**1 Обзор средств автоматизированного создания интернет-магазинов**

В наши дни любая организация сталкивается с необходимостью реализовать свою продукцию, будь то товары или услуги, с дальнейшей целью получения прибыли, причём неважно, занимается ли данная организация крупным производством или же просто продаёт то, что закупает у поставщиков. Современную торговую деятельность трудно представить без применения интернет-магазинов, ведь благодаря им поддерживается стабильность торговли, рекламы, сокращаются используемые складские площади, обеспечивается клиентоориентированность и доступность всем и каждому. Однако, стоит отметить, что по большей части торговлей через интернет занимаются именно розничные структуры, крупные производственные компании занимаются этим нечасто.

В данном разделе будет описана предметная область и представлены результаты формализации требований по категориям пользователей, требуемым функциям, используемым данным и заявленным ограничениям.

**1.1 Анализ предметной области. Обзор литературы по теме работы**

Электронная коммерция (или E-commerce) уже давно стала частью практически любой сферы бизнеса. Конечно, до сих пор существуют и такие компании, которые наотрез отвергают «онлайн», которые продолжают вести дела только на физическом реальном рынке. И они вполне имеют право на существование. Однако, медленно, но верно бизнес перестаёт даже рассматривать Интернет отдельно от оффлайновой составляющей, поскольку интернет в наши дни – это часть маркетинговой стратегии, один из каналов коммуникации с имеющимися и потенциальными клиентами [1]. А помочь компании вступить в E-commerce безусловно сможет интернет-магазин.

Что же такое интернет-магазин? Согласно свободной энциклопедии Википедии, интернет-магазин – это сайт, торгующий товарами посредством сети Интернет [2]. То есть, подобно тому, как происходит аренда торгового помещения в реальности, коммерсант «арендует» некоторое пространство в интернете, размещает там веб-сайт со своими товарами и/или услугами и торгует. Такое пространство называется веб-хостинг – услуга по предоставлению ресурсов для размещения некоторой информации на сервере провайдера [3]. Аренда виртуального пространства несравнимо дешевле, чем реального. К примеру, месячная оплата хостинга в России может начинаться примерно от 100 рублей [4]. За такие деньги не найти реальное помещение для ведения торговли. Отсюда первый плюс для организации – малые расходы на поддержание самой торговой площадки. Второй плюс очевиден из самого понятия «интернет-магазин» – данный магазин находится в сети Интернет, а значит, при наличии интернета доступен абсолютно всем и всегда по всему миру. Также, стоит отметить и то, что благодаря размещению товаров в интернет-магазине стираются границы и препятствия для малоподвижных граждан, инвалидов и пенсионеров, которые просто физически не смогли бы посетить настоящий магазин с целью купить что-то необходимое.

Выше описаны три очевидных и неотъемлемых достоинства интернет-магазина. Кроме них к достоинствам можно отнести те, которые станут таковыми только при усилиях самого продавца. Например, тщательно продуманная структура организации магазина добавит лояльности потребительской массы и клиентоориентированности. Данная структура включает в себя составление каталога товаров, возможность его сортировки и фильтрации, а также наполнение каждой товарной позиции исчерпывающей информацией, ознакомление с которой не оставит у покупателя никаких вопросов.

И если полной информацией обо всей своей продукции коммерсант точно владеет, то информацией о том, как же этот интернет-магазин спроектировать, сверстать, запрограммировать и заставить работать как надо, скорее всего, нет. Если же продавец в одном лице является также и веб-разработчиком, тогда, возможно, никаких проблем не возникнет. Но в обратном случае неизбежно встанет вопрос о том, что же делать.

Здесь возникают три наиболее распространенных решения [5]:

1. Рассчитывать только на свои силы, начать изучать языки программирования и верстки. Абсолютно бесплатно, но крайне долго, поскольку невозможно выучить язык программирования за неделю или месяц, а языков придется изучить несколько.

2. Заказать разработку интернет-магазина в специализированной организации у профессионалов. Данный вариант является быстрым, но дорогим. Цены на создание интернет-магазинов варьируются, но в среднем составляют около 50 тысяч рублей. Зато заказчику вообще не придется углубляться в технологии создания сайтов, а исполнители сделают всё, что потребуется.

3. Арендовать готовый интернет-магазин. Практически моментально, но так же дорого. Помесячная плата составит около тысячи рублей. Также, некоторые готовые площадки можно выкупить полностью. В этом случае вообще не нужно будет задумываться о том, как открыть интернет-магазин, ведь он уже открыт.

Анализ вышеприведенных вариантов действий потенциального электронного коммерсанта показывает, что в любом случае потребуются огромные затраты – либо временные, либо материальные.

Решить данную проблему поможет четвертый вариант, объединяющий быстроту и дешевизну открытия интернет-магазина – использование так называемых конструкторов или CMS.

CMS (Content management system) – система управления контентом, информационная система, обеспечивающая удобный процесс создания, редактирования и управления содержимым сайта, иногда даже без требований к знанию программирования [6]. Кроме того, помимо управления содержимым CMS предоставляет пользователю уже готовые к использованию функции бизнес-логики и всё внутреннее устройство сайта. Таким образом, у пользователя отпадает необходимость какой-либо работы с исходным кодом.

Как и другое ПО, конструктор может быть платным или бесплатным. Порой от этого зависит и набор функций, предоставляемых пользователю. Так или иначе, отсутствие требований к знаниям языков программирования и вёрстки сайтов вкупе с бесплатным распространением обязательно помогут с созданием интернет-магазина. Существует большое количество самых разных конструкторов сайтов, однако в данной работе речь идет исключительно о конструкторах интернет-магазинов.

В общем, конструктор позволяет человеку далекому от информационных технологий создать полноценный веб-сайт и в дальнейшем его администрировать, управлять содержимым (контентом) сайта.

Основные преимущества конструкторов сайтов [7]:

1. Простота в использовании. Всё выполняется пошагово и легко, практически полное отсутствие необходимости работать с кодом.

2. Разделение визуальной и текстовой составляющих. Простая смена дизайна.

3. Наличие готовых шаблонов сайтов. Отсюда главный недостаток конструкторов – привязанность только к имеющимся дизайнам. Для создания чего-либо собственного нового необходимы знания языков вёрстки и программирования.

В книге-бестселлере 2017 года «Ваш интернет-магазин от А до Я» автора Тимофея Шиколенкова [1] рассматривается комплексная организация собственного интернет-магазина. Рассмотрен каждый аспект создания и управления магазином. Помимо устройства самого интернет-магазина рассмотрены маркетинговые, экономические и логистические составляющие.

В книге «Как открыть интернет-магазин. И не закрыться через месяц» автора Александра Вереса [8], электронного коммерсанта с пятнадцатилетним стажем, рассматривается не только организация собственного интернет-магазина, но и все тонкости E-commerce. Что примечательно, в данной книге уделяется внимание даже тому, как выбрать нишу – самое главное, с чего надо начинать создание магазина, а также выбору названия для магазина.

В учебном пособии бизнес-школы Imsider 2020 «План открытия интернет-магазина» за авторством Тимура Шакова и Николая Федоткина [9] также рассмотрены вопросы, касающиеся как технических нюансов интернет-магазинов, так и экономических, маркетинговых и других. Кроме того, в данном источнике имеются некоторые сведения о продвижении магазина (SEO) и, собственно, о выборе движка (CMS) для конструирования сайта.

В настоящее время проблема выбора конструктора для интернет-магазинов является актуальной. Она заключается в том, что, во-первых, не так уж и много CMS, направленных на электронную коммерцию. Во-вторых, большинство существующих решений – платные, некоторые даже очень дорогие (например, InSales, 1C-UMI). Они, может, и правда несут в себе гигантский массив функций и пользы для продавца, но начинающий попросту не сможет оплатить их услуги. В-третьих, одни продукты предлагают слишком малую функциональность, а другие наоборот – настолько сложные, что необходимо тратить время на изучение того, как же работать с данной CMS.

Исходя из вышеперечисленных проблем, и была выбрана тема выпускной квалификационной работы, результатом выполнения которой должен стать простой в использовании, но функционально насыщенный конструктор интернет-магазинов.

**1.2 Определение задач проекта**

В ходе описания предметной области была выделена цель: разработать автоматизированную информационную систему управления содержимым (CMS), направленную на создание интернет-магазинов.

Кроме того, были сформулированы задачи проекта:

1. Определить требования системы, аппаратные и программные средства;
2. Выбрать среду разработки, языки программирования и вёрстки;
3. Выполнить проектирование системы;
4. Реализовать формирование внешнего отображения сайта;
5. Реализовать подсистему добавления товаров в каталог различными способами;
6. Реализовать подсистемы обратной связи и оповещений;
7. Реализовать подсистему формирования отчётностей по продажам;
8. Обеспечить хранение всей необходимой информации в базе данных;
9. Разработать графический пользовательский веб-интерфейс системы.

Предполагается, что разрабатываемой системой будут пользоваться люди, далекие от программирования, но желающие создать свой собственный интернет-магазин. Следовательно, основное требование состоит в том, что при работе с конструктором пользователь не должен столкнуться с необходимостью работать с кодом.

В целом, базовая работа с конструктором должна строиться из следующих шагов:

1. Выбор названия для будущего интернет-магазина.
2. Выбор дизайна из имеющегося списка.
3. Заполнение контактных данных.
4. Заполнение каталога товаров ассортиментом.
5. Заполнение информации об оплате и доставке.

На данном этапе базовый интернет-магазин будет готов. Все дальнейшие модификации и дополнения сайта (например, SEO, аналитика и т.п.) вполне возможны, но необязательны и вариативны.

