

Министерство образования Иркутской области
Государственное бюджетное профессиональное образовательное учреждение
Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

Рассмотрено
на заседании ВЦК ПКС
Протокол № 8 от 17.01.2017 г.



МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

09.02.03 Программирование в компьютерных системах

Иркутск 2017

Содержание

Введение	3
1 Общие положения	4
1.1 Цели и задачи дипломного проектирования	5
1.2 Выбор темы дипломного проекта	5
2 Структура дипломного проекта	7
2.1 Задание на дипломное проектирование	7
2.2 Содержание пояснительной записки	7
2.3 Оформление пояснительной записки	22
Приложение А Пример оформления бланк-заказа на разработку программного продукта	23
Приложение Б Пример оформления бланка о внедрении программного продукта	24
Приложение В Образец оформления задания на дипломное проектирование	25
Приложение Г Образец оформления титульного листа	27
Приложение Д Пример оформления содержания	28

Введение

Настоящие методические указания к выполнению дипломного проекта по специальности 09.02.03 Программирование в компьютерных системах написаны в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования к минимуму содержания и уровню подготовки выпускников указанной специальности.

Характеристика профессиональной деятельности техника-программиста предусматривает подготовку студентов к деятельности по разработке, сопровождению и эксплуатации программного обеспечения компьютерных систем.

Методические указания выполнены с учетом положения «О выпускной квалификационной работе» СМК.3-ПТ-4.2.3-100-2014, методических указаний по расчету экономической части дипломного проекта (расчета стоимости разработки и внедрения программного продукта).

1 Общие положения

Дипломный проект выполняется на заключительном этапе обучения по специальности и характеризует уровень подготовки выпускника к самостоятельной работе.

Дипломный проект по содержанию может носить практический или опытно-экспериментальный характер, в котором решается актуальная задача для направления подготовки 230100 Информатика и вычислительная техника. Дипломный проект по содержанию должен соответствовать современному уровню развития информационных и телекоммуникационных технологий, аппаратных и программных средств вычислительной техники. Объем и степень сложности должны соответствовать теоретическим знаниям и практическим навыкам, полученным им в период обучения, а также в период прохождения учебной и производственной практики.

Процесс подготовки, выполнения и защиты дипломного проекта состоит из ряда последовательных этапов:

- назначение руководителя;
- выбор темы;
- выдача задания на дипломный проект;
- анализ литературы и интернет-источников по выбранной тематике;
- систематизация и обобщение материала как результат работы над источниками, проведение исследований, написание программного кода, анализ полученных данных;
- оформление текста пояснительной записки в соответствии с требованиями, предъявляемыми к дипломным проектам, и сдача его руководителю на проверку;
- доработка текста по замечаниям руководителя;
- письменный отзыв руководителя;
- представление пояснительной записки на подпись консультанту по экономической части;
- представление пояснительной записки на подпись нормоконтролеру;
- рецензирование дипломного проекта;
- представление завершенной работы на подпись заместителю директора по УР;
- подготовка к защите (разработка тезисов доклада для защиты, изучение отзыва руководителя и замечаний рецензента, создание презентации), предзащита;
- защита дипломного проекта на заседании государственной экзаменационной комиссии.

1.1 Цели и задачи дипломного проектирования

Целью дипломного проектирования является закрепление и расширение теоретических и практических знаний студента. Выпускник должен показать способность и умение применять теоретические положения изучаемых в техникуме дисциплин, профессиональных модулей и передовые достижения науки и техники; грамотно, самостоятельно и творчески решать поставленные задачи; четко и логично излагать свои мысли и решения; анализировать полученные результаты и делать необходимые выводы.

Задачей дипломного проектирования, состоящего из двух основных этапов: преддипломной практики и выполнения дипломного проекта, является самостоятельное выполнение студентом теоретической и практической частей дипломного проекта, характерных для техника-программиста. Студент при этом должен показать свой уровень подготовки, умение выбрать и обосновать решение стоящих перед ним проблем, навыки работы с технической и справочной литературой, умение применять вычислительную технику в своей деятельности.

Дипломный проект является выпускной квалификационной работой студента, на основании которой государственная экзаменационная комиссия оценивает качество подготовки студента и решает вопрос о присвоении ему квалификации техника-программиста.

Дипломный проект оформляется в виде пояснительной записки.

1.2 Выбор темы дипломного проекта

Темы дипломных проектов подбираются руководителем дипломного проекта. Студенту предоставляется право выбора темы выпускной квалификационной работы, в том числе предложения своей тематики с необходимым обоснованием целесообразности ее разработки для практического применения. При этом тематика выпускной квалификационной работы должна соответствовать содержанию одного или нескольких профессиональных модулей, входящих в программу подготовки специалистов среднего звена. Наименование дипломного проекта должно быть лаконичным и точно отражать суть проекта. Выбранные темы рассматриваются на заседании выпускающей цикловой комиссии по специальности.

