|  |  |
| --- | --- |
| Описание: Описание: Описание: http://almetpt.ru/img/emblema.png | Министерство образования и науки Республики Татарстан  Государственное автономное профессиональное образовательное учреждение  **«Альметьевский политехнический техникум»** (ГАПОУ «АПТ») |

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ**

**ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

МДК.02.01. Технология разработки программного обеспечения

по специальности 09.02.07 Информационные системы и программирование

2020 г.

|  |  |
| --- | --- |
| **ОДОБРЕНО** | **УТВЕРЖДАЮ** |
| Цикловой комиссией | Зам. директора по УР |
| информационно-коммуникационных  дисциплин | \_\_\_\_\_\_\_ Р.М.Бородина |
| Председатель | 05 сентября 2020г. |
| \_\_\_\_\_\_\_\_\_Ю.А.Рамазанова |  |
| \_\_\_\_ \_\_\_\_\_\_\_\_\_\_2020г. |  |
|  |  |

Составитель:

Филимонова А.Ю.. преподаватель информационных технологий ГАПОУ «Альметьевский политехнический техникум»;

Эксперты:

Внутренняя экспертиза

Техническая экспертиза: Захарова И.М., методист ГАПОУ «Альметьевский политехнический техникум»;

Содержательная экспертиза: Рамазанова Ю.А., председатель ЦК ГАПОУ «Альметьевский политехнический техникум».

Внешняя рецензия: Куприянов Д.А., ведущий инженер ООО «Процессинговый Центр», Центр АСУТП.

**Содержание**

[Лабораторная работа №1. Анализ предметной области 4](#_Toc64631974)

[Лабораторная работа №2-3. Разработка и оформление технического задания. 11](#_Toc64631975)

[Лабораторная работа №4. Разработка архитектуры программного средства. 21](#_Toc64631976)

[Лабораторная работа №5. Изучение работы в системе контроля версий 27](#_Toc64631977)

**Лабораторная работа №1****. Анализ предметной области**

**Цель работы**: провести анализ и оформить результаты обследования деятельности предприятия; осуществлять постановку задач по обработке информации; проводить анализ предметной области.

**Ход работы**

В процессе выполнения лабораторной работы проводится анализ и оформление результатов обследования деятельности гипотетического предприятия "МЕД", и на его основе разрабатываются документы, необходимые для настройки типовой ИС.

По итогам проведения обследования обычно формируется отчет об обследовании.

Обследование начинается со сбора предварительной информации о компании. Итогом являются следующие данные:

* краткая информация о компании (профиль клиента).
* цели проекта.
* подразделения и пользователи системы.

В процессе выполнения лабораторной работы проводится анализ и оформление результатов обследования деятельности гипотетического предприятия "МЕД", и на его основе разрабатываются документы, необходимые для настройки типовой ИС.

По итогам проведения обследования обычно формируется отчет об обследовании.

Обследование начинается со сбора предварительной информации о компании. Итогом являются следующие данные:

* краткая информация о компании (профиль клиента).
* цели проекта.
* подразделения и пользователи системы.

На основе предварительной информации формируется и согласовывается с заказчиком общее представление о проекте:

Видение выполнения проекта и границы проекта - документ, который кратко описывает, в каких подразделениях и в какой функциональности будет внедряться ИС.

Затем выполняется детальное обследование предприятия, результаты которого оформляются в виде отдельного документа - отчета об обследовании.

Отчет об обследовании содержит следующие разделы:

* анализ существующего уровня автоматизации.
* составляется список программного обеспечения, используемого в компании, и приводятся данные об использовании этих пакетов в каждом из подразделений организации.
* общие требования к ИС
* формулируются общие требования к функциональности разрабатываемой системы.
* организационная диаграмма

Организационная диаграмма используется для отражения организационной структуры подразделений предприятия и их зон ответственности.

* описание состава автоматизируемых бизнес-процессов

Все бизнес-процессы компании должны быть перечислены в общем списке и каждый должен иметь свой уникальный номер.

* диаграммы прецедентов.

Для выделения автоматизируемых бизнес-процессов и их основных исполнителей используются диаграммы прецедентов.

* физическая диаграмма

Физическая диаграмма служит для того, чтобы описать взаимодействие организации на верхнем уровне с внешними контрагентами.

* описания бизнес-процессов (книга бизнес-процессов).

Далее в отчет об обследовании включается книга бизнес-процессов, содержащая подробное описание автоматизируемых бизнес-процессов. Модели бизнес-процессов позволяют выделить отдельные операции, выполнение которых должно поддерживаться разрабатываемой ИС.

На последнем этапе осуществляется отображение модели предметной области на функциональность типовой системы - выбираются модули системы для поддержки выделенных операций, определяются особенности их настройки, выявляется необходимость разработки дополнительных программных элементов.

**Пример отчета обследования предметной области.**

Краткая информация о компании "МЕД"

Компания - дистрибьютор "МЕД" закупает медицинские препараты отечественных и зарубежных производителей и реализует их через собственную дистрибьюторскую сеть и сеть аптек. Компания осуществляет доставку товаров как собственным транспортом, так и с помощью услуг сторонних организаций.

Основные бизнес-процессы компании - закупки, складирование запасов, продажи, взаиморасчеты с поставщиками и клиентами.

Уровень конкуренции для компании в последнее время возрос, так как на рынок вышли два новых конкурента, к которым перешла часть клиентов и ряд наиболее квалифицированных сотрудников ЗАО "МЕД". ЗАО "МЕД" имеет два филиала - в Курске и Санкт-Петербурге. Каждый филиал функционирует как самостоятельное юридическое лицо, являясь полностью принадлежащей ЗАО "МЕД" дочерней компанией.

По предварительным планам, Компания намерена открыть также дочернее предприятие для организации производства в непосредственной близости к своим заказчикам.