В качестве используемых данных будут выступать данные по большей части о товарах продавца.

Система должна выполнять следующие функции:

1. Пошаговое создание интернет-магазина.
2. Дальнейшая поддержка сайта и его администрирование.
3. Возможность формирования отчётностей по проданным товарам.
4. Обеспечение обратной связи покупателя с продавцом.
5. Смена дизайна сайта в один клик.
6. Хранение в базе данных всей информации о товарах и продажах.

**1.3 Обзор аналогов**

На данный момент существует достаточное количество аналогов конструктора интернет-магазинов. Рассмотрим наиболее популярные из них [10]. Среди основных критериев сравнения можно выделить платную либо бесплатную лицензию, наличие ознакомительного периода, мультиязычность (в частности, наличие русского языка), интуитивность и простота в использовании.

**1.3.1 Shop-Script**

Shop-Script – облачный конструктор интернет-магазинов от Webasyst. Богат настройками и плагинами. Среди них есть платные (от 300 рублей) и бесплатные. Имеются такие маркетинговые функции, как интеграции социальных сетей, почтовые рассылки, всплывающие окна, баннеры, SMS-уведомления и т.п.

Бесплатных шаблонов всего лишь 9 вариантов. Стоимость платных от 2 тысяч рублей. Большинство важных преобразований нужно вносить через код.

Имеются три платных тарифа на 500, 1600 и 3600 рублей в месяц, а также 30-дневный пробный период.

Были сформулированы основные достоинства:

1. Функциональность и масштабируемость магазина.
2. Большой набор плагинов.
3. Подробные отчёты.

Также были сформулированы основные недостатки:

1. Малое количество бесплатных шаблонов.
2. В подавляющем большинстве – платные плагины.
3. Сложная панель управления.
4. Платное пользование.

Интерфейс системы представлен на рисунке 1.1.

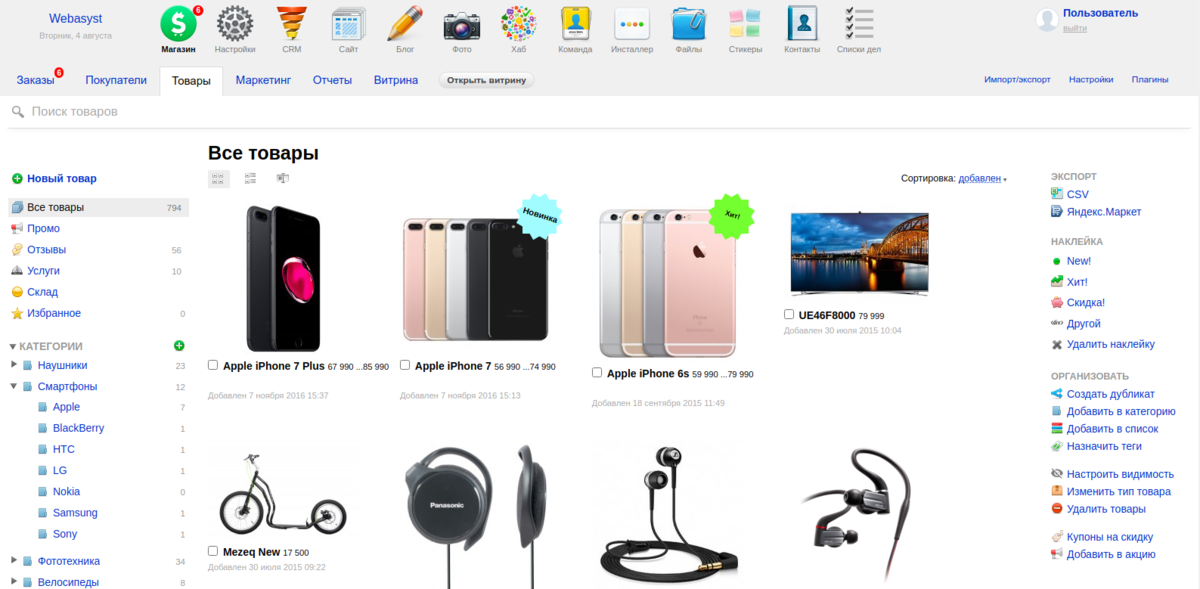


Рисунок 1.1 – Интерфейс конструктора Shop-Script

**1.3.2 AdvantShop**

AdvantShop – отечественный E-commerce конструктор. Умеренно простой в освоении, но далеко не очевидный в глазах новичков. Содержит большое количество нюансов, настроек, незаметных при поверхностном знакомстве. Самая сильная сторона – удобство управления торговой площадкой. Многие типовые действия после настройки выполняются автоматически, снижая нагрузку на администратора.

AdvantShop содержит неплохой встроенный магазин модулей, но платные образцы недешёвые, а бесплатных около трёх десятков. Шаблоны, в основном, платные. Бесплатных дизайнов всего 6.

Имеются три платных тарифа на 1100, 2200 и 3300 рублей в месяц, а также 14-тидневный пробный период.

Были сформулированы основные достоинства:

1. Гибкость кастомизации оформления.
2. Доступ к API.
3. Поддержка мультиязычных магазинов.
4. Огромный инструментарий для маркетинга.

Также были сформулированы основные недостатки:

1. Довольно высокая стоимость.
2. Крайне сложен и неинтуитивен для новичков.
3. Платное пользование.

Интерфейс системы представлен на рисунке 1.2.

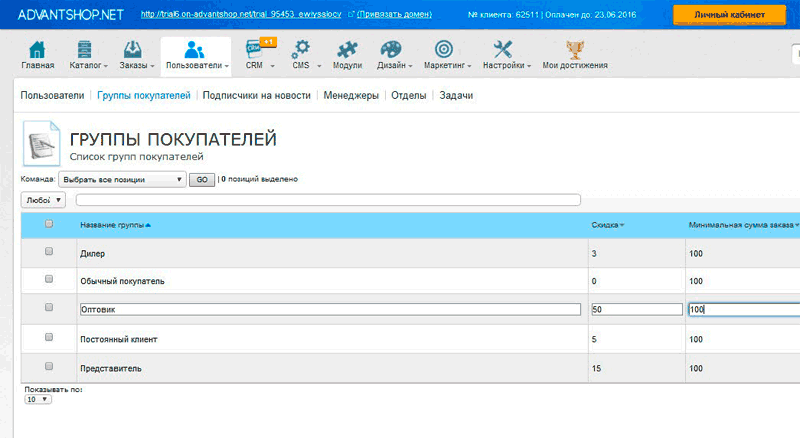


Рисунок 1.2 – Интерфейс конструктора AdvantShop

**1.3.3 OpenCart**

OpenCart – бесплатный конструктор, подходящий для управления магазином. Предлагает открытый исходный код и расширение стандартных возможностей с помощью модулей. Система достаточно интуитивна для начинающих, однако некоторые моменты в её работе потребуют редактирования кода.

Имеется достаточно большое количество бесплатных шаблонов, однако полностью отсутствует возможность кастомизации. Если же это всё-таки необходимо, то нужно редактировать код.

OpenCart предоставляет большое количество настроек и плагинов, большинство из которых платные.

Данный конструктор – бесплатный. Его можно свободно скачать и пользоваться. Единственные статьи расходов – это шаблоны и плагины.

Были сформулированы основные достоинства:

1. Открытый код.
2. Огромная база дополнительных материалов (плагины, шаблоны).
3. Интеграция с сервисами Яндекса.
4. Поддержка мультиязычности.

Также были сформулированы основные недостатки:

1. Случайное дублирование страниц по разным адресам.
2. Конфликтная работа плагинов.
3. Проблемы с массовым добавлением новых товаров.
4. Уязвимости при добавлении расширений.

Интерфейс системы представлен на рисунке 1.3.

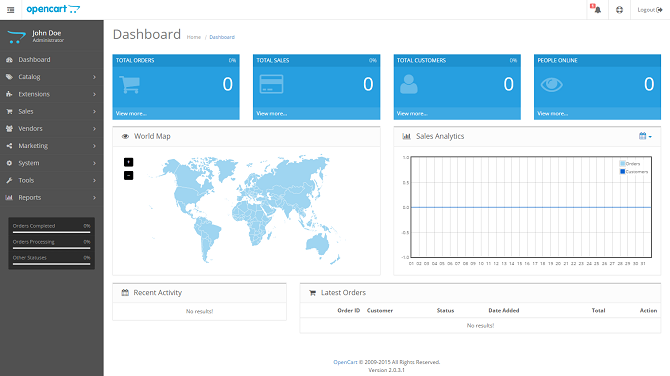


Рисунок 1.3 – Интерфейс конструктора OpenCart

**1.3.4 1C-UMI**

1C-UMI – фирменный конструктор от «1С». Поддерживается синхронизация с программным комплексом «1С: Управление торговлей». Также можно подключить социальные сети, платёжные системы, доставки, сервисы Яндекса и Google, импортировать/экспортировать товары из CSV/XML/YML.

Доступа к коду сайта здесь нет, кроме вставки CSS-стилей в шаблон. Соответственно, внедрить дополнительные интеграции не получится – придётся пользоваться тем, что есть.

Шаблонов не много, большинство платные. Присутствуют такие компоненты, как SEO, CRM, отчетности. Панель управления простая, подходит новичкам, но она крайне медленно работает и имеет устаревший дизайн.

Имеется один платный тариф на 825 рублей в месяц, а также 14-тидневный пробный период.

Были сформулированы основные достоинства:

1. Простота и понятность для новичков.
2. Функциональность SEO.
3. Синхронизация с 1С.