Закрепление за студентами тем выпускных квалификационных работ, назначение руководителей и консультантов осуществляется приказом директора техникума.

После утверждения тема дипломного проекта может быть изменена только дополнительным приказом директора, что допускается лишь в исключительных случаях.

В дипломном проекте разрабатывается программный продукт или база данных.

Тематические направления дипломных проектов:

1. Разработка Web-сайтов.
2. Разработка тестирующих систем.
3. Разработка баз данных для конкретной системы управления базами данных.
4. Разработка мобильных приложений.
5. Создание учебно-методических материалов.
6. Разработка информационных систем.

2 Структура дипломного проекта

Готовый дипломный проект должен содержать:

- пояснительную записку (далее – ПЗ);
- техническое задание и/или бланк-заказ на разработку программного продукта (приложение А);
- бланк о внедрении программного продукта (приложение Б);
- отзыв руководителя на дипломный проект;
- рецензию на дипломный проект;
- диск с записанными на нем материалами: ПЗ, программным продуктом;
- демонстрационный материал для выступления.

2.1 Задание на дипломное проектирование

Задание на дипломное проектирование разрабатывается и оформляется руководителем дипломного проекта на специальных бланках (приложение В).

Задание подписывается руководителем дипломного проекта и утверждается заместителем директора по учебной работе.

Задание на дипломное проектирование содержит перечень вопросов, подлежащих разработке.

2.2 Содержание пояснительной записки

Пояснительная записка к дипломному проекту должна содержать (в приведенной последовательности):

- титульный лист (приложение Г)
- задание на ДП
- содержание (приложение Д)
- обозначения и сокращения
- введение
- анализ требований к проекту:
 - обзор аналогов программного обеспечения;
 - анализ инструментов используемых в разработке программного обеспечения;
 - техническое задание на разработку программного обеспечения.
- проектирование:

- архитектура программного обеспечения;
- функциональное проектирование;
- проектирование базы данных;
- проектирование пользовательского интерфейса программного обеспечения.
- реализацию программного обеспечения
 - кодирование программного обеспечения
- тестирование и отладку программного обеспечения
- внедрение и сопровождение программного обеспечения
- документирование программного обеспечения
 - руководство по установке программного обеспечения
 - руководство пользователя программного обеспечения
 - руководство системного администратора программного обеспечения
- стоимость разработки и внедрения программного продукта
- заключение
- список используемых источников
- приложения (при необходимости)

Содержание

Содержание включает введение, наименования всех разделов, подразделов и пунктов, заключение, список использованных источников и наименования приложений с указанием номеров страниц, с которых начинаются эти элементы пояснительной записки.

Введение

Во введении указывается актуальность темы, цель, объект и предмет исследования, гипотеза (при необходимости), задачи работы, методы исследования (при необходимости), практическая значимость, база исследования (при необходимости).

Особое внимание рекомендуется уделить актуальности выбранной темы.

Анализ требований к проекту

Анализ требований – часть процесса разработки программного обеспечения, включающая в себя сбор требований к программному обеспечению (ПО), их систематизацию, выявление взаимосвязей, а также документирование.

В процессе сбора требований важно принимать во внимание возможные противоречия требований различных заинтересованных лиц, таких как заказчики, разработчики или пользователи.

Полнота и качество анализа требований играют ключевую роль в успехе всего проекта. Требования к ПО должны быть документируемые, выполнимые, тестируемые, с

уровнем детализации, достаточным для проектирования системы. Требования могут быть функциональными и нефункциональными.

Анализ требований включает три типа деятельности:

Сбор требований – общение с клиентами и пользователями, чтобы определить, каковы их требования; анализ предметной области.

Анализ требований – определение, являются ли собранные требования неясными, неполными, неоднозначными или противоречащими; решение этих проблем; выявление взаимосвязи требований.

Документирование требований – требования могут быть задокументированы в различных формах, таких как простое описание, сценарии использования, пользовательские истории, или спецификации процессов.

Результатом этого этапа является законченное техническое задание к продукту. Оно должно содержать полное описание поведения будущего программного продукта и не содержать неоднозначностей и вопросов.

Обзор аналогов программного обеспечения

Аналоги выбираются и анализируются в зависимости от темы. Они обязательно должны быть поделены на группы/типы, как минимум на прямые и косвенные. Под прямыми аналогами в первую очередь понимаются проектные решения, выполненные по аналогичным темам. Под косвенными аналогами могут пониматься проектные решения из других тем и областей, которые можно применить в условиях конкретного дипломного проектирования. К ним могут относиться различные технологии, приёмы и решения из любого другого вида программного продукта или профессиональной деятельности. По выделенным группам аналогов должен быть приведён краткий анализ (например, пользовательский интерфейс, стиль представления информации, содержание, архитектура построения и т.п.). На основании этого анализа делается вывод о том, какими концептуальными, визуальными и иными признаками должен обладать будущий проект.