Адреса и телефоны

Москва, К-123 Центральная улица, д. 20, стр. 7, офис 709

Телефон: (095) 345-6789, факс: (095) 345-9876

Контактные лица

Борис Нефедьев - Генеральный директор

Дмитрий Кононов - Исполнительный директор

Артур Иванченко - Директор по маркетингу

Сотрудники

На момент проведения Диагностики штат компании составляет 110 сотрудников.

Основными целями проекта автоматизации компании "МЕД" являются:

1. Разработка и внедрение комплексной автоматизированной системы поддержки логистических процессов компании.
2. Повышение эффективности работы всех подразделений компании и обеспечение ведения учета в единой информационной системе.

Видение выполнения проекта и границы проекта

В рамках проекта развертывание новой системы предполагается осуществить только в следующих подразделениях ЗАО "МЕД":

* Отдел закупок;
* Отдел приемки;
* Отдел продаж;
* Отдел маркетинга;
* Группа планирования и маркетинга;
* Группа логистики;
* Учетно-операционный отдел;
* Учетный отдел;
* Отдел сертификации (в части учета сертификатов на медикаменты);
* Бухгалтерия (только в части учета закупок, продаж, поступлений и платежей).

Не рассматривается в границах проекта автоматизация учета основных средств, расчета и начисления заработной платы, управления кадрами. Выходит за рамки проекта автоматизация процессов взаимоотношений с клиентами.

Количество рабочих мест пользователей - 50.

Отчет об обследовании

Список программного обеспечения, используемого компанией на момент обследования

1. "1С: Предприятие 7.7" ("Бухгалтерия", "Торговля", "Зарплата", "Кадры", "Касса", "Банк") для работы бухгалтерии.
2. Две собственные разработки на базе конфигуратора "1С" - "Закупки" и "Продажи".
3. Собственная разработка на базе FOXPRO для финансового отдела.
4. Excel для планирования продаж.

Существующий уровень автоматизации представлен в таблице 1.

Таблица 1 – Уровень автоматизации

|  |  |
| --- | --- |
| Количество рабочих станций, всего: | 90 |
| Количество сотрудников отдела IT | 2 |
| Количество ПК, одновременно работающих в сети | 50 |
| Наличие и форма связи с удаленными объектами | Терминальная связь со складом |
| Количество рабочих станций на удаленном объекте | 8 |
| Характеристики компьютеров | От Celeron 600 и выше |
| Операционная система | Windows 98, XP |
| Системы, которые представляется возможным оставить без изменения | "1С: Предприятие 7.7" в модульном составе "Бухгалтерия", "Зарплата", "Кадры", для работы бухгалтерии |

Общие требования к информационной системе

Одно из основных требований компании "МЕД" к будущему решению состоит в том, чтобы оно было построено на фундаменте единой интегрированной системы, а работа всех сотрудников велась в одном информационном пространстве.

Ключевые функциональные требования к информационной системе:

1. Мощные средства защиты данных от несанкционированного доступа. Разграничения доступа к данным в соответствии с должностными обязанностями.
2. Возможность удаленного доступа.
3. Управление запасами. Оперативное получение информации об остатках на складе.
4. Управление закупками. Планирование закупок в разрезе поставщиков.
5. Управление продажами. Контроль лимита задолженности с возможностью блокировки формирования отгрузочных документов.
6. Полный контроль взаиморасчетов с поставщиками и клиентами.
7. Получение управленческих отчетов в необходимых аналитических срезах - как детальных для менеджеров, так и агрегированных, для руководителей подразделений и директоров фирмы.

Организационная диаграмма

Оргструктура предприятия оптовой торговли ЗАО "МЕД" имеет вид, представленный в соответствии с рисунком 1.

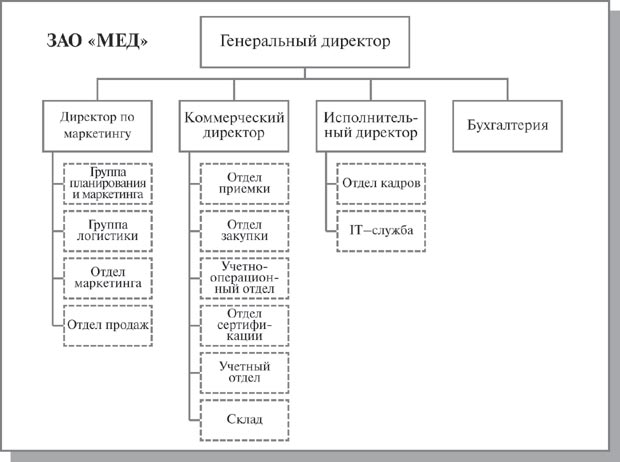


Рисунок 1 – Организационная структура предприятия ЗАО «МЕД»

Описание состава автоматизируемых бизнес-процессов

Бизнес-процессы компании, подлежащие автоматизации, приведены в следующей таблице 6.

Таблица 6 – Бизнес-процессы компании

|  |  |  |
| --- | --- | --- |
| **п.п** | **Код бизнес-процесса** | **Наименование бизнес-процесса** |
| 1. | Закуп-1 | Закупки |
| 2. | Склад-2 | Запасы-Склад |
| 3.  4. | Прод-3  Врасч-4 | Продажи  Взаиморасчеты с поставщиками и клиентами |

Каждый бизнес-процесс имеет свой уникальный номер. Нумерация бизнес-процессов построена по следующему принципу: "префикс-номер", где префикс обозначает группу описываемых бизнес-процессов, а номер - порядковый номер бизнес-процесса в списке.

**Задание для самостоятельного выполнения.**

Задание: провести анализ предметной области по вариантам. Номер варианта соответствует номеру в списке в классном журнале. Список предметных областей приведен в ниже:

1.Компания по добычи нефти и газа

2.Компания по транспортировке нефти и газа

3.Компания по проведению подземного и капитального ремонта скважин

4.Компания по производству и продаже оборудования для добычи нефти и газа

5.Компания по проведению геофизических исследований скважин

6. Компания по производству и продаже оборудования для проведения ремонта скважин

7.Компания по проведению бурения скважин

8. Компания по производству и продаже оборудования для геофизических исследований скважин

9. Компания по разведке месторождений нефти и газа

10.Компания по хранению и складированию нефти и продуктов ее переработки.