Также были сформулированы основные недостатки:

1. Панель управления устаревшая и медленная.
2. Слабый визуальный редактор, невозможность кастомизации.
3. Недостаточное количество шаблонов.
4. Платное пользование.

Интерфейс системы представлен на рисунке 1.4.

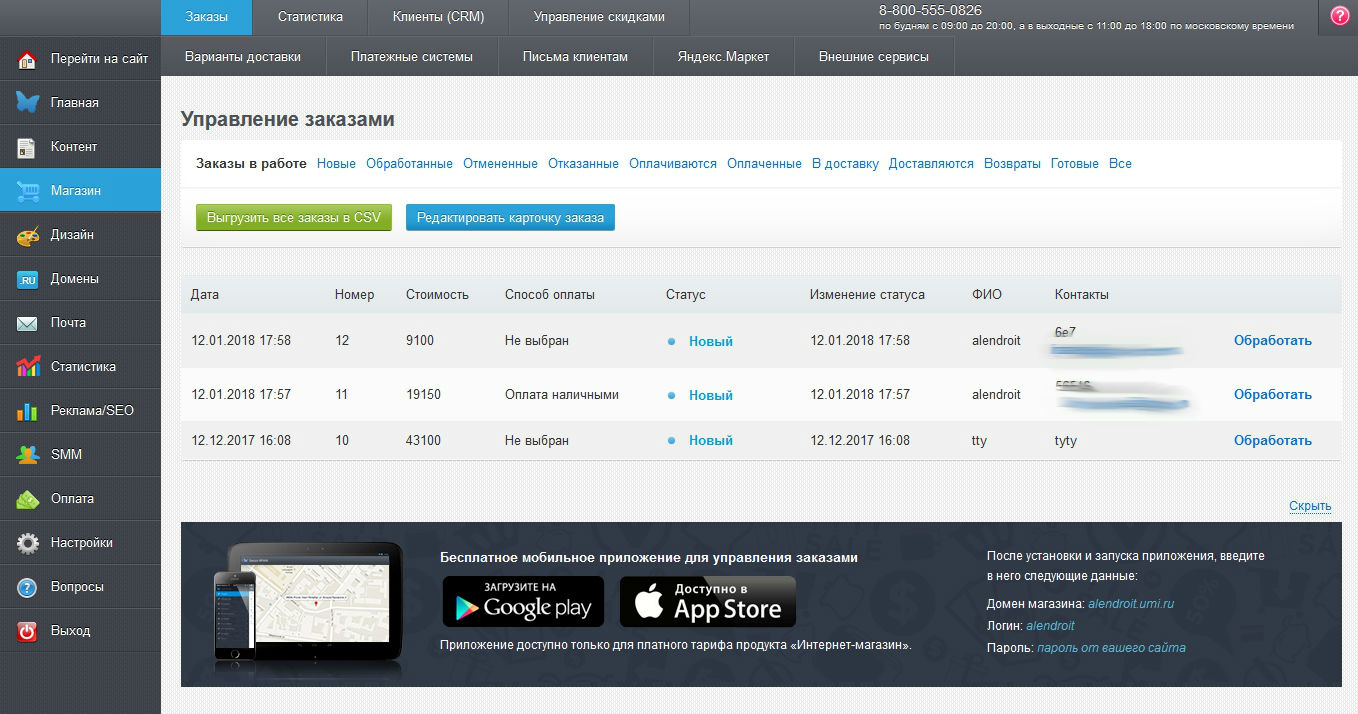


Рисунок 1.4 – Интерфейс конструктора 1C-UMI

**1.4 Выводы к разделу 1**

В данном разделе были описаны предметная область проекта, требования к разрабатываемой информационной системе, её основные необходимые функции, а также произведен обзор некоторых существующих аналогов.

Поскольку рассмотренные аналоги по тем или иным причинам (платная лицензия, платные шаблоны, сложный для понимания и неинтуитивный интерфейс и т.п.) не подходят для решения поставленных в проекте задач, было принято решение разработать конструктор интернет-магазинов, лишенный вышеописанных недостатков.

**2 Планирование проекта**

В ходе написания данного раздела следует определить структуру проекта, сроки его выполнения, а также составить календарный план-график, оценить необходимые ресурсы, идентифицировать и проанализировать риски.

**2.1 Определение структуры проекта**

Определение структуры проекта подразумевает под собой декомпозицию задач проекта на подзадачи, оценку сроков выполнения задач, а также их трудоемкость. Детализация каждой задачи значительно упростит работу над проектом.

Задача по выбору программных средств и определению требований несёт в себе следующие подзадачи:

1. Определить требования к техническим средствам. Данная подзадача описывает предъявляемые заказчиком требования к аппаратным средствам системы.
2. Определить требования к программным средствам. Данная подзадача описывает требования, касающиеся программных средств системы.

Проектирование можно условно разделить на следующие 5 этапов:

1. Разработка функциональной модели в нотации IDEF0;
2. Разработка функциональной модели в нотации IDEF3;
3. Разработка функциональной модели в нотации DFD;
4. Разработка моделей данных IDEF1X;
5. Разработка UML-диаграмм.

Выполнение данных этапов позволит подробно описать функциональную структуру разрабатываемой системы и структуру базы данных.

Также, можно выделить следующие задачи приложения:

1. Создание сайта интернет-магазина. Данная задача подразумевает разработку веб-приложения, реализующего комплексное создание сайта, включающее такие функции, как выбор названия магазина, шаблона дизайна сайта, заполнение каталога товаров, указание способов оплаты и доставки.
2. Создание панели администрирования. Данная задача подразумевает создание административного пользовательского интерфейса для управления созданным сайтом.
3. Реализация подсистемы оповещений и обратной связи. Данная задача подразумевает предоставление возможности продавцу получать SMS и/или E-mail оповещения, а также сообщения, полученные по обратной связи от покупателей.
4. Реализация подсистемы формирования отчётности. Данная задача подразумевает создание функциональности, позволяющей продавцу в ходе торговой деятельности сформировать необходимый тип отчёта по продажам.
5. Обеспечение хранения данных. Данная задача подразумевает хранение всей конфигурационной информации, а также информации о товарах в базе данных.

Что касается архитектуры проекта, то проектируемая система представляет собой клиент-серверное приложение трехуровневой архитектуры, составные части которого располагаются на одной машине.

Трёхуровневая архитектура – это такая архитектурная модель программного комплекса, которая предполагает наличие в нём трёх компонентов, также называемых слоями: клиента, сервера приложений и сервера баз данных.

В слое «Клиент» будет содержаться модуль «Веб-интерфейс», представляющий собой пользовательский веб-интерфейс и включающий в себя все необходимые для работы приложения страницы. Данный слой будет напрямую взаимодействовать со слоем «Сервер».

В слое «Сервер» будут располагаться модули «Бизнес-логика» и «Доступ к данным».

Модуль «Бизнес-логика» будет включать в себя всю логику сервера. В данный модуль будут входить следующие модули:

1. Модуль «Контроллер», представляющий собой набор функций для взаимодействия сервера и клиента;
2. Модуль «Статистика», выполняющий подсчёт различного рода статистики и в дальнейшем формирующий из полученных данных отчёты;
3. Модуль «Добавление данных в БД», отвечающий за добавление данных в базу данных;
4. Модуль «Оповещения», реализующий функциональность SMS и E-mail-оповещений;
5. Модуль «Обратная связь», выполняющий функцию обратной связи покупателя с продавцом по электронной почте.

Модуль «Доступ к данным», включающий в себя модуль «Доступ к БД», предназначен для взаимодействия слоя «Сервер» со слоем «СУБД».

В слое «СУБД» будут содержаться такие модули, как «Бизнес-логика» и «Хранилище».

Модуль «Бизнес-логика» включает в себя модули «Триггеры» и «Хранимые процедуры», предназначенные для поддержания стабильной работы с БД и обеспечения целостности данных. Благодаря им будет выполняться автоматическое обновление базы данных.

Модуль «Хранилище», собственно, предназначен для хранения данных и корректной работы всей системы.

На рисунке 2.1 представлена схема вышеописанных слоёв приложения.

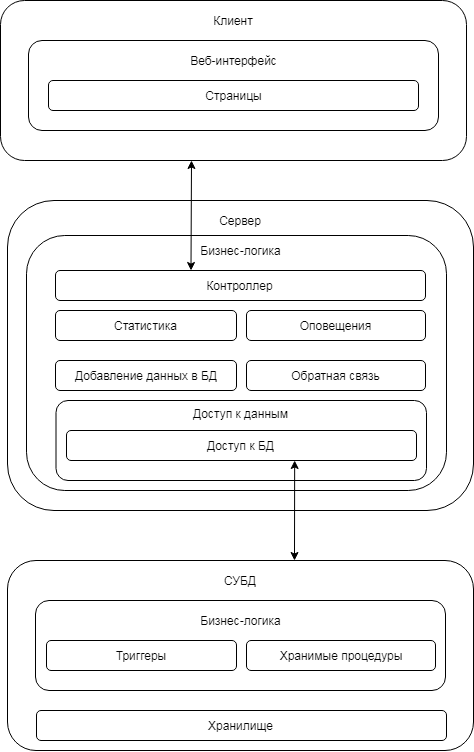


Рисунок 2.1 – Схема слоёв проектируемой системы

Также, кроме выбора архитектуры проектируемой системы следует определиться с используемыми технологиями и инструментариями для разработки.

Для написания всех частей приложения был выбран свободно распространяемый редактор кода Microsoft Visual Studio Code. Несмотря на возможность его бесплатного использования, VS Code предоставляет огромные возможности для работы практически с любыми языками программирования, структурирования, вёрстки. Также, имеется помощь при написании кода, подсветка синтаксиса и множество бесплатных плагинов, использование которых может превратить простой редактор кода, например, в локальный сервер или даже в упрощенную интегрированную среду разработки.