Анализ инструментов используемых в разработке программного обеспечения

Анализ инструментов используемых в разработки программного обеспечения – часть процесса работы над ВКР, включающая в себя сбор и анализ информации по использованию инструментальных средств в процессе разработки на каждом этапе жизненного цикла программного продукта. Данный пункт, может представлен перечнем используемых инструментов в процессе разработки, их анализом с выявлением основных критериев выбора их обоснованием. Результат можно представлять в виде сравнительной информации с выделением основных критериев и конкретных выводов. В данном разделе

студент должен продемонстрировать способность делать самостоятельный обоснованный выбор в анализе и использовании инструментов для разработки.

В данном разделе делается обоснованный выбор средств и технологий, которые предполагается использовать для решения поставленных задач. Например, осуществляется выбор тестирующих программ с указанием их преимуществ и уникальных свойств. При сравнительном тестировании программ создается перечень ключевых характеристик, по которым предполагается производить сравнение. При разработке в области сетевых технологий, например, можно привести основные характеристики необходимых сетевых устройств или приложений

В данном разделе студент должен продемонстрировать способность делать самостоятельный обоснованный выбор и защищать свое решение.

Техническое задание на разработку программного обеспечения

Техническое задание (ТЗ) – это основной документ в рамках проектной документации. Именно в ТЗ описываются все основные требования на разработку программного обеспечения, будь то создание либо простенькой программы или сайта, либо же разработка крупномасштабной информационной системы или программно-аппаратного комплекса. Техническое задание может разрабатываться как в рамках эскизного проекта (это когда только описание функций и структуры системы без рассмотрения технологий реализации решения), так и в технический проект (более детальное описание с учетом выбранных технологий).

Техническое задание может быть как поверхностным (например, общеконцептуальное ТЗ, предназначенное для инвесторов проекта), так и более детальным (например, подробное ТЗ для программиста).

В некоторых случаях можно обойтись только подготовкой одного технического задания для описания разрабатываемой системы.

Стандарты для технического задания

Существует несколько ГОСТов, регламентирующих разработку ТЗ: это ГОСТ 34.602 (автоматизированные системы) и ГОСТ 19.201 (программное обеспечение). Документы, выполненные по этим стандартам, значительно отличаются как по наполнению, так и по содержанию.

Состав типового технического задания

Техническое задание, вне зависимости от выбранного ГОСТа, всегда включает следующие основные сведения по разрабатываемому ПО:

- 1) *наименование* – полное и краткое названия, условное обозначение разрабатываемого ПО;

- 2) *назначение* – то, для чего, в какой области и с какой целью разрабатывается ПО;
- 3) *основание для разработки* – документы, на основании которых производится разработка ПО;
- 4) *функции* – перечень и описание функций разрабатываемого ПО;
- 5) *структура* – описание архитектуры и компонентов разрабатываемого ПО;
- 6) *пользовательский интерфейс* – в современном мире обязателен;
- 7) *надежность, безопасность, условия эксплуатации* и проч. важные требования;
- 8) *документация* – какая документация, в каком объеме и в соответствии с какими требованиями ГОСТов будет также разработана;
- 9) *стадии и этапы разработки* – что и в какой последовательности разрабатывается;
- 10) *порядок контроля и приемка* – как именно будет происходить сдача разработанного ПО Заказчику.

Техническое задание – один из основных документов представляемых на защите ВКР. Этот документ не вшивается в пояснительную записку, а прикладывается к ВКР.

Проектирование

На основе предыдущего этапа проводится проектирование системы (программного продукта). Методология проектирования соединяет в себе объектную декомпозицию, приемы представления физической, логической, а также динамической и статической моделей системы.

Во время проектирования разрабатываются проектные решения по выбору платформы, где будет функционировать система языка или языков реализации, назначаются требования к пользовательскому интерфейсу, определяется наиболее подходящая СУБД. Разрабатывается функциональная спецификация ПО: выбирается архитектура системы, оговариваются требования к аппаратному обеспечению, определяется набор орг. мероприятий, которые необходимы для внедрения ПО, а также перечень документов, регламентирующих его использование.

На этапе проектирования используют CASE-технологии. Для выполнения поставленной цели CASE-технологии используют два принципиально разных подхода к проектированию: структурный и объектно-ориентированный.

Структурный подход предполагает декомпозицию (разделение) поставленной задачи на функции, которые необходимо автоматизировать. В свою очередь, функции также разбиваются на подфункции, задачи, процедуры. В результате получается упорядоченная иерархия функций и передаваемой информацией между функциями.

Структурный подход подразумевает использование определенных общепринятых методологий при моделировании различных информационных систем:

SADT (structured analysis and design technique);

DFD (data flow diagrams);

ERD (entity-relationship diagrams).