11. Компания по исследованию полезных ископаемых и горных пород.

**Требования к отчету**

На протяжении лабораторной работы должен быть получен отчет об обследовании предметной области, содержащий разделы, перечисленные выше.

**Лабораторная работа №2-3. Разработка и оформление технического задания.**

**Цель работы:** Ознакомиться с процедурой разработки технического задания на создание программного продукта (ПП) с применением ГОСТ 19.102-77 «Стадии разработки программ и программной документации» и ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы».

**Основные теоретические сведения**

*Техническое задание* — это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления.

*Техническое задание* представляет собой документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемо-сдаточных испытаний. В разработке технического задания участвуют как представители заказчика, так и представители исполнителя. В основе этого документа лежат исходные требования заказчика, анализ передовых достижений техники, результаты выполнения научно-исследовательских работ, предпроектных исследований, научного прогнозирования и т. п.

При разработке технического задания (ТЗ) необходимо решить следующие задачи:

* установить общую цель создания АИС;
* установить общие требования к проектируемой системе;
* разработать и обосновать требования, предъявляемые к информационному, математическому, программному, техническому и технологическому обеспечению;
* определить состав подсистем и функциональных задач;
* разработать и обосновать требования, предъявляемые к подсистемам; − определить этапы создания системы и сроки их выполнения;
* провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения;
* определить состав исполнителей.

В Российской Федерации действует ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы», также на техническое задание существует стандарт ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению».

ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам устанавливает общие требования к оформлению программных документов. Программный документ должен состоять из следующих частей:

* Титульной;
* Информационной;
* Основной.

Титульная часть оформляется согласно ГОСТ 19.104-78 ЕСПД. Основные надписи.

Информационная часть должна состоять из аннотации и содержания. В аннотации приводят сведения о назначении документа и краткое изложение основной части.

Содержание включает перечень записей о структурных элементах основной части документа.

Состав и структура основной части программного документа устанавливается стандартами ЕСПД на соответствующие документы.

Основная часть технического задания должна содержать следующие разделы:

**ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению**

* введение;
* основания для разработки;
* назначение разработки;
* требования к программному продукту;
* требования к программной документации;
* технико-экономические показатели; − стадии и этапы разработки;
* порядок контроля и приемки;

В зависимости от программного продукта допускается уточнять содержание разделов, объединять отдельные из них, вводить новые разделы. В техническое задание допускается включать приложения.

**Состав и содержание технического задания (ГОСТ 34.602—89)**

|  |  |
| --- | --- |
| Раздел | Содержание |
| Общие сведения | Полное наименование системы и ее условное обозначение. Шифр темы или шифр (номер) договора. Наименование предприятий разработчика и заказчика системы, их реквизиты. Перечень документов, на основании которых создается ИС. Плановые сроки начала и окончания работ. Сведения об источниках и порядке финансирования работ. Порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств |
| Назначение и цепи создания (развития) системы | Вид автоматизируемой деятельности. Перечень объектов, на которых предполагается использование системы. Наименования и требуемые значения технических, технологических, производственно-экономических и др. показателей объекта, которые должны быть достигнуты при внедрении 1/1С |
| Характеристика объектов автоматизации | Краткие сведения об объекте автоматизации. Сведения об условиях эксплуатации и характеристиках окружающей среды |
| Требования к системе | Требования к системе в целом:  − требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы);  − требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки);  − показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров)  − требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации.  Требования к функциям (по подсистемам):  − перечень подлежащих автоматизации задач;  − временной регламент реализации каждой функции;  − требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов;  − перечень и критерии отказов.  Требования к видам обеспечения:  − математическому (состав и область применения математических моделей и методов, типовых и разрабатываемых алгоритмов);  − информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам);  − лингвистическому (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода-вывода);  − программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ);  − техническому;  − метрологическому;  − организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала);  − методическому (состав нормативно-технической документации) |
| Состав и содержание работ по созданию системы | Перечень стадий и этапов работ. Сроки исполнения. Состав организаций-исполнителей работ. Вид и порядок экспертизы технической документации. Программа обеспечения надежности. Программа метрологического обеспечения |
| Порядок контроля и приемки системы | Виды, состав, объем и методы испытаний системы. Общие требования к приемке работ по стадиям. Статус приемной комиссии |
| Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие | Преобразование входной информации к машиночитаемому виду. Изменения в объекте автоматизации. Сроки и порядок комплектования и обучения персонала |
| Требования к документированию | Перечень подлежащих разработке документов. Перечень документов на машинных носителях |
| Источники разработки | Документы и информационные материалы, на основании которых разрабатывается ТЗ и система |

**Задания для выполнения**

1. Разработать техническое задание на программный продукт (см. варианты заданий) в соответствии с ГОСТ 19.201-78 и ГОСТ 34.602—89
2. Оформить работу в соответствии с ГОСТ 19.106—78. При оформлении использовать MS Office или OpenOffice.org.
3. Сдать и защитить работу

**Содержание и оформление отчета по лабораторной работе**

Отчет по лабораторной работе должен состоять из:

1. Постановки задачи.

2. Технического задания на программный продукт. Отчет должен содержать титульный лист, аннотацию, содержание и основную часть, оформленную в соответствии с ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» или ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению». В техническое задание в раздел календарного плана включить сетевой график/диаграмму Ганта, выполненную с помощью одного из инструментальных средств

Защита отчета по лабораторной работе заключается в предъявлении преподавателю полученных результатов (на экране монитора или у доски), демонстрации полученных навыков и ответах на вопросы преподавателя.