Для разработки будут использоваться такие средства, как язык разметки HTML, каскадные таблицы стилей CSS, клиентский язык программирования JavaScript, серверный язык программирования PHP, и локальный сервер OpenServer, на котором и будет вестись разработка системы. В дальнейшем локальный сервер будет заменен веб-хостингом.

В качестве средств для структурного и объектно-ориентированного моделирования было выбрано веб-приложение draw.io.

В качестве СУБД была выбрана MySQL, так как она свободна в использовании, является одной из самых быстрых баз данных, имеющихся на современном рынке, крайне проста в обращении, легко работает с большими объёмами данных, легко масштабируется, а также крайне популярна для веб-разработки.

**2.2 Определение сроков выполнения проекта и составление его календарного плана**

После определения структуры проекта необходимо выбрать систему управления проектами, провести календарное планирование проекта, оценить необходимые ресурсы и выполнить их распределение.

**2.2.1 Выбор системы управления проектами**

Система управления проектами – это набор программных инструментов и методов, процедур и методологий, обеспечивающий управление проектами и повышающий эффективность их выполнения. Системы управления проектами позволяют грамотно распределять ресурсы между задачами и контролировать процесс выполнения проекта, а также его сроки.

Рассмотрим наиболее популярные и используемые системы управления проектами, из которых выберем одну для дальнейшего использования при выполнении курсового проекта [11].

**2.2.1.1 Trello**

Trello – это kanban-ориентированный инструмент для ведения личных и командных проектов (рисунок 2.2). Данный инструмент удобен, прост в понимании и хорошо реализован визуально. Имеется бесплатная версия с урезанной функциональностью.

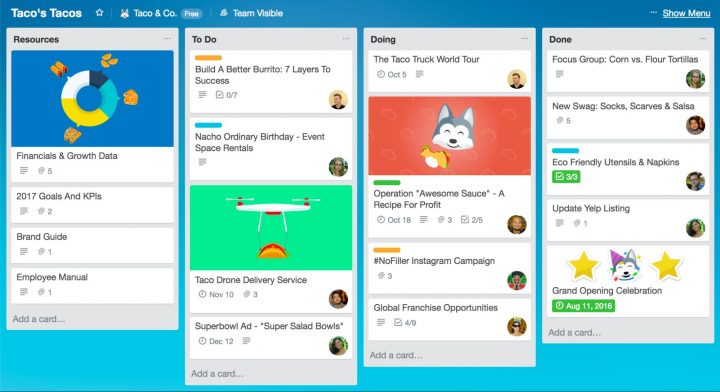


Рисунок 2.2 – Интерфейс Trello

Trello дает полную картину по проекту в трёх категориях: что уже сделано, что в работе, что предстоит сделать. С помощью drag'n'drop карточки легко перетягивать из одной категории в другую. Имеется возможность прикреплять файлы, приглашать участников, вставлять ссылки и т.д.

Инструмент работает и в онлайне, и на мобильных операционных системах Android и iOS, и на персональных компьютерах под управлением Windows и MacOS.

Были сформулированы основные достоинства:

1. Простота и понятность для новичков.
2. Многоязычность.
3. Возможность работать бесплатно.
4. Наличие мобильного приложения.

Также были сформулированы основные недостатки:

1. Дополнительные платные функции.
2. Не эффективен для крупных долгосрочных проектов.

**2.2.1.2 Asana**

Asana – платформа для организации работы и управления проектами (рисунок 2.3). Подойдет для небольших команд и личного пользования. Asana помогает фокусироваться на целях, самих проектах, ежедневных задачах, вехах и дедлайнах.

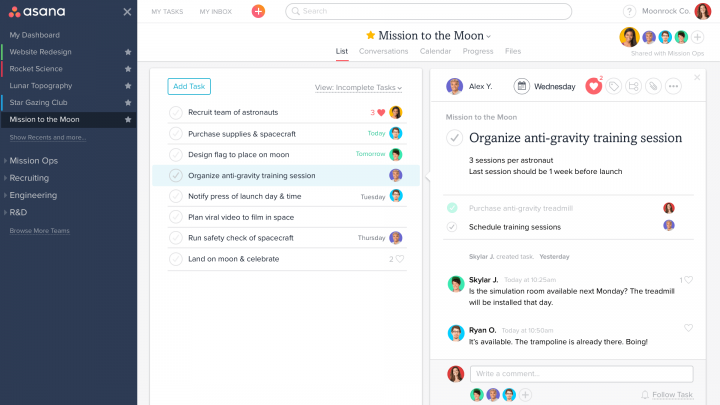


Рисунок 2.3 – Интерфейс Asana

Сервис имеет хорошую интерфейсную визуализацию, качественно проработан дизайн. Asana отлично подходит для ведения маркетинговых и креативных кампаний. Но этими сферами применение инструмента не ограничивается.

Поддерживаются три платформы – онлайн версия, Android, iOS.

Были сформулированы основные достоинства:

1. Проработанный дизайн.
2. Синхронизация с другими сервисами.
3. Широкая область применения.
4. Имеется бесплатный пробный период.

Также были сформулированы основные недостатки:

1. Не подходит для крупных команд.
2. Отсутствие русского языка.
3. Платное пользование.
4. Отсутствие десктопной версии.

**2.2.1.3 Bitrix 24**

Bitrix 24 – мощная комбинированная система для организации CRM и управления проектами (рисунок 2.4). В данной системе возможность управлять проектами представлена лишь как одна из других возможностей: CRM для продаж, контакт-центр для коммуникации с клиентами, возможность создавать сайты и просто средство для общения внутри компании.

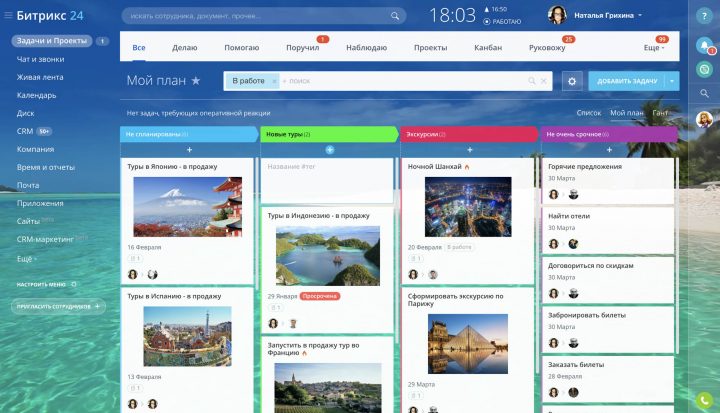


Рисунок 2.4 – Интерфейс Bitrix 24

В сервисе легко создавать задачи, которые можно организовать в виде списка, диаграммы Ганта, kanban-доски. Имеется возможность назначать роли: ответственный, постановщик и наблюдатель. Возможно бесплатное использование при условии не более 12 пользователей.

Поддерживаются три платформы – онлайн версия, Android, iOS.

Были сформулированы основные достоинства:

1. Связь задач с CRM.
2. Поддержка мультиязычности, включая русский язык.
3. Возможность оценивать эффективность работы сотрудников.
4. Шаблоны для повторяющихся проектов.

Также были сформулированы основные недостатки:

1. Крайне сложный интерфейс.
2. Требует время на изучение, обучение сотрудников и внедрение.
3. Платное пользование.
4. Отсутствие десктопной версии.

**2.2.1.4 GanttPRO**

GanttPRO – онлайн диаграмма Ганта с возможностью переключения на kanban-доску задач (рисунок 2.5). Работа с системой сводится к двум этапам: создание задач, которые размещаются по вертикали, и указание их дат, которые отображаются по горизонтали. На выходе имеем визуализированный график, по которому можно оценить объем работ.

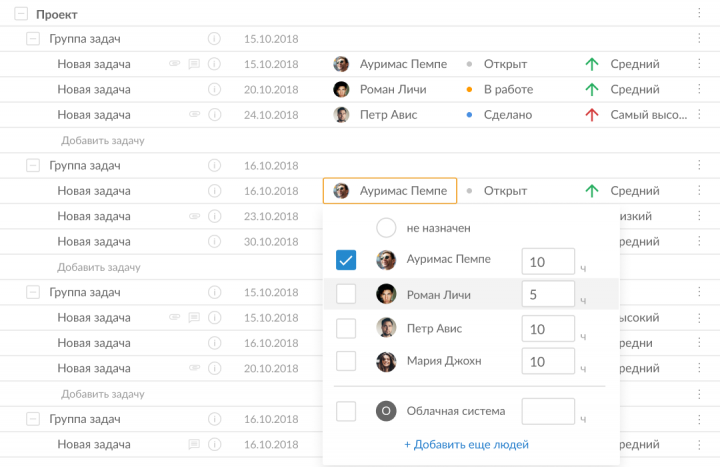


Рисунок 2.5 – Интерфейс GanttPRO

При этом GanttPRO – это не только диаграмма Ганта. Инструмент подходит для управления ресурсами и стоимостью проекта, а также для командной работы. Кроме того, сфера применения данной системы не ограничена: строительство, разработка ПО, сфера услуг, организация мероприятий и т.д. Для этих целей сервис предлагает готовые шаблоны.

Система работает только как веб-сервис. Имеется двухнедельный бесплатный пробный период с полной функциональностью.

Были сформулированы основные достоинства:

1. Удобный пользовательский интерфейс.
2. Простота в изучении.
3. Наличие русского языка.
4. Готовые шаблоны проектов для быстрого старта.

