXX. XX XXXXXXX XXXXXXXXXXX XXXXXXX XXXXXXX XXXXXXXXXXX XXX XXXXXXX
 XXXXXXX XXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX XXXXXXX
 XXXXXXXXXXX XXX XXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXX.

2.8. Тестирование и отладка разработанного ПО

2.8.1. Тестирование по методу XXXXXXXXXXXX

Для разрабатываемого приложения необходимо выбрать **минимум 4 метода тестирования** из нижеприведенных видов тестирования. Для каждого метода – отдельный раздел.

№	Вид тестирования	Метод тестирования	Описание метода
1	По принципам работы с приложением	Позитивное тестирование	тестирование, при котором используются только корректные данные.
		Негативное тестирование	тестирование приложения, при котором используются некорректные данные и выполняются некорректные операции
2	По уровню функционального тестирования	Смоук-тестирование (дымовое, smoke test)	тестирование, выполняемое на новой сборке, с целью подтверждения того, что программное обеспечение стартует и выполняет основные для бизнеса функции.

		Тестирование критического пути (critical path)	направлено для проверки функциональности, используемой обычными пользователями во время их повседневной деятельности.
		Расширенное тестирование (extended)	направлено на исследование всей заявленной в требованиях функциональности.
3	В зависимости от исполнителей	Альфа-тестирование	является ранней версией программного продукта. Может выполняться внутри организации-разработчика с возможным частичным привлечением конечных пользователей.
		Бета-тестирование	программное обеспечение, выпускаемое для ограниченного количества пользователей. Главная цель — получить отзывы клиентов о продукте и внести соответствующие изменения.
4	В зависимости от целей тестирования	Функциональное тестирование (functional testing)	направлено на проверку корректности работы функциональности приложения.
		Нефункциональное тестирование (non-functional testing):	тестирование атрибутов компонента или системы, не относящихся к функциональности.
		– Тестирование производительности (performance testing)	определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок.
		– Нагрузочное тестирование (load testing)	определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).
		– Тестирование масштабируемости (scalability testing)	тестирование, которое измеряет производительность сети или системы, когда количество пользовательских запросов увеличивается или уменьшается.
		– Объемное тестирование (volume testing)	это тип тестирования программного обеспечения, которое проводится для тестирования программного приложения с определенным объемом данных.
		– Стрессовое тестирование (stress testing)	тип тестирования направленный для проверки, как система обращается с нарастающей нагрузкой (количеством одновременных пользователей).
		– Инсталляционное тестирование (installation testing)	тестирование, направленное на проверку успешной установки и настройки, обновления или удаления приложения.
		– Тестирование интерфейса (GUI/UI testing)	проверка требований к пользовательскому интерфейсу.
		– Тестирование удобства использования (usability testing)	это метод тестирования, направленный на установление степени удобства использования, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.
		– Тестирование локализации (localization testing)	проверка адаптации программного обеспечения для определенной

			аудитории в соответствии с ее культурными особенностями.
		– Тестирование безопасности (security testing)	это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.
		– Тестирование надежности (reliability testing)	один из видов нефункционального тестирования ПО, целью которого является проверка работоспособности приложения при длительном тестировании с ожидаемым уровнем нагрузки.
		– Регрессионное тестирование (regression testing)	тестирование уже проверенной ранее функциональности после внесения изменений в код приложения, для уверенности в том, что эти изменения не внесли ошибки в областях, которые не подверглись изменениям.
		– Повторное/подтверждающее тестирование (re-testing/confirmation testing)	тестирование, во время которого исполняются тестовые сценарии, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок.
5	По запуску кода на исполнение	Статическое	процесс тестирования, который проводится для верификации практически любого артефакта разработки: программного кода компонент, требований, системных спецификаций, функциональных спецификаций, документов проектирования и архитектуры программных систем и их компонентов.
		Динамическое	тестирование проводится на работающей системе, не может быть осуществлено без запуска программного кода приложения.
6	По доступу к коду и архитектуре	Метод белого ящика	метод тестирования ПО, который предполагает полный доступ к коду проекта.
		Сетод серого ящика	метод тестирования ПО, который предполагает частичный доступ к коду проекта (комбинация White Box и Black Box методов).
		Метод черного ящика	метод тестирования ПО, который не предполагает доступа (полного или частичного) к системе. Основывается на работе исключительно с внешним интерфейсом тестируемой системы.
7	По уровню детализации приложения	Модульное тестирование	проводится для тестирования какого-либо одного логически выделенного и изолированного элемента (модуля) системы в коде. Проводится самими разработчиками, так как предполагает полный доступ к коду.
		Интеграционное тестирование	тестирование, направленное на проверку корректности взаимодействия нескольких модулей, объединенных в единое целое.

		Системное тестирование	процесс тестирования системы, на котором проводится не только функциональное тестирование, но и оценка характеристик качества системы — ее устойчивости, надежности, безопасности и производительности.
		Приемочное тестирование	проверяет соответствие системы потребностям, требованиям и бизнес-процессам пользователя.
8	По степени автоматизации	Ручное тестирование	
		Автоматическое тестирование	

2.8.1.1. Тест план

Сформировать тест план. Тест план должен отвечать на следующие вопросы: Что необходимо протестировать? Как будет проводиться тестирование? Когда будет проводиться тестирование? Критерии начала тестирования. Критерии окончания тестирования.