**Пример разработки технического задания на программный продукт**

Разработать техническое задание на разработку «Модуля автоматизированной системы оперативно-диспетчерского управления теплоснабжением корпусов Московского института».

|  |
| --- |
| «Утверждаю»  Зам. директора по УПР  \_\_\_\_\_\_\_\_ Иванов И.И.  «\_\_» \_\_\_\_\_\_\_201\_ г.  **Техническое задание**  на разработку «Модуля автоматизированной системы оперативно-диспетчерского управления теплоснабжением корпусов Московского института»  г.Москва, 2013 |

1. **Введение**

Работа выполняется в рамках проекта «Автоматизированная система оперативно- диспетчерского управления электротеплоснабжением корпусов Московского института».

**2. Основание для разработки**

2.1. Основанием для данной работы служит договор № 1234 от 10 марта 2013 г.

2.2. Наименование работы: «Модуль автоматизированной системы оперативно-диспетчерского управления теплоснабжением корпусов Московского института».

2.3. Исполнители: ОАО «Лаборатория создания программного обеспечения».

2.4. Соисполнители: нет.

**3. Назначение разработки**

Создание модуля для контроля и оперативной корректировки состояния основных параметров теплообеспечения корпусов Московского института.

**4. Технические требования**

4.1. Требования к функциональным характеристикам.

4.1.1. Состав выполняемых функций. Разрабатываемое ПО должно обеспечивать:

− сбор и анализ информации о расходовании тепла, горячей и холодной воды по данным теплосчетчиков SA-94 на всех тепловых выходах;

− сбор и анализ информации с устройств управления системами воздушного отопления и кондиционирования типа РТ1 и РТ2 (разработки кафедры СММЭ и ТЦ);

− предварительный анализ информации на предмет нахождения параметров в допустимых пределах и сигнализирование при выходе параметров за пределы допуска;

− выдачу рекомендаций по дальнейшей работе;

− отображение текущего состояния по набору параметров - циклически постоянно (режим работы круглосуточный), при сохранении периодичности контроля прочих пара метров;

− визуализацию информации по расходу теплоносителя: - текущую, аналогично показаниям счетчиков;

− с накоплением за прошедшие сутки, неделю, месяц - в виде почасового графика для информации за сутки и неделю;

− суточный расход — для информации за месяц.

Для устройств управления приточной вентиляцией текущая информация должна содержать номер приточной системы и все параметры, выдаваемые на собственный индикатор.

По отдельному запросу осуществляются внутренние настройки.

В конце отчетного периода система должна архивировать данные.

4.1.2. Организация входных и выходных данных.

Исходные данные в систему поступают в виде значений с датчиков, установленных в помещениях института. Эти значения отображаются на компьютере диспетчера. После анализа поступившей информации оператор диспетчерского пункта устанавливает необходимые параметры для устройств, регулирующих отопление и вентиляцию в помещениях. Возможна также автоматическая установка некоторых параметров для устройств регулирования.

Основной режим использования системы — ежедневная работа.

4.2. Требования к надежности.

Для обеспечения надежности необходимо проверять корректность получаемых данных с датчиков.

4.3. Условия эксплуатации и требования к составу и параметрам технических средств.

Для работы системы должен быть выделен ответственный оператор. Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования системы.

4.4. Требования к информационной и программной совместимости. 28 Программа должна работать на платформах Windows 98/ NT/2000.

4.5. Требования к транспортировке и хранению. Программа поставляется на лазерном носителе информации.

Программная документация поставляется в электронном и печатном виде.

4.6. Специальные требования:

− программное обеспечение должно иметь дружественный интерфейс, рассчитанный на пользователя (в плане компьютерной грамотности) квалификации;

− ввиду объемности проекта задачи предполагается решать поэтапно, при этом модули ПО, созданные в разное время, должны предполагать возможность наращивания системы и быть совместимы друг с другом, поэтому документация на принятое эксплуатационное ПО должна содержать полную информацию, необходимую для работы программистов с ним;

− язык программирования - по выбору исполнителя, должен обеспечивать возможность интеграции программного обеспечения с некоторыми видами периферийного оборудования (например, счетчик SA-94 и т. п.).

**5. Требования к программной документации**

Основными документами, регламентирующими разработку будущих программ, должны быть документы Единой Системы Программной Документации (ЕСПД): руководство пользователя, руководство администратора, описание применения.

**6. Технико-экономические показатели**

Эффективность системы определяется удобством использования системы для контроля и управления основными параметрами теплообеспечения помещений Московского института, а также экономической выгодой, полученной от внедрения аппаратно-программного комплекса.

**7. Порядок контроля и приемки**

После передачи Исполнителем отдельного функционального модуля программы Заказчику последний имеет право тестировать модуль в течение 7 дней. После тестирования Заказчик должен принять работу по данному этапу или в письменном виде изложить причину отказа принятия. В случае обоснованного отказа Исполнитель обязуется доработать модуль.

**8. Календарный план работ**

|  |  |  |  |
| --- | --- | --- | --- |
| № этапа | Название этапа | Сроки этапа | Чем заканчивается этап |
| 1 | Изучение предметной области. Проектирование системы. Разработка предложений по реализации системы | 01.02.200\_- 28.02.200\_ | Предложения по работе системы. Акт сдачи-приемки |
| 2 | Разработка программного модуля по сбору и анализу информации со счетчиков и устройств управления. Внедрение системы для одного из корпусов МИЭТ | 01.03.200\_- 31.08.200\_ | Программный комплекс, решающий поставленные задачи для пилотного корпуса МИЭТ. Акт сдачи-приемки |
| 3 | Тестирование и отладка модуля. Внедрение системы во всех корпусах МИЭТ | 01.09.200\_- 30.12.200\_ | Готовая система контроля теплообеспечения МИЭТ, установленная в диспетчерском пункте. Программная документация. Акт сдачи - приемки работ |

Руководитель работ Григорьева Г. Д.