Существует три основных типа моделей, используемых при структурном подходе: функциональные, информационные и структурные.

Основным инструментом объектно-ориентированного подхода является язык UML – унифицированный язык моделирования, который предназначен для визуализации и документирования объектно-ориентированных систем с ориентацией их на разработку программного обеспечения. Данный язык включает в себя систему различных диаграмм, на основании которых может быть построено представление о проектируемой системе.

Архитектура программного обеспечения

Процесс проектирования архитектуры программного обеспечения включает в себя сбор требований клиентов, их анализ и создание проекта для компонента программного обеспечения в соответствии с требованиями.

Архитектура программного обеспечения включает в себя определение структурированного решения, соответствующего всем техническим и рабочим требованиям, одновременно оптимизируя общие атрибуты качества, такие как производительность, безопасность и управляемость.

Современное программное обеспечение редко бывает автономным. Как минимум, в большинстве случаев оно будет взаимодействовать с источником данных, например корпоративной базой данных, предоставляющим информацию, с которой работают пользователи программного обеспечения. Обычно современное программное обеспечение также должно взаимодействовать с другими службами и сетевыми функциями для выполнения проверки подлинности, получения и публикации информации и предоставления интегрированных сред работы пользователей. Без соответствующей архитектуры может быть сложно, если вообще возможно, осуществить развертывание, эксплуатацию, обслуживание и успешную интеграцию с другими системами; кроме того, требования пользователей не будут соблюдены.

Результат этого этапа представляется в виде схем, моделей их описанием и обоснованием выбранного решения.

Функциональное проектирование

Прежде чем проектировать структуры данных, разработчик должен построить функциональную модель проектируемой системы.

Любой объект служит лишь материальным носителем функции, то есть функция — первична, объект — вторичен и создается по причине невозможности иными,

нематериальными средствами удовлетворить потребности людей. Так, автомобиль нужен для перевозки грузов и людей (функция — перемещать в пространстве, создан вследствие нереальности перемещения предметов только усилием мысли), назначение ручки — писать, а книги — хранить информацию и т.д.

Наряду со словом «функция» часто используется слово «назначение», особенно при рассмотрении не технических объектов.

Функциональное проектирование нацелено, прежде всего, на создание эффективно работающего объекта. Выполнение требуемой функции — главная цель и основа разработки объекта. Во внимание принимаются, прежде всего, функциональные показатели качества и показатели надёжности.

Функциональная модель проектируемой системы напрямую вытекает из функциональных требований, которые, в свою очередь, могут быть описаны в виде прецедентов.

Прецедент, также: вариант использования, сценарий использования — спецификация последовательностей действий (варианты последовательностей и ошибочные последовательности) в Унифицированном языке моделирования (UML), которые может осуществлять система, подсистема или класс, взаимодействуя с внешними действующими лицами.

Результат этого этапа представляется в виде диаграммы и ее описания на языке UML.

Проектирование базы данных

Проектирование баз данных – процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи проектирования баз данных:

- Обеспечение хранения в БД всей необходимой информации.
- Обеспечение возможности получения данных по всем необходимым запросам.
- Сокращение избыточности и дублирования данных.
- Обеспечение целостности базы данных.

Основные этапы проектирования баз данных: концептуальное (инфологическое) проектирование, логическое (дatalogическое) проектирование, физическое проектирование.

Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо

конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные *ER-диаграммам*.

Логическое (даталогическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

При проектировании реляционных баз данных обычно выполняется так называемая нормализация.

На данном этапе в работе над ВКР студент строит модель «сущность-связь».

Модель «сущность-связь», или ER-модель - моделей предметной области. ER-модель обычно представляется в графической форме, с использованием оригинальной нотации ER-диаграмма. Основные преимущества ER-моделей:

- наглядность;
- модели позволяют проектировать базы данных с большим количеством объектов и атрибутов;

- ER-модели реализованы во многих системах автоматизированного проектирования баз данных (например, ERWin).

Проектирование пользовательского интерфейса программного обеспечения

Разработка пользовательского интерфейса является неотъемлемой частью любого проекта связанного с созданием программного обеспечения. Интерфейс пользователя является точкой взаимодействия человека и программы, зачастую имеющей сложную функциональность. От того насколько удобным будет разработанный интерфейс пользователя будет зависеть и успех продукта.

Задачи решаемые разработкой пользовательского интерфейса:

- повышение эффективности работы пользователя, заключающейся в повышении скорости и простоты выполнения операций;
- увеличение удовлетворенности работы пользователя, связанной с сокращением времени на обучение, уменьшением частоты использования справочной документации и обращением в службу поддержки, уменьшением общего числа пользовательских ошибок.