Также были сформулированы основные недостатки:

1. Отсутствует возможность отображения всех проектов на одном экране.
2. Малое количество интеграций.
3. Платное пользование.
4. Отсутствие десктопной и мобильных версий.
5. Отсутствие мультиязычности.

Выше были рассмотрены наиболее популярные системы управления проектами. Для составления плана работы над ВКР была выбрана система управления проектами Trello, поскольку данная система хоть и проста в освоении и использовании, но, так или иначе, несёт в себе достойную функциональность при бесплатном использовании, а также, что немаловажно, имеет удобный и понятный русскоязычный интерфейс.

**2.2.2 Календарное планирование проекта**

Одна из самых важных и сложных задач при планировании проекта – оценка времени для его реализации. Своевременному завершению проекта способствует соблюдение сроков выполнения соответствующих задач. Для определения сроков выполнения задач необходимо представить их в виде элементов с чётко определёнными сроками.

Сроки выполнения проекта:

1. Дата начала проекта – 11.09.2020 г.
2. Дата завершение проекта – 30.05.2021 г.

Затем необходимо определить сроки выполнения каждой из поставленных задач. Сроки выполнения задач проекта представлены в таблице 1.

Таблица 1 – Сроки выполнения задач проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование задачи | Длительность, ч. | Обоснование длительности | Дата начала | Дата окончания |
| **1 Определение требований** | 40 | С учётом выбора средств | 11.09.2020 | 25.09.2020 |
| 1.1 Требования к аппаратным средствам | 20 | С учётом выбора и приобретения средств | 11.09.2020 | 18.09.2020 |
| 1.2 Требование к программным средствам | 20 | С учётом выбора и установки средств | 18.09.2020 | 25.09.2020 |
| **2 Проектирова-ние структуры проекта** | 80 | С учётом изучения моделей и их проектирования | 25.09.2020 | 23.10.2020 |
| 2.1 Проектиро-вание модели IDEF0 | 20 | С учётом изучения модели и её исправления | 25.09.2020 | 02.10.2020 |
| 2.2 Проектиро-вание модели IDEF3 | 10 | С учётом изучения модели и её исправления | 02.10.2020 | 06.10.2020 |
| 2.3 Проектиро-вание модели DFD | 10 | С учётом изучения модели и её исправления | 06.10.2020 | 09.10.2020 |
| 2.4 Проектиро-вание модели IDEF1X | 20 | С учётом изучения модели и её исправления | 09.10.2020 | 16.10.2020 |
| 2.5 Проектиро-вание UML-  диаграмм | 20 | С учётом изучения модели и её исправления | 16.10.2020 | 23.10.2020 |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование задачи | Длительность, ч. | Обоснование длительности | Дата начала | Дата окончания |
| **3 Разработка программы** | 300 | С учётом написания программы и её проверки | 23.10.2020 | 05.02.2021 |
| 3.1 Разработка функции формирования внешнего отображения сайта | 75 | С учётом написания кода и его проверки | 23.10.2020 | 18.11.2020 |
| 3.2 Разработка функции добавления товаров в каталог | 75 | С учётом написания кода и его проверки | 18.11.2020 | 14.12.2020 |
| 3.3 Разработка функции реализации оповещений и обратной связи | 75 | С учётом написания кода и его проверки | 14.12.2020 | 09.01.2021 |
| 3.4 Разработка функции формирования отчётностей | 75 | С учётом написания кода и его проверки | 09.01.2021 | 05.02.2021 |
| **4 Обеспечение хранения данных** | 120 | С учётом создания БД и проверкой её функционирования | 05.02.2021 | 19.03.2021 |
| **5 Разработка графического пользователь-ского веб-интерфейса** | 200 | С учётом написания программы и её проверки | 19.03.2021 | 28.05.2021 |

Все вышеуказанные сроки выполнения задач были указаны в ранее выбранной системе управления проектами Trello (рисунок 2.6).

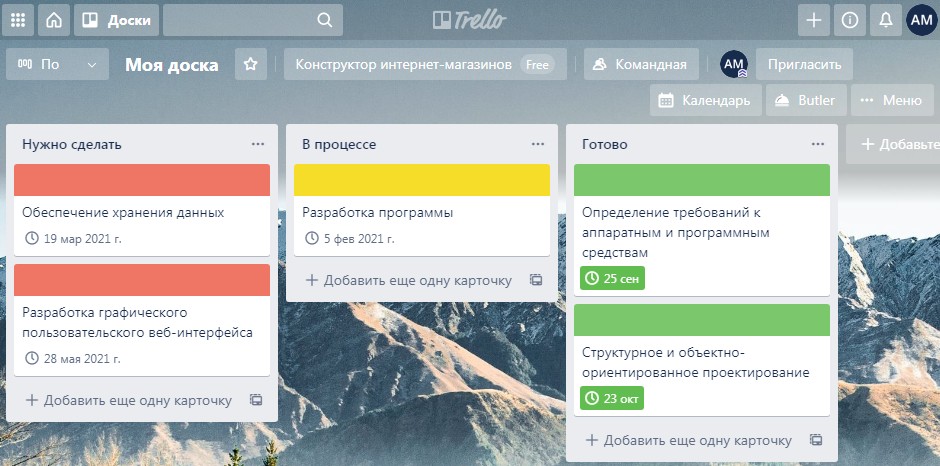


Рисунок 2.6 – Распланированный проект в Trello

**2.2.3 Оценка необходимых ресурсов**

В данном разделе будет приведена оценка ресурсов, которые необходимы и которые будут использоваться для выполнения всего проекта. Под оценкой ресурсов подразумевается то, какие ресурсы и в каком количестве понадобятся для выполнения проекта.

Ресурсы можно классифицировать по пяти категориям: трудовые, технические, программные, финансовые и временные.

В качестве трудовых ресурсов выступает студент, который выполняет все поставленные работы по выполнению проекта, и заказчик, задающий требования проекта.

В качестве технических ресурсов будет использоваться персональный компьютер с выходом в интернет.

В качестве программных ресурсов будут выступать редактор исходного кода Microsoft Visual Studio Code, в котором на языках HTML, JavaScript, PHP будет производиться большая часть разработки системы, свободная система управления базами данных MySQL, в которой будет создаваться база данных, локальный сервер OpenServer, который будет обеспечивать работу серверного языка программирования PHP и работу вышеупомянутой СУБД MySQL, веб-приложение draw.io, в котором будут создаваться все необходимые диаграммы, текстовый редактор Microsoft Word, в котором будет выполняться написание пояснительной записки, веб-браузер Google Chrome, при помощи которого будет производиться поиск информации в сети интернет.

Вышеперечисленные используемые программные ресурсы не требуют финансовых вложений, поэтому финансовыми ресурсами будет являться только затраты на оплату электроэнергии, на которой функционируют технические ресурсы.

В качестве временных ресурсов будет выступать тот период времени, который предоставлен на выполнение данного проекта.

**2.2.4 Распределение ресурсов между задачами**

После оценки всех необходимых ресурсов следует распределить их по ранее описанным задачам проекта. Данное распределение представлено в таблице 2.

Таблица 2 – Распределение ресурсов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование задачи | Трудовые ресурсы | Технические ресурсы | Программные ресурсы |
| **1 Определение требований** | Студент, заказчик | Компьютер, интернет | Microsoft Word,  Google Chrome |
| 1.1 Требования к аппаратным средствам | Студент, заказчик | Компьютер, интернет | Microsoft Word,  Google Chrome |
| 1.2 Требование к программным средствам | Студент, заказчик | Компьютер, интернет | Microsoft Word,  Google Chrome |
| **2 Проектирование структуры проекта** | студент | Компьютер, интернет | Google Chrome, draw.io |
| 2.1 Проектирование модели IDEF0 | студент | Компьютер, интернет | Google Chrome, draw.io |
| 2.2 Проектирование модели IDEF3 | студент | Компьютер, интернет | Google Chrome, draw.io |
| 2.3 Проектирование модели DFD | студент | Компьютер, интернет | Google Chrome, draw.io |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование задачи | Трудовые ресурсы | Технические ресурсы | Программные ресурсы |
| 2.4 Проектирование модели IDEF1X | студент | Компьютер, интернет | Google Chrome, draw.io |
| 2.5 Проектирование UML-  диаграмм | студент | Компьютер, интернет | Google Chrome, draw.io |
| **3 Разработка программы** | студент | Компьютер, интернет | VS Code, OpenServer |
| 3.1 Разработка функции формирования внешнего отображения сайта | студент | Компьютер, интернет | VS Code, OpenServer |
| 3.2 Разработка функции добавления товаров в каталог | студент | Компьютер, интернет | VS Code, OpenServer |
| 3.3 Разработка функции реализации оповещений и обратной связи | студент | Компьютер, интернет | VS Code, OpenServer |
| 3.4 Разработка функции формирования отчётностей | студент | Компьютер, интернет | VS Code, OpenServer |
| **4 Обеспечение хранения данных** | студент | Компьютер, интернет | VS Code, OpenServer, MySQL |
| **5 Разработка графического пользовательского веб-интерфейса** | студент | Компьютер, интернет | VS Code, OpenServer |

**2.3 Идентификация и анализ рисков**

В данном разделе будут произведены идентификация рисков, их качественный и количественный анализ, а также будет разработано планирование реагирования на риски.

Под риском в проектной деятельности понимается вероятностное событие или условие, которое в случае возникновения имеет позитивное или негативное воздействие на одну или несколько целей проекта, например сроки, стоимость, содержание или качество. Риск характеризуется определенными источниками или причинами и имеет последствия, т.е. оказывает влияние на результаты проекта [12].