**Лабораторная работа №4. Разработка архитектуры программного средства.**

**Цель работы:** реализация начальных этапов процесса разработки программного средства в соответствии с ГОСТ Р ИСО/МЭК 12207

1. **Краткие теоретические сведения**

При возникновении потребностей в заказе, приобретении, разработке, эксплуатации и сопровождении программ перед всеми сторонами, вовлеченными в жизненный цикл программного средства (ПС), возникает целый ряд вопросов, связанных с определением и детальным структурированием жизненного цикла (ЖЦ) ПС, с организационными и техническими правами и обязанностями сторон, с управлением ЖЦ и контролем за его реализацией. Одним из действенных инструментов для решения данных вопросов является использование унифицированных подходов, закрепленных в современных международных и российских стандартах.

Понятия «жизненный цикл системы» или «жизненный цикл программного средства» часто появляются в статьях и звучат в разговорах разработчиков, по крайней мере руководителей проектов и подразделений. Всем понятно, что относятся они к тому, что и в какой последовательности должно делаться при создании и эксплуатации систем. Но прежде чем две организации или два специалиста договорятся о том, что конкретно входит или не входит в ЖЦ, проходит значительное время. А позже вполне может обнаружиться, что эти двое (две «стороны») все-таки по-разному понимают, какие работы будут входить в ЖЦ, а какие - нет, какие проверки будут планироваться, когда и т. д. Естественно, общие принципы организации работ описаны давно, но что делать сторонам в конкретном проекте — это каждый раз приходится решать заново.

В стандартах, регламентирующих жизненный цикл программных средств, обобщаются опыт и результаты исследований множества специалистов и рекомендуются наиболее эффективные современные методы и процессы создания и развития комплексов программ. В результате таких обобщений оттачиваются технологические процессы и приемы разработки, а также методическая база для их автоматизации.

ЖЦ ПС в стандартах представляет собой набор этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение работ от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС или информационной системы (ИС).

Стандарты включают правила описания исходной информации, способов и методов выполнения операций, устанавливают правила контроля технологических процессов, требования к оформлению их результатов, а также регламентируют содержание технологических и эксплуатационных документов на комплексы программ. Они определяют организационную структуру коллектива, обеспечивают распределение и планирование заданий, а также контроль за ходом создания ПС.

Кроме вопросов выбора типа общего устройства ЖЦ есть проблемы с решением частных вопросов о включении или невключении в ЖЦ отдельных работ, очень важных для качества ПС и системы: что документировать при создании системы и ПС, какие работы должны будут гарантировать качество продукта, с какой степенью организационной независимости должны выполняться проверочные процедуры разных типов, чем будет обеспечиваться соответствие разрабатываемого ПС требованиям ко всей системе и соответствие ПС потребностям в системе.

Для того чтобы привнести порядок и понимание, общие для любых сторон, участвующих в ЖЦ систем и ПС, давно разрабатывались стандарты различных уровней утверждения - национальные и международные.

В России основы построения и использования профилей стандартов ЖЦ ПС заложены принятием в качестве базового стандарта ГОСТ Р ИСО/МЭК 12207. Данный документ введен в действие с 1 июля 2000 г., тесно взаимоувязан с рядом стандартов, принятых ранее, и с некоторыми стандартами, разрабатываемыми в данное время на основе прямого применения стандартов ИСО.

Актуальность стандарта ГОСТ Р ИСО/МЭК 12207 для современных условий настолько высока, что принятие в ISO его исходного, международного варианта вскоре вызвало самую положительную оценку российских экспертов. Был дан ряд рекомендаций, но его использованию в реальных условиях.

В данном стандарте **программное обеспечение** (ПО) или программный продукт определяется как набор компьютерных программ,процедур и,возможно, связанной с ними документации и данных.

**Процесс** определяется как совокупность взаимосвязанных действий*,* преобразующих некоторые входные данные в выходные. Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными от других процессов, и результатами.

В соответствии с ГОСТ Р ИСО/МЭК 12207 все процессы ЖЦ ПО разделены на три **группы:**

**1)Основные процессы:**

−приобретение;

−поставка;

−разработка;

−эксплуатация;

−сопровождение.

**2)Вспомогательные процессы:**

−документирование;

−управление конфигурацией;

−обеспечение качества;

−верификация;

−аттестация;

−совместная оценка;

−аудит;

−разрешение проблем.

**3)Организационные процессы:**

−управление;

−усовершенствование;

−создание инфраструктуры;

−обучение.

**Процесс разработки** предусматривает действия и задачи, выполняемые разработчиком, и включает следующие действия:

А) **Подготовительная работа** начинается с выбора модели ЖЦ ПО, соответствующей масштабу, значимости и сложности проекта. Действия и задачи процесса должны соответствовать выбранной модели. Разработчик должен выбрать, адаптировать к условиям проекта и использовать согласованные с заказчиком стандарты, методы и средства разработки, а также составить план выполнения работ.

Б) **Анализ требований к системе** подразумевает определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам и т.д. Требования к системе оцениваются исходя из критериев реализуемости и возможности проверки при тестировании.

Анализ требований к ПО предполагает определение следующих характеристик для каждого компонента:

−функциональных возможностей, включая характеристики производительности и среды функционирования компонента;

−внешних интерфейсов;

−спецификаций надежности и безопасности;

−эргономических требований;

−требований к используемым данным;

−требований к установке и приемке;

−требований к пользовательской документации;

−требований к эксплуатации и сопровождению.

Требования к ПО оцениваются исходя из критериев соответствия требованиям к системе, реализуемости и возможности проверки при тестировании.

В) **Проектирование архитектуры системы** на высоком уровне заключается в определении компонентов ее оборудования, ПО и операций, выполняемых эксплуатирующим систему персоналом. Архитектура системы должна соответствовать требованиям, предъявляемым к системе, а также принятым проектным стандартам и методам.