Этапы разработки пользовательского интерфейса

- Сбор и анализ требований бизнес-заказчиков.
- Разработка сценариев работы пользователей.
- Выбор показателей оценки эффективности пользовательского интерфейса.
- Прототипирование и разработка макетов.
- Разработка средств поддержки пользователей.
- Юзабилити тестирование и обеспечение качества интерфейса.
- Разработка документации.

Построение пользовательского интерфейса основывается на принципах построения пользовательских интерфейсов.

Реализация программного обеспечения

Данный этап разработки программного обеспечения организован в соответствии с моделями эволюционного типа жизненного цикла ПО. При разработке применяются экспериментирование и анализ, строятся прототипы, как целой системы, так и ее частей. Прототипы дают возможность глубже вникнуть в проблему и принять все необходимые проектные решения еще на ранних этапах проектирования. Такие решения могут затрагивать разные части системы: внутреннюю организацию, пользовательский интерфейс, разграничение доступа и т.д.

В результате этапа реализации появляется рабочая версия продукта.

Кодирование программного обеспечения

Кодирование – процесс написания программного кода, скриптов с целью реализации определённого алгоритма на определённом языке программирования.

Кодирование может происходить параллельно со следующим этапом разработки — тестированием программного обеспечения, что помогает вносить изменения непосредственно по ходу написания кода. Уровень и эффективность взаимодействия всех элементов, задействованных для выполнения сформулированных задач компанией-разработчиком, на текущем этапе является самым важным — от слаженности действий программистов, тестировщиков и проектировщиков зависит качество реализации проекта.

Тестирование и отладка программного обеспечения

Тестирование тесно связано с такими этапами разработки программного обеспечения как проектирование и реализация. В систему встраиваются специальные механизмы, которые дают возможность производить тестирование системы на соответствие требований к ней, проверку оформления и наличие необходимого пакета документации.

Результатом тестирования является устранение всех недостатков системы и заключение о ее качестве.

Существует несколько признаков по которым принято производить классификацию видов тестирования. Обычно выделяют следующие признаки:

По объекту тестирования:

- Функциональное тестирование (functional testing)
- Нагрузочное тестирование
- Тестирование производительности (performance/stress testing)
- Тестирование стабильности (stability/load testing)
- Тестирование удобства использования (usability testing)
- Тестирование интерфейса пользователя (UI testing)
- Тестирование безопасности (security testing)
- Тестирование локализации (localization testing)
- Тестирование совместимости (compatibility testing)

По знанию системы:

- Тестирование чёрного ящика (black box)
- Тестирование белого ящика (white box)
- Тестирование серого ящика (gray box)
- По степени автоматизированности:
- Ручное тестирование (manual testing)

- Автоматизированное тестирование (automated testing)
- Полуавтоматизированное тестирование (semiautomated testing)

По степени изолированности компонентов:

- Компонентное (модульное) тестирование (component/unit testing)
- Интеграционное тестирование (integration testing)
- Системное тестирование (system/end-to-end testing)

По времени проведения тестирования:

- Альфа тестирование (alpha testing)
- Тестирование при приёмке (smoke testing)
- Тестирование новых функциональностей (new feature testing)
- Регрессионное тестирование (regression testing)
- Тестирование при сдаче (acceptance testing)
- Бета тестирование (beta testing)

По признаку позитивности сценариев:

- Позитивное тестирование (positive testing)
- Негативное тестирование (negative testing)

По степени подготовленности к тестированию:

- Тестирование по документации (formal testing)
- Эд Хок (интуитивное) тестирование (ad hoc testing)

Уровни тестирования

Модульное тестирование (юнит-тестирование) – тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция. Часто модульное тестирование осуществляется разработчиками ПО.

Интеграционное тестирование – тестируются интерфейсы между компонентами, подсистемами. При наличии резерва времени на данной стадии тестирование ведётся итерационно, с постепенным подключением последующих подсистем.

Системное тестирование – тестируется интегрированная система на её соответствие требованиям.

Альфа-тестирование – имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования. Иногда альфа-тестирование выполняется под отладчиком или с использованием окружения, которое помогает быстро

выявлять найденные ошибки. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования в окружении, подобном тому, в котором будет использоваться ПО.

Бета-тестирование – в некоторых случаях выполняется распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц, с тем чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей.

Часто для свободного/открытого ПО стадия Альфа-тестирования характеризует функциональное наполнение кода, а Бета тестирования – стадию исправления ошибок. При этом как правило на каждом этапе разработки промежуточные результаты работы доступны конечным пользователям.

В данном разделе указывается как проходило тестирование программного продукта, с помощью какого метода проводилось тестирование.

Внедрение и сопровождение программного обеспечения

Внедрение системы обычно предусматривает следующие шаги:

- установка системы,
- обучение пользователей,
- эксплуатация.

Внедрение программного обеспечения – процесс настройки программного обеспечения под определённые условия использования, а также обучения пользователей работе с программным продуктом.