Под идентификацией рисков понимается определение того, какие риски могут повлиять на проект, и документальное оформление их характеристик [12].

Анализ рисков бывает качественный и количественный. Качественный анализ – это оценка и расположение рисков по степени их приоритета для дальнейшего анализа или обработки путем суммирования вероятности их возникновения и воздействия на проект. Количественный анализ – это определение вероятности возникновения рисков и влияния последствий рисков на проект [12].

**2.3.1 Идентификация рисков**

Идентификация рисков позволяет определить, какие риски влияют на проект, и описывает характеристики этих рисков. Идентификация рисков проводится регулярно на протяжении реализации всего проекта.

Риски можно разделить на пять категорий:

1. Технологические – это риски, связанные с технологиями, используемыми в проекте;
2. Технические – это риски, связанные с техникой, используемой в проекте;
3. Риски, связанные с требованиями;
4. Риски, связанные с организацией процесса выполнения проекта;
5. Риски оценивания.

Исходя из вышеописанных категорий, выделим те риски, которые могут повлиять на процесс выполнения данного проекта. Результат определения рисков представлен в таблице 3.

Таблица 3 – Риски проекта

|  |  |
| --- | --- |
| **Категория рисков** | **Риск** |
| Технологические | Сбой программы, повлекший за собой потерю рабочих данных |
| Неверный подбор программных средств |
| Технические | Выход оборудования из строя |
| Недостаточное техническое оснащение |
| Организационные | Прокрастинация процесса выполнения проекта |
| Выход из установленных временных рамок |
| Риски, связанные с требованиями | Изменение требований в ходе работы |
| Несоответствие требованиям по окончании работы над проектом |
| Риски оценивания | Неправильная оценка времени работы над проектом |
| Неверное оценивание экономической составляющей проекта |

**2.3.2 Качественный анализ рисков**

Качественный анализ рисков – это оценка условий возникновения рисков и определение их вероятностей и влияния на проект [12]. Вероятность и влияние рисков на проект могут определяться качественными оценками: очень высокие, высокие, средние, низкие, очень низкие. Примеры оценки данных показателей приведены в таблице 4 [13].

Таблица 4 – Оценки вероятности риска и его влияния

|  |  |  |  |
| --- | --- | --- | --- |
| Вероятность риска, % | Влияние на проект | Качественная характеристика | Оценка (ранг) |
| Менее 5 (очень малая) | Очень слабое | Событие может произойти в исключительных случаях. | 0,01 |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Вероятность риска, % | Влияние на проект | Качественная характеристика | Оценка (ранг) |
| 5 – 10 (малая) | Слабое | Редкое событие, но уже однажды происходило. | 0,1 |
| 10 – 30 (средняя) | Среднее | Событие произошло 1-2 раза на других проектах. | 0,2 |
| 30 – 60 (высокая) | Сильное | Вполне вероятное событие. Часто случалось на других проектах. | 0,4 |
| 60 – 99 (очень высокая) | Очень сильное | Событие, скорее всего, случится. | 0,8 |

Опираясь на вышеприведенные показатели, выполним качественный анализ ранее выделенных рисков проекта (таблица 5).

Таблица 5 – Качественный анализ рисков

|  |  |  |
| --- | --- | --- |
| Риск | Вероятность (ранг) | Влияние (ранг) |
| Сбой программы, повлекший за собой потерю рабочих данных | 0,4 | 0,8 |
| Неверный подбор программных средств | 0,1 | 0,2 |
| Выход оборудования из строя | 0,1 | 0,2 |
| Недостаточное техническое оснащение | 0,1 | 0,1 |
| Прокрастинация процесса выполнения проекта | 0,4 | 0,4 |
| Выход из установленных временных рамок | 0,2 | 0,4 |
| Изменение требований в ходе работы | 0,1 | 0,2 |
| Несоответствие требованиям по окончании работы над проектом | 0,1 | 0,2 |
| Неправильная оценка времени работы над проектом | 0,4 | 0,4 |
| Неверное оценивание экономической составляющей проекта | 0,2 | 0,2 |

Далее, построим карту рисков для наглядного представления всех рисков проекта. Карта рисков строится в виде матрицы, состоящей из мер риска. В свою очередь, мера риска вычисляется как произведение показателя вероятности риска на показатель его воздействия. Построенная карта рисков представлена в таблице 6. Чем темнее цвет ячейки, тем выше уровень риска, соответственно, можно выделить три уровня риска: низкий, средний и высокий.

Таблица 6 – Карта рисков

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вероятность | Мера риска = вероятность \* влияние | | | | |
| 0,8 | 0,008 | 0,08 | 0,16 | 0,32 | 0,64 |
| 0,4 | 0,004 | 0,04 | 0,08 | 0,16 | 0,32 |
| 0,2 | 0,002 | 0,02 | 0,04 | 0,08 | 0,16 |
| 0,1 | 0,001 | 0,01 | 0,02 | 0,04 | 0,08 |
| 0,01 | 0,0001 | 0,001 | 0,002 | 0,004 | 0,008 |
| Влияние | 0,01 | 0,1 | 0,2 | 0,4 | 0,8 |

В таблице 7 представлены риски данного проекта с указанием их мер и уровня, к которому они относятся.

**2.3.3 Количественный анализ рисков**

Количественный анализ рисков – это процесс, определяющий потенциальное воздействие идентифицированных рисков и включающий в себя численное определение величин и уровней рисков. Анализ карты рисков, построенной в предыдущем подпункте (таблица 6), позволяет разделить все риски данного проекта на 3 уровня. Данная классификация представлена в таблице 7.

Таблица 7 – Риски проекта

|  |  |  |
| --- | --- | --- |
| Риск | Мера риска | Уровень риска |
| Сбой программы, повлекший за собой потерю рабочих данных | 0,32 | Высокий |
| Неправильная оценка времени работы над проектом | 0,16 | Высокий |
| Прокрастинация процесса выполнения проекта | 0,16 | Высокий |
| Неверный подбор программных средств | 0,02 | Средний |
| Неверное оценивание экономической составляющей проекта | 0,04 | Средний |
| Выход из установленных временных рамок | 0,08 | Средний |
| Изменение требований в ходе работы | 0,02 | Средний |

Продолжение таблицы 7

|  |  |  |
| --- | --- | --- |
| Риск | Мера риска | Уровень риска |
| Выход оборудования из строя | 0,02 | Средний |
| Несоответствие требованиям по окончании работы над проектом | 0,02 | Средний |
| Недостаточное техническое оснащение | 0,01 | Низкий |

Данные, представленные в таблице 7, позволяют явно выделить риски, имеющие высокий, средний и низкий уровень значимости в проекте.

**2.3.4 Планирование реагирования на риски**

Планирование реагирования на риски – это разработка методов и процедур, направленных на повышение положительных возможностей и снижение угроз для достижения целей проекта [14]. Способы реагирования рассматриваются для каждого риска отдельно.

В предыдущем подпункте было выделено три риска с наивысшим влиянием на проект.

Для уменьшения влияния риска сбоя программы, влекущего за собой потерю рабочих данных, необходимо следить за общим состоянием программного обеспечения компьютера, в частности выполнять проверку на всевозможные внешние угрозы, например, вирусы. Также, следует иметь резервные копии всех необходимых данных на компьютере, съемных носителях, в облаке и производить сохранения после каждого изменения в работе. Кроме того, не лишним станет использование в работе удалённого репозитория, например, GitHub.

Для уменьшения влияния таких рисков, как неправильная оценка времени работы над проектом и прокрастинация процесса выполнения проекта, необходимо точно обозначить сроки выполнения поставленных в проекте задач и постоянно с ними сверяться. Также, следует реализовать управление выполнением проекта в какой-либо системе управления проектами. Следует отметить, что данное планирование подойдёт так же и для реагирования на влияние выхода из установленных временных рамок.

Также, было выделено шесть рисков со средним влиянием на проект.

Неверный подбор программных средств. Влияние данного риска довольно легко уменьшить, если еще в самом начале выполнения проекта ответственно и тщательно подойти к анализу поставленных требований и предметной области. Также, следует вести разработку таким образом, чтобы быть готовым в любой незапланированный момент преобразовать и подготовить все необходимые данные для пользования другим программным средством.

Неверное оценивание экономической составляющей проекта. Для уменьшения влияния данного риска, аналогично, необходимо грамотно оценить поставленные требования и изучить предметную область, а также внимательно проводить необходимые вычисления.

Изменение требований в ходе работы. Для уменьшения влияния данного риска следует выполнять переоценку проекта каждый раз, когда добавляются новая функция программы или изменяется существующая. Сюда также можно отнести риск несоответствия требованиям по окончании работы над проектом

Выход оборудования из строя. Для уменьшения влияния данного риска, опять же, следует обеспечить резервное хранение всех необходимых данных и вести работу через репозиторий, а также регулярно проверять рабочее оборудование на наличие неисправностей. В случае необходимости – проводить необходимые ремонтные мероприятия.

Наконец, был определен всего один риск, слабо влияющий на проект – недостаточное техническое оснащение. Низкий уровень риска говорит о том, что данное событие вероятно практически не наступит. Это объясняется тем, что поставленные задачи проекта не являются слишком требовательными к техническому оснащению рабочего места.

**2.4 Выводы к разделу 2**

В данном разделе определена структура проекта и сроки его выполнения. Произведён выбор системы управления проектами – выбрана система Trello.

Также, было выполнено календарное планирование проекта, обозначены сроки проекта: начало 11.09.2020 и окончание 30.05.2021. Произведены оценка необходимых ресурсов и их распределение между задачами. Распланированный календарный график проекта был описан в системе Trello.