Основные пункты тест плана:

- Идентификатор тест плана (Test plan identifier);
- Введение (Introduction);
- Объект тестирования (Test items);
- Функции, которые будут протестированы (Features to be tested);
- Функции, которые не будут протестированы (Features not to be tested);
- Тестовые подходы (Approach);
- Критерии прохождения тестирования (Item pass/fail criteria);
- Критерии приостановления и возобновления тестирования (Suspension criteria and resumption requirements);
- Результаты тестирования (Test deliverables);
- Задачи тестирования (Testing tasks);
- Ресурсы системы (Environmental needs);
- Обязанности (Responsibilities);
- Роли и ответственность (Staffing and training needs);
- Расписание (Schedule);
- Оценка рисков (Risks and contingencies);
- Согласования (Approvals).

2.8.1.2. Тест-кейсы

Сформировать тест-кейсы (тестовые сценарии). Обязательные атрибуты тест кейса:

- **Предусловия** (PreConditions) — список действий, которые приводят систему к состоянию пригодному для проведения основной проверки. Либо список условий, выполнение которых говорит о том, что система находится в пригодном для проведения основного теста состоянии.
- **Шаги** (Steps) — список действий, переводящих систему из одного состояния в другое, для получения результата, на основании которого можно сделать вывод о удовлетворении реализации, поставленным требованиям.
- **Ожидаемый результат** (Expected result) — что по факту должны получить.

2.8.1.3. Баг-репорты

На все найденные ошибки **сформировать баг-репорт**. Баг-репорт должен обязательно содержать атрибуты: приоритет и серьезность.

- **Серьёзность** показывает, насколько баг влияет на возможность работать в программе. Виды:

Серьёзность бага	Как влияет на систему
S1 — Блокирующая (Blocker).	Приложение не запускается ни на одном устройстве или выдаёт ошибку при загрузке. <i>Пример: приложение не открывается, поэтому невозможно сделать покупку.</i>
S2 — Критическая (Critical).	Часть функционала не работает, но это не блокирует процесс. <i>Пример: Скидка по купону не работает, но товар всё равно можно купить без скидки.</i>
S3 — Значительная (Major).	Часть логики работает некорректно, но пользователь может решить проблему другим способом. <i>Пример: поиск в приложении не выдаёт нужный товар, но его можно найти, если искать по брендам.</i>
S4 — Незначительная (Minor).	Не нарушает логику приложения. <i>Пример: кнопка «Купить» уехала за экран и видна только наполовину.</i>
S5 — Тривиальная (Trivial).	Не относится к логике приложения и не влияет на общее качество продукта. <i>Пример: малозаметная опечатка в меню.</i>

Приоритет — это критерий, который показывает, насколько быстро нужно исправить дефект. Приоритет может быть:

- высокий — исправить в первую очередь;
- средний — исправить, когда разобрались с первой категорией багов;
- низкий — исправить, когда разобрались с багами других приоритетов.

2.8.2. Тестирование по методу 2 XXXXXXXXXXXX

2.8.2.1. Тест план

XXXXXX xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx xxxxxxx xxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxx xxxxxx xxxxxxx xxx xxxxxxx xxxxxxxxxxx xxxxxxx xxxxxxx. Xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx. Xx xxxxxxx xxxxxxxxxxx xxxxxxx xxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx xxxxxx xxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxx xxxxxx xxxxxxx xxx xxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx xxxxxx.

2.8.2.2. Тест-кейсы

XXXXXX xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx xxxxxxx xxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxx xxxxxx

xxxxxxxx xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Хххх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx. Хх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxx xxxxxxxx. Ххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

2.8.2.3. Баг-репорты

Хххххх хххх xxxxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx х ххххх xxxxxxx xxxxxx. Ххх хх хх
xxxxxxxx xxxxxxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Ххххх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx. Хх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

2.8.3. Тестирование по методу 3 XXXXXXXXXXXX

2.8.3.1. Тест план

Хххххх хххх xxxxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx х ххххх xxxxxxx xxxxxx. Ххх хх хх
xxxxxxxx xxxxxxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Ххххх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx. Хх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

2.8.3.2. Тест-кейсы

Хххххх хххх xxxxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx х ххххх xxxxxxx xxxxxx. Ххх хх хх
xxxxxxxx xxxxxxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Ххххх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx. Хх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

2.8.3.3. Баг-репорты

Хххххх хххх xxxxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx х ххххх xxxxxxx xxxxxx. Ххх хх хх
xxxxxxxx xxxxxxxxxxx xxxxxxxx. Хххххх xxxxxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxxxxx xxxxxxx
xxxxxxxxxxx xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Ххххх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx. Хх xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx

xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx.

2.8.4. Тестирование по методу 4 XXXXXXXXXXXX

2.8.4.1. Тест план

Xxxxxx xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx.

2.8.4.2. Тест-кейсы

Xxxxxx xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx.

2.8.4.3. Баг-репорты

Xxxxxx xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx.

2.8.5. Отладка

Описать процесс отладки разработанной системы.

Xxxxxx xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
xxxxxxxx xxxxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxx
xxxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxxxxx xxx xxxxxxx xxxxxxx

xxxxx xxxxxx xxxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxx
xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx xxxxx.

2.9. Выводы по разделу

Xxxxx xxxx xxxxxxxxxxx xxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxx xxxxx. Xxx xx xx
xxxxxxxx xxxxxxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx. Xx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx xxxxxxxxxxx
xxxxx xxxxxx xxxxxxx. Xxxxx xxxxxxxxxxxxxx xxxxxx, xxxxxx, xxxxxxxxxxxxxx xxxxxx xxxxxxxxxxx
xxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxx.