Сопровождение программного обеспечения – это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию.

Данный этап и раздел ВКР можно подтверждать документом – бланком о внедрении программного продукта, который подтверждает практическую значимость программного продукта.

Документирование программного обеспечения

Документация на программное обеспечение – печатные руководства пользователя, диалоговая (оперативная) документация и справочный текст, описывающие, как пользоваться программным продуктом.

Документ – элемент документации: целевая информация, предназначенная для конкретной аудитории, размещенная на конкретном носителе (например, в книге, на диске, в краткой справочной карте) в заданном формате.

Программный документ – документ, содержащий в зависимости от назначения данные, необходимые для разработки, производства, эксплуатации, сопровождения программы или программного средства.

Существует основные типы документации на ПО:

- архитектурная/проектная — обзор программного обеспечения, включающий описание рабочей среды и принципов, которые должны быть использованы при создании ПО;
- техническая – документация на код, алгоритмы, интерфейсы, API;
- пользовательская – руководства для конечных пользователей, администраторов системы и другого персонала.

Проектная (архитектурная) документация обычно описывает продукт в общих чертах. Не описывая того, как что-либо будет использоваться, она скорее отвечает на вопрос «почему именно так». Например, в проектом документе программист может описать обоснование того, почему структуры данных организованы именно таким образом. Описываются причины, почему какой-либо класс сконструирован определённым образом, выделяются паттерны, в некоторых случаях даже даются идеи как можно будет выполнить улучшения в дальнейшем. Ничего из этого не входит в техническую или пользовательскую документацию, но всё это действительно важно для проекта.

Техническая документация. При создании программы, одного лишь кода, как правило, недостаточно. Должен быть предоставлен некоторый текст, описывающий различные аспекты того, что именно делает код. Такая документация часто включается непосредственно в исходный код или предоставляется вместе с ним.

Подобная документация имеет сильно выраженный технический характер и в основном используется для определения и описания API, структур данных и алгоритмов.

Часто при составлении технической документации используются автоматизированные средства – генераторы документации, такие как Doxygen, javadoc, NDoc и другие. Они получают информацию из специальным образом оформленных комментариев в исходном коде, и создают справочные руководства в каком-либо формате, например, в виде текста или HTML.

Использование генераторов документации и документирующих комментариев многими программистами признаётся удобным средством, по различным причинам. В частности, при таком подходе документация является частью исходного кода, и одни и те же инструменты могут использоваться для сборки программы и одновременной сборки документации к ней. Это также упрощает поддержку документации в актуальном состоянии.

Пользовательская документация.

В отличие от технической документации, сфокусированной на коде и том, как он работает, пользовательская документация описывает лишь то, как использовать программу.

В случае если продуктом является программная библиотека, пользовательская документация и документация на код становятся очень близкими, почти эквивалентными понятиями. Но в общем случае, это не так.

Обычно, пользовательская документация представляет собой *руководство пользователя*, которое описывает каждую функцию программы, а также шаги, которые нужно выполнить для использования этой функции. Хорошая пользовательская документация идёт ещё дальше и предоставляет инструкции о том, что делать в случае возникновения проблем. Очень важно, чтобы документация не вводила в заблуждение и была актуальной. Руководство должно иметь чёткую структуру; очень полезно, если имеется сквозной предметный указатель. Логическая связность и простота также имеют большое значение.

Существует три подхода к организации пользовательской документации. Вводное руководство (англ. tutorial), наиболее полезное для новых пользователей, последовательно проводит по ряду шагов, служащих для выполнения каких-либо типичных задач. Тематический подход, при котором каждая глава руководства посвящена какой-то отдельной теме, больше подходит для совершенствующихся пользователей. В последнем, третьем подходе, команды или задачи организованы в виде алфавитного справочника – часто это хорошо воспринимается продвинутыми пользователями, хорошо знающими, что они ищут. Жалобы пользователей обычно относятся к тому, что документация охватывает только один из этих подходов, и поэтому хорошо подходит лишь для одного класса пользователей.

Во многих случаях разработчики программного продукта ограничивают набор пользовательской документации лишь встроенной системой помощи, содержащей справочную информацию о командах или пунктах меню. Работа по обучению новых пользователей и поддержке совершенствующихся пользователей перекладывается на частных издателей, часто оказывающих значительную помощь разработчикам.

Руководство по установке программного обеспечения

Установка программного обеспечения, инсталляция – процесс установки программного обеспечения на компьютер конечного пользователя. Руководство по установке программного обеспечения – один из документов сопровождающий

программный продукт. В нем указывается шаги по процессу установки программного продукта.

Руководство системного администратора программного обеспечения

Если программа более-менее проста, пользователь может установить ее себе на компьютер самостоятельно. Поддерживать ее работоспособность, например, выполнить переустановку, если возникнут какие-нибудь проблемы, он тоже, как правило, в состоянии.