В заключение, были выполнены идентификация рисков и их качественный и количественный анализы. Риски были распределены на три уровня: низкий, средний, высокий. В результате работы с рисками были определены три риска высокого уровня, шесть рисков среднего уровня и один риск низкого уровня. Для каждого идентифицированного риска было распланировано соответствующее реагирование.

**4 Экономическое обоснование разработки программного обеспечения**

В данном разделе будут выполнены расчёты затрат на создание ПО, расчёт выручки и прибыли от реализации программного продукта и расчёт затрат, связанных с покупкой, внедрением и использованием ПО.

**4.1 Расчёт затрат на создание ПО, цены и прибыли его реализации**

Для расчета затрат на разработку программного продукта будем основываться на том, что над проектом работает группа, состоящая из двух человек – руководителя проекта и инженера-программиста.

Состав группы приведен в таблице 8.

Таблица 8 – Состав группы разработчиков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование должности | Численность, чел. | Базовая ставка з/п, руб. | Повышающий коэффициент | Месячный оклад, руб. |
| Руководитель проекта | 1 | 13879 | 3,4 | 47188,6 |
| Инженер - программист | 1 | 12384 | 1,9 | 23529,6 |

Базовые ставки разработчиков были получены, как усредненные значения базовых ставок аналогичных специалистов, взятых по городу Киров на 2020 год.

Повышающий коэффициент, согласно трудовому кодексу Российской Федерации, должен быть не больше 5,0.

Время участия в создании проекта каждого специалиста определяется на основе перечня работ и трудоемкости их выполнения. Трудоемкость каждой работы определяется по формуле (1):

, (1)

где – минимальное время, необходимое для выполнения работы;

– максимальное время, необходимое для выполнения работы.

Рассчитанная трудоемкость выполнения поставленных работ представлена в таблице 9.

Таблица 9 – Работы и трудоёмкость их выполнения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование работы | , ч | , ч | , ч | В том числе заняты | |
| Руководитель, ч | Программист, ч |
| 1 Анализ предметной области | 30 | 40 | 34 | 4 | 30 |
| 2 Проектирование | 50 | 60 | 54 | 11 | 43 |
| 3 Разработка ПО | 155 | 185 | 167 | 18 | 149 |
| 4 Отладка и тестирование | 40 | 50 | 44 | - | 44 |
| 5 Написание инструкции для пользователя | 10 | 15 | 12 | - | 12 |
| 6 Экономическое обоснование системы | 20 | 25 | 22 | 4 | 18 |
| 7 Разработка документации | 10 | 15 | 12 | - | 12 |
| **Всего:** | 315 | 390 | 345 | **37** | **308** |

Для удобства и точности проведения дальнейших расчетов сгруппируем данные, приведенные в таблице 9, по комплексам. Полученный результат представлен в таблице 10.

Таблица 10 – Комплексы работ по созданию программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование комплекса работ | Обозначение | , ч | В том числе заняты | |
| Руково-дитель, ч | Програм-мист, ч |
| 1 Создание математического обеспечения, написание программы |  | 255 | 33 | 222 |
| 2 Отладка, тестирование |  | 44 | - | 44 |
| 3 Прочие затраты по разработке программы |  | 12 | - | 12 |
| 4 Маркетинговые исследования |  | 22 | 4 | 18 |
| 5 Оформление программного продукта |  | 12 | - | 12 |
| **Всего:** |  | 345 | **37** | **308** |

Данные из таблицы 10 служат исходной базой для дальнейшего расчета затрат на создание программы. Ниже приведем все необходимые расчётные формулы.

Затраты на зарплату программистов рассчитываются по формуле (2):

, (2)

где – количество операторов (команд) в программе;

– средняя (стандартная) трудоемкость разработки одного оператора (команды) для используемого языка программирования;

– коэффициент новизны разрабатываемой программы;

– трудоемкость других видов работ, выполняемых программистами при создании математического обеспечения и написании программы (разработки общих принципов построения программы, ее структуры, входных и выходных форм и др.);

– месячный оклад работника в соответствии с его категорией или тарифным разрядом (см. таблицу 8);

– длительность смены (8 часов);

– среднее число рабочих дней в месяце (примем 21 день).

В случае разрабатываемой системы определить количество операторов в программе не представляется возможным, так как общее количество кода невероятно большое, а именно несколько тысяч строк кода, каждая из которых может содержать около 5-10 различных либо повторяющихся операторов. Поэтому расчет следует производить по формуле (3). Эта же формула отвечает и за расчёт затрат на выплату зарплаты других работников временной творческой группы, участвующих в разработке математического обеспечения и написании программы.

, (3)

где Р – число работников определенной квалификации, участвующих в разработке математического обеспечения и написании программы;

– время участия работника определенной квалификации в разработке математического обеспечения и написании программы, в часах (см. табл. 10).

Затраты на заработную плату работников, участвующих в создании математического обеспечения и написании программы, определяются по формуле (4):

, (4)

где – заработная плата программистов, участвующих в создании математического обеспечения и написании программы;

– заработная плата других работников временной творческой группы, принимающих участие в разработке математического обеспечения и написании программы (руководитель группы, оператор).

Затраты на создание математического обеспечения и написание программы определяются по формуле (5):

, (5)

где – затраты на выплату заработной платы работникам, участвующим в создании математического обеспечения и написании программы;

П – премия, предусмотренная для работников, участвующих в создании программы (примем в размере 20-25 % от );

– выплаты по районному коэффициенту (для г. Кирова установлены в размере 15% от ( + П));

– общая ставка страховых взносов (30% и ФСС НС 0,2%);

– фонд оплаты труда работников, участвующих в создании математического обеспечения и написании программы (включает в себя зарплату, премию и выплаты по районному коэффициенту – 15%);

– накладные расходы организации, где разрабатывается программа (затраты на отопление, освещение, на содержание административно-управленческого персонала и др. Примем в размере 100-200% от ).

Стоимость одного часа эксплуатации компьютера можно определить укрупнено по формуле (6):

, (6)

где – минимальная заработная плата, с 1 января 2020 года составляет 12130 руб.;

– повышающий коэффициент, соответствующий ставке инженера (программиста);

– коэффициент, учитывающий накладные и другие расходы, связанные с работой компьютера (примем равным 3-4 %).

Затраты, связанные с работой компьютера при разработке ПО, можно рассчитать укрупнено по формуле (7):

, (7)

где – машинное время, необходимое для установки программы, редактирования, отладки, тестирования, корректировки и выполнения, в часах (см. табл. 10);

– стоимость одного часа эксплуатации компьютера определенной модели.

Затраты на разработку программы рассчитываются по формуле (8):

, (8)

где – затраты на создание математического обеспечения и написание программы;

– затраты, связанные с работой компьютера при разработке программы (установка программы и редактирование, отладка, тестирование, корректировка);

– прочие затраты, связанные с разработкой программы (изучение задания, литературы, патентов, анализ проблемы и существующих алгоритмов, проведение экономических расчетов и др. Примем в размере 25-50 % от ЗМО).

Общий фонд оплаты труда работников, участвующих в создании программы, определяется по формуле (9):

, (9)

где – время участия специалиста определенной квалификации в создании программы (в часах);

F – число специалистов определенной квалификации, участвующих в создании программы.

Общие затраты на создание программы находятся по формуле (10):

, (10)

где – затраты на разработку программы;

– затраты на маркетинговые исследования (можно принять в размере 10-20% от ЗРП);

– затраты на оформление программы и подготовку ее к продаже (примем в размере 15-25% от );

– налоги, включаемые в затраты организации, где создается программа.

Далее произведем все необходимые расчеты по формулам (2) – (10).

По формуле (3) найдем заработную плату инженера-программиста, участвующего в создании математического обеспечения и написание программы:

Рассчитаем заработную плату других работников временной группы, участвующие в создании математического обеспечения и написании программы, по той же формуле (3):

Далее рассчитаем затраты на заработную плату работников, участвующих в создании математического обеспечения и написании программы, по формуле (4):

Рассчитаем размер премии, приняв её размер равным 25% от :

Рассчитаем выплаты по районному коэффициенту, равному 15% от суммы заработной платы и премии ( + П):

Выполнив расчеты всех необходимых значений, рассчитаем фонд оплаты труда работников, участвующих в создании математического обеспечения и написании программы, учитывая то, что данный фонд включает в себя зарплату, премию и выплаты по районному коэффициенту:

Накладные расходы организации рассчитаем в размере 100% от :

Рассчитаем затраты на создание математического обеспечения и написание программы по формуле (5):

Рассчитаем стоимость одного часа эксплуатации компьютера по формуле (6):

Рассчитаем затраты, связанные с работой компьютера при разработке ПО, по формуле (7):

Рассчитаем прочие затраты, которые составляют 30% от :

Рассчитаем затраты на разработку программы по формуле (8):

Рассчитаем затраты на маркетинговые исследования как 10% от :

Рассчитаем затраты на оформление программы как 15% от :

Рассчитаем общий фонд оплаты труда работников, участвующих в создании программы, по формуле (9):

Рассчитаем налоги, включаемые в затраты организации, приняв налоговую ставку 15%:

Рассчитаем общие затраты на создание программы по формуле (10):

Полученные данные занесем в таблицу 11.