Проектирование архитектуры ПО включает **задачи** (для каждого компонента ПО):

−трансформацию требований к ПО в архитектуру, определяющую на высоком уровне структуру ПО и состав ее компонентов;

−разработку и документирование программных интерфейсов ПО и баз данных;

−разработку предварительной версии пользовательской документации;

−разработку и документирование предварительных требований к тестам и планам интеграции ПО.

Архитектура компонентов ПО должна соответствовать требованиям, предъявляемым к ним, а также принятым проектным стандартам и методам.

Г) **Детальное проектирование ПО** включает следующие задачи:

−описание компонентов и интерфейсов между ними на более низком уровне, достаточном для их последующего самостоятельного кодирования и тестирования;

−разработку и документирование детального проекта базы данных;

−обновление (при необходимости) пользовательской документации;

−разработку и документирование требований к тестам и плана тестирования компонентов ПО;

−обновление плана интеграции ПО.

Д) **Кодирование и тестирование ПО** охватывает задачи:

−разработку и документирование каждого компонента ПО и базы данных, а также совокупности тестовых процедур и данных для их тестирования;

−тестирование каждого компонента ПО и базы данных на соответствие предъявляемых к ним требованиям. Результаты тестирования компонентов должны быть документированы;

−обновление (при необходимости) пользовательской документации;

−обновление плана интеграции ПО.

Е) **Интеграция ПО** предусматривает сборку разработанных компонентов ПО в соответствии с планом интеграции и тестирование агрегированных компонентов. Для каждого из агрегированных компонентов разрабатываются наборы тестов и тестовые процедуры, предназначенные для проверки каждого из квалификационных требований при последующем квалификационном тестировании.

**Интеграция системы** заключается в сборке всех ее компонентов, включая ПО и оборудование. После интеграции система, в свою очередь, подвергается квалификационному тестированию на соответствие совокупности требований к ней. При этом также производится оформление и проверка полного комплекта документации на систему.

Ж) **Квалификационное тестирование** - это набор критериев и условий,

которые необходимо выполнить, чтобы квалифицировать программный продукт как соответствующий своим спецификациям и готовый к использованию в условиях эксплуатации.

**Квалификационное тестирование ПО** проводится разработчиком в присутствии заказчика (по возможности) для демонстрации того, что ПО удовлетворяет своим спецификациям и готово к использованию в условиях эксплуатации. Квалификационное тестирование выполняется для каждого компонента ПО по всем разделам требований при широком варьировании тестов. При этом также проверяются полнота технической и пользовательской документации и ее адекватность самим компонентам ПО.

З) **Установка ПО** осуществляется разработчиком в соответствии с планом в той среде и на том оборудовании, которые предусмотрены договором. В процессе установки проверяется работоспособность ПО и баз данных. Если устанавливаемое программное обеспечение заменяет существующую систему, разработчик должен обеспечить их параллельное функционирование в соответствии с договором.

И) **Приемка ПО** предусматривает оценку результатов квалификационного тестирования ПО и системы и документирование результатов оценки, которые проводятся заказчиком с помощью разработчика. Разработчик выполняет окончательную передачу ПО заказчику в соответствии с договором, обеспечивая при этом необходимое обучение и поддержку.

**2. Задание:** разработать проект архитектуры программного средства в соответствии с ГОСТ Р ИСО/МЭК 12207

**Алгоритм выполнения работы**

С целью реализации начальных этапов разработки ПС в соответствии с техническим заданием:

−выполнить подготовительную работу;

−провести анализ требований к ПС;

−выполнить проектирование архитектуры ПС на высоком уровне.

# **Лабораторная работа №5. Изучение работы в системе контроля версий**

**Цель работы:** ознакомиться с системой контроля версий Git.

1. **Краткие теоретические сведения**

**Git** — это система управления версиями, которая пришлась по душе практически всем — от разработчиков до дизайнеров. **GitHub** можно считать соцсетью для хранения кода. Это настоящая Мекка для технарей. Здесь вы можете попрактиковаться в разработке и придумать что-то свое, найти множество open-source проектов, передовых технологий, различных функций и дизайнов.

Для начала необходимо запомнить следующие терминальные команды:

git clone

git status

git add

git commit -m “ “

git push

Затем к ним добавим еще вот эти:

git init

git branch

git merge

git checkout

Эти команды вам пригодятся в случае, если вы будете работать с другими людьми или захотите внести какие-то изменения в проект и протестировать их до создания коммита.

Не лишней будет и вот такая команда:

git help

**Задание.**

**Шаг 1: Регистрация и установка**

Зайдите на [GitHub](https://github.com/?source=post_page---------------------------) и создайте свой аккаунт. В принципе, этим можно и ограничиться. При желании можете [установить Git](https://git-scm.com/downloads?source=post_page---------------------------). Но для работы с GitHub это вовсе не обязательно. Однако если вы планируете заниматься проектами на локальном компьютере, то установка вам все-таки нужна. Можете скачать установщик или [установить файлы через менеджер пакетов](https://gist.github.com/derhuerst/1b15ff4652a867391f03?source=post_page---------------------------).

Теперь перейдите в терминал, и начнем работу. Если хотите задать одно имя пользователя для **всех репозиториев** на компьютере, то напишите:

git config — global user.name “<ваше\_имя>”

замените <ваше\_имя> на свое имя в кавычках. Можете написать все, что угодно. Если хотите задать имя только для одного репозитория, то удалите из команды слово global.

Теперь напишите свой адрес электронной почты. Проследите, чтобы он совпадал с адресом, указанным при регистрации на GitHub.

git config — global user.email “<[адрес\_почты@email.com](mailto:%D0%B0%D0%B4%D1%80%D0%B5%D1%81_%D0%BF%D0%BE%D1%87%D1%82%D1%8B@email.com)>”

При желании можете скрыть свой электронный адрес. Это сделать несложно, подробнее написано [здесь](https://help.github.com/en/articles/blocking-command-line-pushes-that-expose-your-personal-email-address?source=post_page---------------------------). По сути, вам нужно проставить 2 галочки в своем GitHub-аккаунте.