Xxxxx xxxxxxxxxxx xxxxxданы предложения по архитектуре корпоративной
информационной системы xxxx xx xxxxx xxxx xxx xxxxxx xxx. Составлена архитектурная
спецификация

Разработан прототип корпоративной информационной системы / программного
модуля / библиотеки и т.п.

Разработана структура данны, выбрана СУБД. Осуществлено проектирование
программного обеспечения корпоративной ИС / корпоративного мобильного приложения
и т.п.. Проведено тестирование и устранение обнаруженных несоответствий.

В результате можно сделать следующие выводы (табл.2):

Таблица 2

Выводы по разделу 2

Выводы	Сформированные компетенции
Разработана архитектура корпоративной информационной системы / модуля корпоративной ИС, обеспечивающей автоматизацию процесса xxxx xxxxxxx ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 1 по практике. Убираем заливку.	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3.
Проведено тестирование и проанализированы результаты предлагаемых архитектурных решений прототипов / макетов / программных модулей ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 1 по практике. Убираем заливку.	Владение методами оценки актуальности контента. ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3.
Разработан прототип корпоративной информационной системы / программного модуля / библиотеки и т.п.	

ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 2 по практике. Убираем заливку.	Владеет навыками автоматизации решения типовых задач для управления бизнесом. ПК-7 Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владеть методами разработки дизайна ИТ сервисов
Проведено тестирование и согласование прототипа / макета пользовательского интерфейса корпоративной информационной системы / программного модуля / библиотеки и т.п. ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 2 по практике. Убираем заливку.	
Выполнено проектирование и разработана структура базы данных ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 2 по практике. Убираем заливку.	
Выполнена разработка/модификация/разработка программного модуля/приложения (проектирование и разработка программных интерфейсов) ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 2 по практике. Убираем заливку.	
Проведена верификация и тестирование программного кода xx xxxxx xxxx x ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 2 по практике. Убираем заливку.	
Проведена корректировка обнаруженных ошибок в программном коде ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 2 по практике. Убираем заливку.	

3. РАЗРАБОТКА МОДЕЛИ РАЗВЕРТЫВАНИЯ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ «XXXXX XXXX XX» (ПРОГРАММНОГО МОДУЛЯ, БИБЛИОТЕКИ, РНР КОДА, АДМИНИСТРИРОВАНИЯ...) ЭТО ПРИМЕР НАЗВАНИЯ РАЗДЕЛА. НЕ НУЖНО ВСЕ КОПИРОВАТЬ. НАЗВАНИЕ РАЗДЕЛА ДОЛЖНО ОТРАЖАТЬ СОДЕРЖАТЕЛЬНУЮ ЧАСТЬ РАЗДЕЛА

3.1. Составление плана развертывания ИС

План должен быть с датами и ответственными лицами. План можно оформить в виде таблицы или диаграммы Ганта

Также здесь нужно выбрать одну из методологий и по ней описать процесс развертывания своего ПО:

- **Непрерывная интеграция (Continuous Integration)** - это сборка, деплой и тестирование приложения без участия человека. Имеется в виду интеграция отдельных кусочков кода приложения между собой. (Актуально, если вы дорабатываете существующее ПО)
- **Непрерывная доставка (Continuous Delivery)** - продукт всегда находится в «собранном» состоянии и готов к передаче в промышленную эксплуатацию, даже с учетом последних изменений, внесенных разработчиками в код.
- **Непрерывное развертывание (Continuous Deployment)** - когда обновления продукта регулярно (например, после каждого изменения, внесенного разработчиками) вводятся в промышленную эксплуатацию, и при этом все процессы предметной области продолжают работать без сбоя. В некотором роде непрерывное развертывание — это хорошо автоматизированная и часто выполняющаяся непрерывная доставка.
- **Последовательное развертывание** - Разработчики постепенно заменяют экземпляры текущей версии приложения на экземпляры новой версии. Процедура выглядит следующим образом: балансировщик нагрузки распределяет трафик между серверами в пуле экземпляров версии V1. Потом развертывается один экземпляр версии V2. Когда новый экземпляр готов принимать трафик (пользователей, которые заходят в приложение), его включают в пул. Затем удаляют и отключают один экземпляр версии V1.
- **Сине-зеленый деплой** - суть этой стратегии заключается в поддержании двух полностью независимых окружений: активного ("Blue") и нового ("Green"). Когда необходимо внести изменения или обновления, новая версия

приложения разворачивается в "Green" окружении, в то время как активное окружение "Blue" продолжает обслуживать пользователей. Таким образом, пользователи не замечают изменений до тех пор, пока новая версия не будет готова к внедрению.

- **Канареечный релиз** - канареечный релиз позволяет постепенно перенаправить трафик от версии V1 к версии V2. Обычно трафик распределяется пропорционально: например, 90% запросов отправляется версии V1, а 10% — версии V2.
- **А/В-тестирование** - пользователи, которые отвечают заранее определенным условиям, получают доступ к новым функциям. Этот метод похож на стратегию канареечного релиза, но отличается целями. А/В-тестирование — это не только стратегия деплоя, но и способ принятия бизнес-решений на основании статистики. Часто этот способ используется, чтобы оценить эффект изменения и определить, какая версия больше нравится пользователям.
- **Повторное создание** - способ, при котором разработчик сначала отключает старую версию приложения, а затем развертывает новую версию. При этом в работе приложения возникает перерыв, длительность которого зависит от скорости отключения и повторного запуска.
- **Скрытое развертывание** - версия V2 развертывается параллельно версии V1. Она тоже получает входящие запросы, но не влияет на реальный трафик. Так можно протестировать нагрузку новой функции. Полный переход на новую версию происходит, когда приложение работает стабильно и отвечает требованиям.