Сложными, в том числе, серверными и сетевыми программными продуктами приходится заниматься квалифицированным специалистам, системным администраторам. Более того, во многих компаниях сотрудникам запрещено устанавливать на своих рабочих местах программы по своему усмотрению. Даже простые программы там ставит только системный администратор.

В обязанности системного администратора также входит поддержание работоспособности программ, используемых в рамках тех или иных систем. Эта деятельность может заключаться в периодической проверке логов, резервном копировании данных, замерах производительности, устранении различных технических проблем.

Руководство системного администратора – программный документ, предоставляющий специалисту информацию, необходимую для выполнения этой работы.

Стоимость разработки и внедрения программного продукта

Основной целью экономической части является обоснование экономической целесообразности разработки данного проекта. Задачи включают технико-экономическое обоснование разработки студента, проведение анализа уже существующих аналогичных разработок, определение экономического эффекта от ее использования. Методика расчета экономической части представлена в Методических указаниях по расчету экономической части дипломного проекта специальностей: 09.02.01 Компьютерные системы и комплексы, 09.02.03 Программирование в компьютерных системах, 24.02.01 Производство летательных аппаратов, 15.02.08 Технология машиностроения.

Заключение

В данном разделе подводятся итоги проведенных исследований соответственно задачам, обозначенным во введении. Дается оценка проделанной работе и рекомендации по возможным путям дальнейшего развития исследований в данном направлении с учетом перспектив развития информационных технологий.

Список использованных источников

Список использованных источников оформляется в полном соответствии с библиографическими обоснованиями; составляется согласно ГОСТ 7.32-2001.

Приложения

В приложения помещаются исследовательские материалы, чертежи, таблицы и графики, авторские, методические разработки, рисунки, схемы, листинги программ, образцы документации, спецификации и т.д., например:

- Приложение А Техническое задание
- Приложение Б Код клиентской части
- Приложение С Код серверной части

2.3 Оформление пояснительной записки

Оформление пояснительной записки осуществляется в соответствии с п. 4 Требования к оформлению выпускной квалификационной работы согласно ГОСТ 7.32-2001 Положения о выпускной квалификационной работе СМК.3-ПТ-4.2.3-100-2014.

Приложение А
Пример оформления бланк-заказа на разработку программного обеспечения

Министерство образования Иркутской области
Государственное бюджетное профессиональное
образовательное учреждение Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

Студенту гр. ПКС-7 специальности
09.02.03 Программирование в
компьютерных системах
Чернигову И.С.

БЛАНК-ЗАКАЗ
на разработку программного продукта

Прошу Вас разработать программный продукт.

Название программного продукта: Информационная система тестирования.

Цель программного продукта: автоматизация процесса проверки знаний студентов по профессиональным дисциплинам и создания, хранения и обработки тестовых наборов и данных пользователей.

Место использования программного продукта: лаборатория 228 ГБПОУИО «ИАТ».

Предполагаемые сроки: начало работы _____
сдача работы _____

Требования к информационной системе:

1. Использование многопользовательского режима, с одновременным подключением не менее 15 пользователей.
2. Экспортирование/импортирование в офисное программное обеспечение.
 - а. экспортирование данных из таблиц базы данных в форматы HTML или XLSX.
 - б. Импортирование данных из файла Excel в одну из таблиц базы данных.
3. Удобный и интуитивно понятный интерфейс.
4. Внесение тестовых наборов в информационную систему преподавателем.
5. Интегрирование «Информационной системы тестирования» в информационную систему ГБПОУИО «ИАТ», в части связи аутентификации с доменными параметрами ЛВС техникума (путём сравнения входных данных авторизации с записями в директориях сервера техникума).
6. Наличие руководств пользователя и системного программиста.
7. Создание отчета процесса тестирования с отображением личных пройденных тестов для студентов и преподавателя. Просмотр вопросов, в которых были допущены ошибки при тестировании.
8. Генерация порядка вопросов случайным образом для каждого нового прохождения теста пользователем.
9. Возможность редактировать таблицы базы данных из программы для преподавателя.

Заказчик: председатель ВЦК ПКС ГБПОУИО «ИАТ» М.А. Богачева

Исполнитель разработки
Заказчик

И. С. Чернигов
М.А. Богачева

Приложение Б
Пример оформления бланка о внедрении программного продукта

Министерство образования Иркутской области
Государственное бюджетное профессиональное
образовательное учреждение Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

Блан о внедрении
Информационной системы тестирования

Настоящим документом подтверждаю, что программный продукт «Информационная система тестирования», разработанный Черниговым И.С. в качестве дипломного проекта внедрен в информационную систему лаборатории 228 ГБПОУИО «ИАТ».

Программный продукт соответствует всем требованиям, предъявленным к нему в бланк-заказе, поэтому он используется в учебном процессе для проверки знаний студентов, а также для создания, хранения и обработки тестовых наборов и данных пользователей.