**Теперь вы готовы к работе с Git на локальном компьютере.**

Начнем с создания нового репозитория на сайте GitHub. Вы также можете выполнить git init и создать новый репозиторий из директории проекта.

Репозиторий состоит из трех «деревьев». Первое «дерево» — это **рабочая директория,** в которой хранятся актуальные файлы. Второе — это **index** или область подготовленных файлов. А еще есть **head** — указатель на ваш последний коммит.

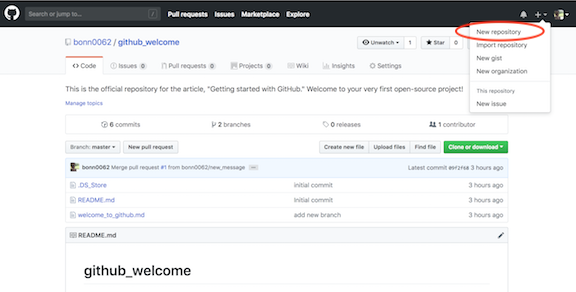
**Шаг 2. Создание репозитория**

Допустим, вы хотите создать новый репозиторий. Это место, где будет «жить» ваш проект. Если вы не хотите создавать новый репозиторий, то можете склонировать уже существующий. Именно так вы копируете чужой проект или берете нужную вам информацию для работы/учебы. Мы еще к этому вернемся, но чуть позже.

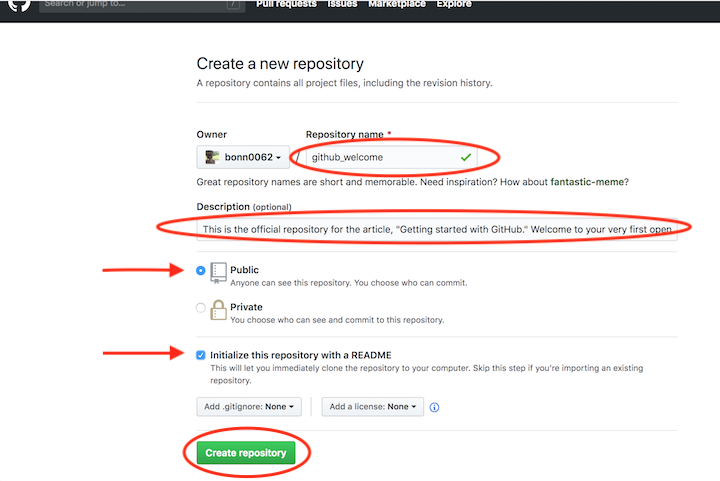
**Репозиторий** — это место, в котором вы систематизируете свой проект. Здесь вы храните файлы, папки, видео, изображения, блокноты Jupyter Notebook, наборы данных и т.д. Перед началом работы с Git необходимо инициализировать репозиторий для проекта и правильно его подготовить. Это можно сделать на сайте GitHub.

Лучше сразу добавлять в репозиторий **README**-файл с информацией о проекте. Это можно сделать в момент создания репозитория, поставив галочку в соответствующем поле.

* Перейдите на сайт GitHub. Нажмите на значок **+** в верхнем правом углу, а затем выберите **New repository**.
* Придумайте имя репозитория и добавьте короткое описание.
* Решите, будет ли этот репозиторий размещаться в открытом доступе или останется закрытым для просмотра.
* Нажмите **Initialize this repository with a README** для добавления README-файла. Настоятельно рекомендую снабжать все ваши проекты файлом-описанием, ведь README — это первая вещь, на которую люди обращают внимание при просмотре репозитория. К тому же, здесь можно разместить нужную информацию для понимания или запуска проекта.



Новый репозиторий



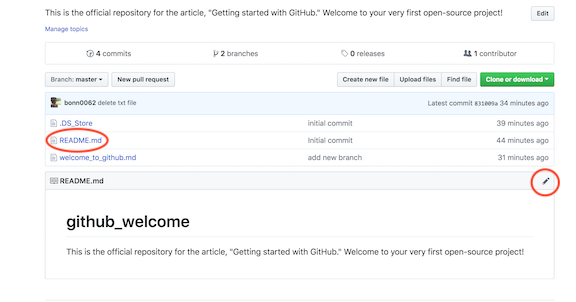
Создание нового репозитория

При желании можете уже сейчас начинать работать над проектом. Добавляйте файлы, вносите в них изменения и т.д. напрямую с сайта GitHub. Однако конечный результат подобной деятельности может вас немного огорчить.

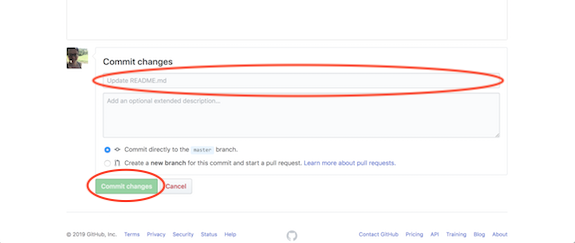
Вносить изменения в проект можно двумя способами. Вы можете изменять файлы/блокноты на компьютере либо делать это на сайте GitHub.

Допустим, вам захотелось подкорректировать README-файл на сайте GitHub.

* Для начала перейдите в ваш репозиторий.
* Для выбора файла кликните по его названию (например, кликните по **README.md** для перехода к файлу-описанию).
* В верхнем правом углу вы увидите иконку с карандашом. Нажмите на нее для внесения изменений.
* Напишите короткое сообщение, передающее суть изменений (и подробное описание, если сочтете это нужным).
* Нажмите кнопку **Commit changes**.