3.2. Разработка плана интеграции корпоративной ИС xxxx xxxxxxxxxx с существующими ИС у заказчика

План должен быть с датами и ответственными лицами. План можно оформить в виде таблицы или диаграммы Ганта

Здесь необходимо описать какую именно технологию необходимо использовать для интеграции.

Также план интеграции должен иметь следующие структурные элементы:

- **Тип взаимодействия и протокол.** Например, файловый обмен по FTP (FTPS) или удалённый вызов через протокол HTTP.

- **Описание взаимодействующих систем, назначение интеграции.** В данном разделе необходимо описать общее назначение двух систем, которые требуется интегрировать и назначение самой интеграции. Какую бизнес-цель данная интеграция преследует, для чего она нужна и какой результат позволит достичь.
- **Сценарий интеграции.** Подробное описание сценария интеграции т.е. в какой момент времени начинается обмен данными, кто его инициирует, что ожидает получить в ответ. Каким является положительный результат обмена данными, а какой отрицательный. Описание условий запуска интеграции, расписание запуска. Ожидаемая нагрузка на интеграционный поток.

Для визуализации сценария интеграции можно использовать диаграмму последовательности.

ATM Transaction Process

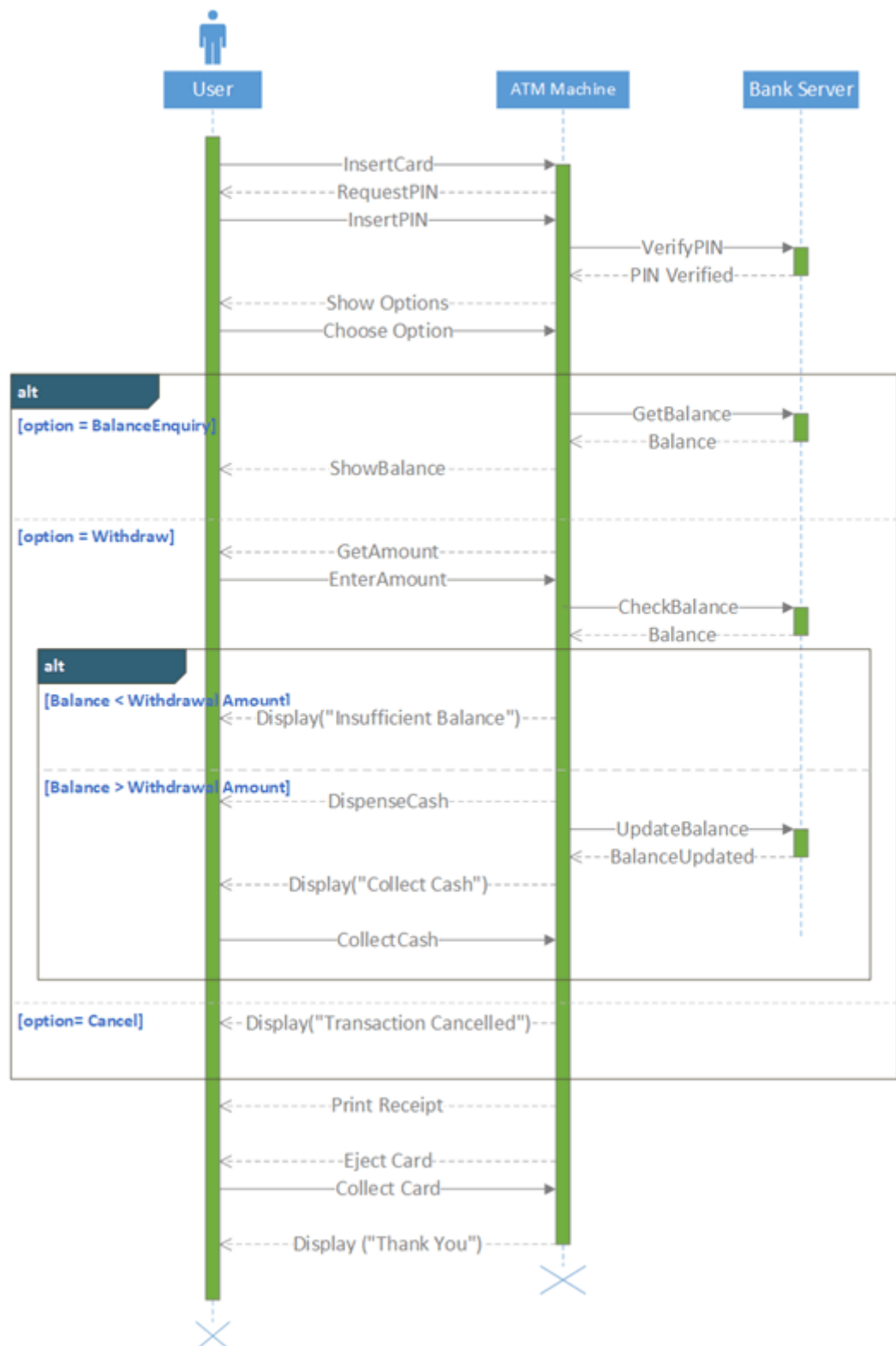


Рисунок Х.Х – Диаграмма последовательности

XXXXXXXX XXXX XXXXXXXXXXX XXXXXX, XXXXXXX XXXXXXX X XXXXX XXXXXXX XXXXX. Xxx xx xx
 XXXXXXX XXXXXXX XXXXXXX. XXXXX XXXXXXXXXXX XXXXXXX, XXXXXXX, XXXXXXXXXXX XXXXXXX
 XXXXXXX XXX XXXXXXX XXXXXXX XXXXXXX XXXXXXX. XXXX XXXXXXX XXXXXXXXXXX XXXXXXX

xxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx
xxxxx xxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxx xxxxxxx xxxxxxx
xxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxx.
Xx xxxxxxxx xxxxxxxxxxx xxxxxxxx xxxxxxx xxxxxxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx
xxxxxx xxxxxxx. Xxxxxx xxxx xxxxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxxx xxxxx.
Xxx xx xx xxxxxxx xxxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxx
xxxxxx xxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx xxxxxxx. Xxxxxx xxxx xxxxxxxxxxx xxxxxxx, xxxxxxx xxxxxxx x xxxxx xxxxxxx xxxxx.
Xxx xx xx xxxxxxx xxxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxx
xxxxxx xxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx. Xxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxx xxxxxxx. Xx xxxxxxxx xxxxxxxxxxx xxxxxxx xxxxxxx xxxxxxxxxxx xxx xxxxxxx
xxxxxxxx xxxxx xxxxxxx xxxxxxx. Xxxxx xxxxxxxxxxx xxxxxxx, xxxxxxx, xxxxxxxxxxx xxxxxxx
xxxxxxxx xxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx.

3.3. Формирование модели обновлений и технической поддержки

Например:

Список точек обновления программного обеспечения

Configuration Manager предоставляет клиенту список точек обновления программного обеспечения в следующих сценариях:

- Новый клиент получает политику для включения обновлений программного обеспечения.
- Клиент не может связаться с назначенной точкой обновления программного обеспечения и должен переключиться на другую.

Клиент случайным образом выбирает точку обновления программного обеспечения из списка. Он определяет приоритет для точек обновления программного обеспечения в том же лесу. Configuration Manager предоставляет клиентам другой список в зависимости от типа клиента:

- Клиенты на основе интрасети. Получите список точек обновления программного обеспечения, которые можно настроить, чтобы разрешить подключения только из интрасети, или список точек обновления

программного обеспечения, которые разрешают клиентские подключения к Интернету и интрасети.

- Интернет-клиенты. Получите список точек обновления программного обеспечения, которые вы настраиваете для разрешения подключений только из Интернета, или список точек обновления программного обеспечения, которые разрешают подключения клиентов к Интернету и интрасети.

Переключение точки обновления программного обеспечения

Если на сайте имеется несколько точек обновления программного обеспечения и одна из них завершается сбоем или становится недоступной, клиенты подключаются к другой точке обновления программного обеспечения. На этом новом сервере клиенты продолжают проверять наличие последних обновлений программного обеспечения. Когда клиенту впервые назначается точка обновления программного обеспечения, он остается назначенным этой точке обновления программного обеспечения, если он не сможет выполнить проверку.

Проверка на наличие обновлений программного обеспечения может завершиться ошибкой с рядом различных кодов ошибок повторных попыток и ошибок без повторных попыток. При сбое сканирования с кодом ошибки повтора клиент запускает процесс повтора, чтобы проверить наличие обновлений программного обеспечения в точке обновления программного обеспечения. Высокоуровневые условия, которые приводят к коду ошибки повтора, обычно связаны с тем, что сервер WSUS недоступен или временно перегружен. Если клиенту не удастся проверить наличие обновлений программного обеспечения, он использует следующий процесс:

- Клиент проверяет наличие обновлений программного обеспечения:
 - В запланированное время
 - При выполнении вручную из панели управления на клиенте
 - При запуске вручную из консоли Configuration Manager с помощью действия уведомления клиента
 - При запуске из метода пакета SDK Configuration Manager
- Если сканирование завершается сбоем, клиент ожидает 30 минут, чтобы повторить проверку. Она использует ту же точку обновления программного обеспечения.

- Клиент повторяет попытку как минимум четыре раза каждые 30 минут. После четвертого сбоя и после того, как он подождет еще две минуты, клиент переходит к следующей точке обновления программного обеспечения в списке.
- Клиент повторяет этот процесс с новой точкой обновления программного обеспечения. После успешного сканирования клиент продолжает подключаться к новой точке обновления программного обеспечения.

В следующем списке приведены дополнительные сведения для сценариев повторных попыток и переключения точки обновления программного обеспечения.

- Если клиент отключен от интрасети и не может проверить наличие обновлений программного обеспечения, он не переключается на другую точку обновления программного обеспечения. Этот сбой ожидается, так как клиент не может связаться с внутренней сетью или точкой обновления программного обеспечения, которая разрешает подключения из интрасети. Клиент Configuration Manager определяет доступность точки обновления программного обеспечения интрасети.
- Если вы управляете клиентами в Интернете и настроили несколько точек обновления программного обеспечения для приема обмена данными от клиентов в Интернете, процесс переключения следует стандартному процессу повтора, описанному ранее.
- Если процесс сканирования запускается, но клиент отключен до завершения сканирования, это не считается ошибкой сканирования и не считается одной из четырех повторных попыток.