Председатель ВЦК ПКС ГБПОУИО «ИАТ»

_____ М.А. Богачева

Приложение В

Образец оформления задания на дипломное проектирование

Министерство образования Иркутской области
Государственное бюджетное профессиональное образовательное учреждение
Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

УТВЕРЖДАЮ

Зам. директора по УР, к.т.н.

_____ Е.А. Коробкова

« » 2017 г.

ЗАДАНИЕ

на дипломное проектирование

Студенту Зачиняевой Юлии Андреевне, группы ПКС-7

Тема дипломного проекта: Игровое приложение «Чарлэнд»

Дата выдачи задания 04 апреля 2017 г.

Срок окончания проекта 01 июня 2017 г.

Руководитель: _____ (М.А. Богачева)
(подпись, дата)

Студент: _____ (Ю.А. Зачиняева)
(по подписи, дата)

Продолжение приложения В
Образец оформления задания на дипломное проектирование

1. **Исходные данные для проектирования**
Бланк-заказ на разработку программного обеспечения
Техническое задание.
2. **Анализ требований к проекту**
Обзор аналогов программного обеспечения
Анализ инструментов используемых в разработки программного обеспечения
3. **Проектирование программного обеспечения**
Архитектура программного обеспечения
Функциональное проектирование
Проектирование базы данных
Проектирование пользовательского интерфейса программного обеспечения
4. **Реализация программного обеспечения**
Кодирование программного обеспечения
5. **Тестирование и отладка программного обеспечения**
6. **Внедрение и сопровождение программного обеспечения**
7. **Документирование программного обеспечения**
Руководство по установке программного обеспечения
Руководство пользователя программного обеспечения
Руководство системного администратора программного обеспечения
8. **Стоимость разработки и внедрения программного продукта**
Организационно-экономическое обоснование проекта.
Расчет затрат на разработку программного продукта.
Расчет затрат на внедрение программного продукта.
Расчет основных показателей экономической эффективности.
9. **Материалы предоставляемые к защите**
Техническое задание на разработку.
Бланк-заказ.
Бланк о внедрении.
Пояснительная записка.
Презентация PowerPoint.
Диск с записанными на нем материалами, ПЗ, программой.
Отзыв на дипломный проект.
Рецензия на дипломный проект.

Приложение Г
Образец оформления титульного листа

Министерство образования Иркутской области
Государственное бюджетное профессиональное
образовательное учреждение Иркутской области
«Иркутский авиационный техникум»
(ГБПОУИО «ИАТ»)

ДП.09.02.03.16.07.22.ПЗ

УТВЕРЖДАЮ

Зам. директора по УР, к.т.н.

_____ Е.А. Коробкова

ПРИЛОЖЕНИЕ ДЛЯ ИНТЕРАКТИВНЫХ
ГЕОЛОКАЦИОННЫХ КВЕСТОВ

Зав. отделением:	_____	(С.Н. Быкова)
	(по подписи, дата)	
Консультант по экономической части:	_____	(А.А. Белова)
	(по подписи, дата)	
Нормоконтролер:	_____	(В.В. Пашкевич)
	(по подписи, дата)	
Руководитель:	_____	(М.А. Богачева)
	(по подписи, дата)	
Студент:	_____	(Л.А. Чертовских)
	(по подписи, дата)	

Иркутск 2016

Приложение Д
Пример оформления содержания

Содержание				
Обозначения и сокращения.....	3			
Введение.....	4			
1 Анализ требований к проекту.....	8			
1.1 Обзор существующих программных средств для создания интерактивных геолокационных квестов.....	8			
1.2 Обоснование выбора программных средств для создания интерактивного геолокационного квеста «GPS квест».....	21			
2 Проектирование.....	26			
2.1 Постановка задачи.....	26			
2.2 Ход выполнения работы.....	28			
2.3 Разработка приложения «GPS квест».....	28			
3 Внедрение и сопровождение	37			
3.1 Руководство по установке приложения «GPS квест».....	37			
3.1 Руководство пользователя.....	39			
4 Стоимость разработки и внедрения программного продукта.....	45			
Заключение.....	50			
Список используемых источников.....	51			
Приложение А ER-модель базы данных.....	52			
Приложение Б Код клиентского приложения.....	53			
Приложение В Код серверной части.....	83			

					ДП.09.02.03.16.07.22 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	ПРИЛОЖЕНИЕ ДЛЯ ИНТЕРАКТИВНЫХ ГЕОЛОКАЦИОННЫХ КВЕСТОВ пояснительная записка	Лист	Лист	Листов
Разраб.		Чертовских Л.А.						
Провер.		Богачева М.А.					3	83
Реценз.						ГБПОУИО «ИАТ» ПКС-7		
Н. Контр.		Пашкевич В.В.						
Утверд.								