Изменение файла на GitHub

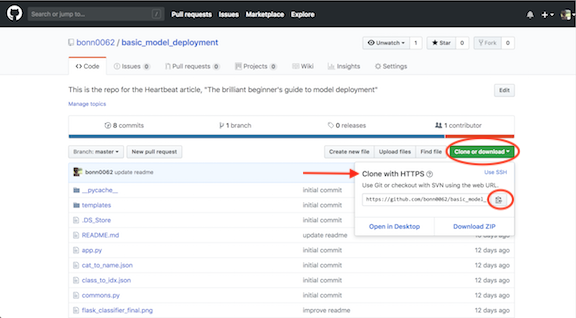


Подготовка коммита с изменениями

Вы успешно внесли изменения в README-файл своего нового репозитория! Обратите внимание на небольшую кнопку на картинке выше. Она позволяет создавать новую ветку этого коммита и добавлять Pull request. Запомните ее, скоро к ней вернемся.

Возможно, вы захотите клонировать свой новый репозиторий для дальнейшей работы с ним на локальном компьютере. Либо у вас уже есть существующий репозиторий, который вы хотели бы клонировать.

Для **клонирования репозитория** на компьютер перейдите в репозиторий на GitHub и нажмите большую зеленую кнопку под названием **Clone or download.** Проследите, чтобы появилась надпись **Clone with HTTPS**. Теперь нажмите на иконку буфера обмена для копирования-вставки (либо выделите ссылку и скопируйте ее).



Клонирование или скачивание репозитория

Откройте **терминал** и перейдите в директорию для копирования репозитория. Например, для перехода на **Рабочий стол** напечатайте вот это:

cd Desktop

Затем клонируйте туда репозиторий по следующей команде:

git clone <то,\_что\_вы\_только\_что\_скопировали>

Все просто! Не забудьте изменить информацию в угловых скобках на нужную вам. И удалите сами скобки **< >.**

*Если вы не очень хорошо ориентируетесь в терминале, то переход по директориям можно осуществлять через команду* ***cd****. Например, откройте терминал и напечатайте* ***ls*** *для отображения перечня доступных директорий. Вполне возможно, что в этом списке вы сразу увидите директорию* ***Desktop****. Либо напечатайте* ***cd******Desktop****. Далее выполните команду* ***git******clone*** *и склонируйте репозиторий на Рабочий стол.*

*Бывает и так, что вместо перечня расположений, вы видите различные имена пользователей. Тогда до того, как перейти в* ***Desktop****, вам потребуется выбрать нужного пользователя через команду* ***cd <пользователь>*** *(замените* ***<пользователь>*** *на нужное вам имя). Затем снова напечатайте* ***ls****, чтобы увидеть весь список. И вот теперь, увидев в списке* ***Desktop****, смело печатайте* ***cd******Desktop****. Сейчас уже можно выполнять* ***git******clone****!*

*Если вдруг в терминале вы захотите «откатиться» на шаг назад, то напишите* ***cd*** *..*

Новый GitHub-репозиторий, склонированный на рабочий стол, готов! Данная команда создает точную копию репозитория в вашей системе. Здесь вы сможете с ним работать, редактировать, индексировать изменения, создавать коммиты с изменениями и отправлять их на GitHub.

*Совсем не обязательно создавать репозиторий на Рабочем столе. Клонировать можно в любое место на компьютере. Команду* ***git******clone*** *можно выполнять и сразу после открытия терминала. Однако, если вы не очень любите копаться в папках на компьютере, то неплохо будет разместить проект на виду, то есть на Рабочем столе…*

Если хотите просто покопаться в каком-то проекте, то вместо клонирования можете сделать **форк** проекта на GitHub. Для этого нажмите кнопку **Fork** в верхнем правом углу сайта. Так вы добавите копию этого проекта в свои репозитории и сможете вносить туда любые изменения без вреда для оригинала.

Добавляем файлы в проект

Вот, чем мы займемся:

git status

git add

git commit -m “ “

git push

Но ничего сложного здесь нет!

Должно быть, у вас уже есть файлы, которые вы бы хотели разместить в новом репозитории. Отыщите их на компьютере и перетащите в новую папку репозитория на Рабочем столе.

Проверьте **статус** проекта.

Откройте терминал и перейдите в папку репозитория. Для проверки обновлений выполните:

git status

Если вы перетаскивали файлы в папку проекта, то потребуется обновить состояние репозитория. Добавлять файлы в репозиторий можно по одному:

git add <имя\_файла>

Либо все сразу:

git add — all

или даже:

git add .

Это ваши предлагаемые изменения. Операцию можно повторить с новыми файлами либо с уже существующими, но измененными. По сути, ничего нового в сам проект вы не добавляете. Вы всего лишь загружаете новые файлы и указываете Git на эти изменения.

Процесс создания коммитов с изменениями начинается с выполнения команды:

git commit -m “<сообщение\_о\_коммите>”

Коммиты изменений добавляются в **head** (указатель), а не в удаленный репозиторий. Не забудьте заменить текст в скобках и убрать **<>.** После внесения изменений создается снимок состояния репозитория, для чего используется команда **commit**. А через **–m** добавляется сообщение об этом снимке.

Сохраненные изменения и называются коммитом. При создании коммита вы добавляете сообщение о том, что именно менялось и почему. Так другие люди смогут лучше понять суть изменений.

Теперь ваши изменения сохранены в указателе локальной копии проекта. Для отправки изменений на удаленный репозиторий выполните команду:

git push

Тем самым вы отправляете изменения напрямую в репозиторий. Если вы работаете на локальном компьютере и хотите, чтобы коммиты отображались в онлайн, то необходимо своевременно отправлять эти изменения на GitHub по команде **git push**.

Актуальность версии можно проверить в любое время через команду **git status.**

Итог: у вас есть свой GitHub репозиторий, вы научились добавлять и изменять в нем файлы.

**Требования к отчету:**

Отчет по лабораторной работе должен содержать краткое описание порядка выполнения работы, так же должны присутствовать скриншоты вашей работы.