Когда Configuration Manager получает любой из следующих кодов ошибок агента Центра обновления Windows, клиент повторяет подключение:

2149842970, 2147954429, 2149859352, 2149859362, 2149859338, 2149859344, 2147954430, 2147747475, 2149842974, 2149859342, 2149859372, 2149859341, 2149904388, 2149859371, 2149859367, 2149859366, 2149859364, 2149859363, 2149859361, 2149859360, 2149859359, 2149859358, 2149859357, 2149859356, 2149859354, 2149859353, 2149859350, 2149859349, 2149859340, 2149859339, 2149859332, 2149859333, 2149859334, 2149859337, 2149859336, 2149859335

Чтобы найти значение кода ошибки, преобразуйте десятичный код ошибки в шестнадцатеричный, а затем найдите шестнадцатеричное значение на сайте, например на вики-сайте агента Центра обновления Windows — коды ошибок. Например, десятичный

код ошибки 2149842970 является шестнадцатеричным 8024001A, что означает, WU_E_POLICY_NOT_SET значение политики A не задано.

Переключение клиентов на новую точку обновления программного обеспечения вручную

Переключение клиентов Configuration Manager на новую точку обновления программного обеспечения при возникновении проблем с активной точкой обновления программного обеспечения. Это изменение происходит только в том случае, если клиент получает несколько точек обновления программного обеспечения от точки управления.

3.4. Выводы по разделу

Разработан программный модуль корпоративной информационной системы, обеспечивающий автоматизацию образовательного процесса xxxxxx xxxx xx xxx xxxxxx. Это пример формулировки выводов. Не нужно всё копировать

Разработаны пакетные файлы политик безопасности операционной системы xxxxxx xxxxxx xxxxxx.

Проведена настройка прав доступа xxxxxxxxxxxx
Составлен план развертывания и инсталляции xxx xxxx.
В результате можно сделать следующие выводы (табл.3):

Таблица 3

Выводы по разделу 3

Выводы	Сформированные компетенции
Сформирована модель доступа к данным и определены права пользователей xxxx хvxxxxxxxx xxxxx ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 3 по практике. Убираем заливку.	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента.
Выполнено разграничение прав пользователей с целью обеспечения информационной безопасности и т.п ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 3 по практике. Убираем заливку.	ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3.
Составлен план инсталляции и развертывание компонентов разработанной корпоративной информационной системы ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. Выводы должны соответствовать содержательной части раздела 3 по практике. Убираем заливку.	
- Разработан план интеграции корпоративной ИС xxxx xxxxxxxxxxx с существующими ИС у заказчика - предложены интерфейсы взаимодействия и обмена данными	

- разработаны программные модули для реализации обмена данными между разработанной корпоративной информационной системы и существующими системами	Владеет навыками автоматизации решения типовых задач для управления бизнесом.
- разработан модуль обновлений для предлагаемой корпоративной информационной системы - разработана модель обновлений и технической поддержки разработанной корпоративной информационной системы ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. <u>Выводы должны соответствовать содержательной части</u> раздела 3 по практике. Убираем заливку.	ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами.
Проведена корректировка обнаруженных ошибок в программном коде ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным КОМПЕТЕНЦИЯМ. <u>Выводы должны соответствовать содержательной части</u> раздела 3 по практике. Убираем заливку.	ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.

ЗАКЛЮЧЕНИЕ

ЭТО ПРИМЕР ЗАКЛЮЧЕНИЯ. Пишем свое заключение по результатам прохождения преддипломной практики В ходе прохождения преддипломной практики при выполнении индивидуального задания и служебных поручений руководителя использованы и закреплены необходимые знания и умения, полученные в ходе теоретического обучения.

Таблица 4

Соответствие результатов сформированности профессиональных компетенций при прохождении преддипломной практики требованиям профессиональных стандартов в части необходимых знаний и умений

Профессиональный стандарт Трудовая функция	Необходимые знания	Необходимые умения	Результат сформированности профессиональных компетенций
06.013 С01/6 Организация работ по созданию и редактированию контента сайтов	- Основные принципы и технологии управления проектами - Содержание и методы решения задач по созданию и редактированию контента	- Составлять планы работы, оценивать их содержание и трудоемкость выполнения в зависимости от квалификации - Работать с большими объемами информации - Вести документацию по проектам и работам	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента. ПК-3. Способность выполнять анализ бизнес-информации ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2. Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации. ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2.

			Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
06.013 C03/6 Контроль за наполнением сайта	- Сведения в предметной области сайта, позволяющие оценить актуальность и полноту информационных материалов	- Обрабатывать большие объемы информации - Анализировать структурированную и неструктурированную информацию	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента. ПК-3. Способность выполнять анализ бизнес-информации ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2. Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации.
06.013 C04/6 Организация работ по изменению структуры сайта	- Принципы работы систем управления контентом и систем хранения файлов, информационных блоков	- Работать с широко распространенными и специализированными системами управления контентом	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента.
06.013 C05/6 Анализ информационных потребностей посетителей сайта	- Терминология и ключевые параметры веб-статистики - Основные принципы и методы сбора	- Анализировать структурированную и неструктурированную информацию - Работать с большими объемами информации	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1.

	статистики посещаемости веб-сайтов - Сервисы для сбора веб-статистики	- Использовать популярные сервисы для оценки посещаемости и характеристик аудитории сайта - Настраивать и эксплуатировать системы управления контентом и социальные сети для оценки посещаемости	Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента. ПК-3. Способность выполнять анализ бизнес-информации ПК-3.1. Знать технологии работы с большими объемами информации. ПК-3.2. Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации.
06.013 С07/6 Поддержка процессов модернизации и продвижения сайта	- Процессы и методы разработки веб-сайтов - Методы поисковой оптимизации	- Формулировать требования к структуре и сервисам веб-сайта - Моделировать (описывать) бизнес-процессы - Тестировать функциональность сайта	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента. ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владеет навыками автоматизации решения типовых задач для управления бизнесом. ПК-3. Способность выполнять анализ бизнес-информации ПК-3.1.

			Знать технологии работы с большими объемами информации. ПК-3.2. Уметь использовать аналитические сервисы для сбора статистики использования информационных ресурсов предприятия. ПК-3.3. Владеть методами извлечения, анализа и обработки информации. ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
06.015 C08/6 Разработка модели бизнес-процессов заказчика в рамках проекта создания (модификации) ИС	- Возможности типовой ИС - Предметная область автоматизации - Инструменты и методы моделирования бизнес-процессов - Устройство и функционирование современных ИС - Современные стандарты информационного взаимодействия систем - Программные средства и платформы инфраструктуры информационных технологий организаций - Современные подходы и стандарты автоматизации организации - Современные инструменты и методы управления		ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента. ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3.

	организацией, в том числе методы планирования деятельности, распределения поручений, контроля исполнения, принятия решений		Владеет навыками автоматизации решения типовых задач для управления бизнесом. ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
06.015 C09/6 Адаптация бизнес-процессов заказчика ИС к возможностям ИС в рамках проекта создания (модификации) ИС	- Возможности типовой ИС - Инструменты и методы моделирования бизнес-процессов в ИС - Предметная область автоматизации - Устройство и функционирование современных ИС - Современные стандарты информационного взаимодействия систем - Современные стандарты информационного взаимодействия систем - Программные средства и платформы инфраструктуры информационных технологий организаций - Современные подходы и стандарты автоматизации организации - Современные инструменты и методы управления организацией, в	- Работать с типовой ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПК-1. Способность создавать, редактировать и наполнять контентом информационные ресурсы ПК-1.1. Проведение работ по созданию, редактированию и контролю за выгрузкой контента на сайт. ПК-1.2. Уметь проводить анализ потребностей пользователей информационных ресурсов предприятия. ПК-1.3. Владение методами оценки актуальности контента. ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владеет навыками автоматизации решения типовых задач для управления бизнесом. ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами ПК-4.1.

	том числе методы планирования деятельности, распределения поручений, контроля исполнения, принятия решений		Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3. Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами.
06.015 C16/6 Проектирование и дизайн ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	- Языки программирования и работы с базами данных - Инструменты и методы проектирования и дизайна ИС - Современные объектно-ориентированные языки программирования	- Кодировать на языках программирования в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС - Инструменты и методы проектирования и дизайна ИС	ПК-7 Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владеть методами разработки дизайна ИТ сервисов
06.015 C40/6 Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию		- Устанавливать права доступа к файлам и папкам в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС - Использовать системы контроля версий ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС - Просматривать права пользователей репозитория данных о выполнении работ по созданию (модификации) и сопровождению ИС - Управлять списком (добавлять, удалять, модифицировать) пользователей репозитория данных о выполнении работ по созданию (модификации) и сопровождению ИС	ПК-2. Способность разработки прикладного программного обеспечения для управления бизнесом ПК-2.1. Знать технологии разработки прикладного программного обеспечения и бизнес-логики приложений. ПК-2.2. Уметь использовать типовые технологические решения для управления бизнесом ПК-2.3. Владеет навыками автоматизации решения типовых задач для управления бизнесом. ПК-4. Умение осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами ПК-4.1. Знать основные стандарты управления проектами, методики планирования и организации проектной деятельности. ПК-4.2. Уметь использовать информационные системы для управления проектами. ПК-4.3.

			Владеть навыками планирования проектной деятельности и ее организации на основе стандартов управления проектами. ПК-7 Способность разрабатывать бизнес-приложения, используя высокоуровневые методы программирования ПК-7.1 Уметь использовать высокоуровневые методы программирования для разработки бизнес-приложений ПК-7.2 Знать высокоуровневые языки программирования и модели доступа к данным ПК-7.3 Владеть методами разработки дизайна ИТ сервисов
--	--	--	---

ЭТО ПРИМЕР ЗАКЛЮЧЕНИЯ. Пишем свое заключение по результатам прохождения преддипломной практики В ходе прохождения преддипломной практики при выполнении индивидуального задания и служебных поручений руководителя сформированы навыки выполнения трудовых действий в соответствие с профессиональными стандартами (табл. 5).

Таблица 5

Соответствие результатов прохождения преддипломной практики требованиям профессиональных стандартов в части сформированных навыков выполнения трудовых действий

Профессиональный стандарт Трудовая функция	Трудовые действия	Сформированные навыки как результат прохождения преддипломной практики
06.013 С01/6 Организация работ по созданию и редактированию контента сайтов	- Планирование работ по наполнению сайта - Подготовка заданий для исполнителей - Распределение работы по созданию и редактированию контента - Координация работы по созданию и редактированию контента	- подготовлен контент для информационного ресурса ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку
06.013 С03/6 Контроль за наполнением сайта	- Оценка полноты сайта, его разделов, представительств в социальных сетях и определение необходимости его пополнения - Оценка актуальности материалов на сайте и определение необходимости их обновления	- проведена оценка актуальности материалов xxxx хх xxxxx xxxx ххх xxxxxx ххх ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку

06.013 C04/6 Организация работ по изменению структуры сайта	- Выявление необходимости изменения структуры сайта или его разделов - Оценка возможности внесения локальных изменений, не требующих обращения к веб-мастеру - Изменение структуры сайта с помощью системы управления контентом - создание новых разделов, подразделов - Перемещение информационных ресурсов в новые разделы, удаление из существующих разделов	- проведен анализ существующего сайта, предложен план модификации xxxx xx xxxxx xxx xxxxxx xxx ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку
06.013 C05/6 Анализ информационных потребностей посетителей сайта	- Общий анализ посещаемости сайта, в том числе с использованием инструментальных средств - Выявление наиболее популярных страниц - Определение обобщенных характеристик аудитории сайта - Анализ оценки, замечаний, жалоб и предложений посетителей сайта - Оценка эффективности работы сайта на основе имеющихся данных - Выработка решения (рекомендации) по наполнению сайта контентом	- проведен анализ эффективности работы сайта xxxx xx xxxxx xxx xxxxxx xxx ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку
06.013 C07/6 Поддержка процессов модернизации и продвижения сайта	- Внутренняя поисковая оптимизация информационных ресурсов - Формирование предложений по развитию сайта - Поддержание процессов проектирования сайта и анализа требований пользователей, бизнес-требований, существующей структуры и содержания веб-сайта - Поддержание процессов разработки и тестирования новой функциональности веб-сайта, систем управления контентом - Поддержание процессов опытной эксплуатации веб-сайта	- проведена SEO оптимизация сайта ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку
06.015 C08/6 Разработка модели бизнес-процессов заказчика в рамках проекта создания (модификации) ИС 06.015 C09/6 Адаптация бизнес-процессов заказчика ИС к возможностям ИС в рамках проекта создания (модификации) ИС	- Сбор исходных данных у заказчика ИС о его бизнес-процессах в рамках проекта создания (модификации) ИС - Разработка модели бизнес-процессов заказчика ИС в рамках проекта создания (модификации) ИС - Согласование с заказчиком ИС модели его бизнес-процессов в рамках проекта создания (модификации) ИС	- проведено моделирование бизнес-процессов «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ» ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку

06.015 C16/6 Проектирование и дизайн ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	- Разработка структуры программного кода ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС - Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС - Разработка пользовательских интерфейсов ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	- разработана структура программного кода модулей корпоративной информационной системы xxxx xx xxxxx xxxx xxx xxxxxxx xxx - выполнение проектирование модели данных - разработана структура базы данных корпоративной информационной системы xxx xxxx xxxxx в соответствии с архитектурной спецификацией - проведена проверка структуры базы данных в соответствии с требованиями архитектуры разрабатываемой корпоративной информационной системы xxx xxxxxx - выполнено проектирование и разработка программных интерфейсов (программного обеспечения) - проведена верификация и тестирование программного кода xx xxxxx xxxx x - проведена корректировка обнаруженных ошибок в программном коде
06.015 C40/6 Организация репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию	- Создание репозитория для хранения базовых элементов конфигурации ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС - Определение прав доступа для репозитория хранения данных о создании (модификации) и вводе ИС в эксплуатацию	- подготовлен git-репозиторий на платформе Gitflic xxx xxxxxx ЭТО ПРИМЕР ФОРМУЛИРОВКИ. Пишем свои выводы по сформированным навыкам при прохождении преддипломной практики. Убираем заливку

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1 ЭТО ПРИМЕР ОФОРМЛЕНИЯ СПИСКА ЛИТЕРАТУРЫ – ТОЛЬКО АКТУАЛЬНЫЕ ИСТОЧНИКИ. ЛИТЕРАТУРА – «ЭПОХИ ВРЕМЁН ДИНОЗАВРОВ» - НЕ ДОПУСКАЕТСЯ!!!! Список литературы должен быть оформлен по ГОСТ Р 7.0.100-2018 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления». Источники должны соответствовать содержательной части отчета по преддипломной практике. Не нужно добавлять в список литературы источники по принципу «чтобы было». На каждый источник должна быть ссылка по тексту отчета согласно ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила составления». **НЕ СЛЕДУЕТ КОПИРОВАТЬ ЭТОТ СПИСОК В СВОЙ ОТЧЕТ.**

2 Провалов, В.С. Информационные технологии управления / В.С. Провалов. – 4-е изд., стер. – Москва: Издательство «Флинта», 2018. – 374 с. – (Экономика и управление). – Режим доступа: – URL: <http://biblioclub.ru/index.php?page=book&id=69111>

3 Кравченко, Ю.А. Информационные и программные технологии / Ю.А. Кравченко, Э.В. Кулиев, В.В. Марков; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет», Инженерно-технологическая академия. – Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2017. – Ч. 1. Информационные технологии. – 113 с.: ил. – Режим доступа: – URL: <http://biblioclub.ru/index.php?page=book&id=499727>

4 Парфенова М.Я., Маликов С.Н. Методологические исследования эффективности информационных технологий: монография. Издательство: МУ им. С.Ю. Витте, 2017. <https://online.muiiv.ru/lib/pdf/117050.pdf/>

5 Балдин, К.В. Информационные системы в экономике / К.В. Балдин, В.Б. Уткин. – 7-е изд. – Москва: Издательско-торговая корпорация «Дашков и К°», 2017. – 395 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=454036/>

6 Соловьев, Н.А. Введение в программную инженерию / Н.А. Соловьев, Л.А. Юркевская; Министерство образования и науки Российской Федерации, Оренбургский Государственный Университет. – Оренбург: ОГУ, 2017. – 112 с.: схем., табл., ил. – Режим доступа: – URL: <http://biblioclub.ru/index.php?page=book&id=481815/>