Таблица 11 – Смета затрат на создание ПО

|  |  |  |
| --- | --- | --- |
| Наименование статьи затрат | Обозначение | Сумма, руб. |
| Зарплата программистов |  | 31092,7 |
| Зарплата других специалистов |  | 9269,2 |
| Итого зарплаты, израсходованной на создание математического обеспечения и написание программы |  | 40361,9 |
| Премия |  | 10090,5 |
| Выплаты по районным коэффициентам |  | 7567,9 |
| Страховые взносы с ФОТ |  | 17522,1 |
| Накладные расходы |  | 40361,9 |
| Итого затрат на создание ПО |  | 115904,3 |
| Затраты, связанные с работой компьютера при разработке ПО |  | 6164,4 |
| Прочие затраты, связанные с разработкой ПО |  | 34771,3 |
| Итого затрат на разработку ПО |  | 156840 |
| Налоги, включаемые в затраты на создание программы |  | 13327,1 |
| Затраты на оформление программного продукта |  | 23526 |
| Затраты на маркетинговые исследования |  | 15684 |
| **Всего затрат на создание ПО** |  | **209377,1** |

Рассчитав общие затраты на создание проекта, следует определить его проектные цены.

Величину прибыли рассчитывают по формуле (11):

, (11)

где – уровень рентабельности программного продукта (можно принять в размере 0,2-0,3).

Цена создания определяется по формуле (12):

, (12)

где – величина прибыли.

Розничную цену ПО можно определить по формуле (13):

, (13)

где НДС – налог на добавленную стоимость (на 1 января 2020 года – 20%);

– торговая наценка при реализации ПО через специализированные магазины (торговых посредников), ее можно принять в размере 10-20% от .

Произведём расчет проектируемых цен на ПО по формулам (11) – (13) с предположением о том, что будет разработан и продан только один экземпляр.

Рассчитаем величину прибыли по формуле (11):

Рассчитаем цену создания программы по формуле (12):

Рассчитаем розничную цену программы по формуле (13):

Результаты расчётов представлены в таблице 12.

Таблица 12 – Расчёт цены ПО

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Обозначение | Сумма, руб. |
| Затраты на создание ПО |  | 209377,1 |
| Прибыль |  | 41875,4 |
| Цена создания ПО |  | 251252,5 |
| Налог на добавленную стоимость |  | 50250,5 |
| Торговая наценка |  | 25125,3 |
| **Розничная цена** |  | **326628,3** |

Таким образом, в ходе выполнения данного подраздела была создана смета затрат на создание ПО, общая сумма затрат составила 209377,1 руб., розничная цена составила 326628,3 руб.

**4.2 Расчёт выручки и прибыли от реализации программного продукта**

После расчета затрат и цен, выполненного в пункте 4.1, нужно определить предполагаемую сумму выручки и прибыли от внедрения программы.

Валовая выручка от реализации ПО по рыночной цене (без учета торговой наценки) вычисляется по формуле (14):

, (14)

где – цена реализации разработанного ПО (без учета торговой наценки);

– количество копий ПО, которые предполагается реализовать.

Объем выручки от продажи программного обеспечения по цене создания рассчитывается по формуле (15):

. (15)

Затраты на одно копирование ПО следует рассчитывать по формуле (16):

, (16)

где – время одного копирования ПО, мин.;

– время подготовки (форматирования) носителя информации;

– розничная цена носителя информации, используемого под копию ПО;

– затраты на копирование или печать сопроводительной документации и приобретение упаковки для хранения этой документации и носителя информации (можно принять в размере 50-100% от минимальной зарплаты ).

Величину прибыли от продажи всех копий программного продукта можно определить по формуле (17):

, (17)

где – затраты на одно копирование ПО.

Балансовую прибыль, которую может получить организация, разрабатывающая и реализующая ПО, рассчитывают по формуле (18):

, (18)

где – прочие доходы (по ценным бумагам, от долевого участия в совместных проектах и др., можно принять в размере 3-4% от );

– прочие расходы (выплаты по экономическим санкциям и др., можно принять в размере 0,5-1% от ).

«Чистую» прибыль можно определить по формуле (19):

, (19)

где – налог на прибыль (20%).

Произведём расчеты по формулам (14) – (19) и результаты представим в таблице 13.

Рассчитаем валовую выручку от реализации ПО по рыночной цене по формуле (14):

Рассчитаем объем выручки от продажи ПО по цене создания по формуле (15):

Рассчитаем затраты на одно копирование ПО по формуле (16), приняв в качестве носителя информации USB-флеш-накопитель стоимостью 700 рублей:

Рассчитаем величину прибыли от продажи всех копий программы по формуле (17):

Рассчитаем балансовую прибыль, которую может получить организация, разрабатывающая и реализующая ПО, по формуле (18):

Рассчитываем «чистую» прибыль по формуле (19):

Полученные результаты представлены в таблице 13.

Таблица 13 – Итоговый расчёт формирования и использования выручки

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Обозначение | Сумма, руб. |
| Валовая выручка от реализации ПО по рыночной цене |  | 326628,3 |
| Налог на добавленную стоимость |  | 50250,5 |
| Выручка от продажи ПО по цене создания |  | 251252,5 |
| Общие затраты на создание одной копии ПО |  | 6774,3 |
| Прибыль от продажи ПО |  | 41875,4 |
| Доходы от внереализационных операций |  | 1256,3 |
| Расходы от внереализационных операций |  | 209,4 |
| Налог на прибыль |  | 7229,6 |
| Балансовая прибыль |  | 36148 |
| **«Чистая» прибыль** |  | **28918,4** |

В данном подразделе был выполнен расчет выручки и прибыли. Выручка составила 326628,3 руб., «чистая» прибыль составила 28918,4 руб.

**4.3 Расчёт затрат, связанных с покупкой, внедрением и использованием программного обеспечения**

При анализе эффективности приобретаемой компьютерной программы следует рассчитать следующие показатели:

1. Капитальные затраты на внедрение программного обеспечения;
2. Текущие затраты пользователя, связанные с применением программного обеспечения;
3. Экономию от использования программного обеспечения;
4. Срок окупаемости капитальных затрат.

Общее время эксплуатации компьютера в течение года рассчитывается по формуле (20):

, (20)

где – длительность смены (8 часов);

– число смен работы компьютера;

– среднее число рабочих дней в месяце (21 день);

– число месяцев в году;

– средний коэффициент использования компьютера в течение года (рекомендуется принять в пределах 0,7–0,9).

Величина затрачиваемого компьютерного времени на решение задач с помощью купленной программы определяется по формулам (21) – (22):

, (21)

, (22)

где – время решения задач с помощью приобретенного ПО в течение одного дня (недели, месяца);

Д – количество дней (недель, месяцев) использования купленного ПО в течение года;

– количество задач определенного вида, решаемых с помощью купленной программы в течение года;

– машинное время, затрачиваемое компьютером на решение одной задачи определенного вида.

Капитальные затраты на техническое оснащение рабочего места пользователя ПО рассчитываются по формуле (23):

, (23)

где – рыночная цена компьютера на момент покупки программы (примем 30000 руб.);

– цена дополнительного технического оснащения компьютера (можно принять в размере 30-100 % от , однако, поскольку дополнительное оборудование для использования ПО не понадобится, то примем = 0);

– коэффициент, учитывающий затраты на транспортировку и отладку компьютера и других технических средств (т.к. будет использоваться действующий компьютер, то = 0);

– коэффициент, учитывающий степень износа действующего компьютера, на котором будут решаться задачи с помощью купленного ПО (можно определить укрупненно, путем сопоставления фактического и проектного (обычно не более 5 лет) сроков службы ПЭВМ).

Капитальные вложения на создание рабочего места пользователя ПО (без учета износа) следует рассчитать по формуле (24):

, (24)

где S – размер площади, которую занимают компьютерный стол и специалист, работающий с помощью компьютера (4-5 м2);

– рыночная цена 1 м2 площади на момент покупки ПО (примем 20000 руб.);

– затраты на приобретение мебели (компьютерный стол, кресло для пользователя, стол для принтера и др., их можно принять 15-20 % от );

– время использования компьютера в течение года для решения всех задач с помощью приобретенной программы;

– общее время эксплуатации компьютера в течение года.

Капитальные затраты на приобретение и внедрение ПО определяются по формуле (25):

, (25)

где – затраты на покупку ПО (принимаются равными рыночной цене программы);

– капитальные вложения на создание рабочего места пользователя ПО;

– капитальные вложения на техническое оснащение рабочего места пользователя ПО;

– прочие капитальные вложения, связанные с внедрением ПО. В прочие затраты, связанные с внедрением ПО, включаются расходы на покупку новых версий программ, по изменению действующей документации, по обучению персонала пользованию новой программой, по адаптации и апробации программы в конкретных условиях работы у пользователя и др. Их можно определить укрупненно: принять в размере 10-20 % от .

Произведём расчеты по формулам (20) – (25) и результаты представим в таблице 14.

Рассчитаем общее время эксплуатации компьютера в течение года по формуле (20):

Рассчитаем величину затрачиваемого компьютерного времени на решение задач с помощью купленной программы по формуле (21), считая, что ПО используется 1 час каждый день в течение месяца:

Рассчитаем капитальные затраты на техническое оснащение рабочего места пользователя ПО по формуле (23):

Рассчитаем капитальные вложения на создание рабочего места по формуле (24):

Рассчитаем капитальные затраты на приобретение и внедрение ПО по формуле (25):

Полученные данные занесем в таблицу 14.

Таблица 14 – Капитальные затраты на покупку и внедрение ПО

|  |  |  |
| --- | --- | --- |
| Наименование затрат | Обозначение | Сумма, руб. |
| Затраты на покупку ПО |  | 326628,3 |
| Затраты на создание рабочего места |  | 15089,3 |
| Затраты на техническое оснащение рабочего места |  | 5357,1 |
| Прочие капитальные затраты |  | 32662,8 |
| **Итого** |  | **379737,5** |

Годовые текущие затраты пользователя, связанные с применением программы, рассчитываются по формуле (26):

, (26)

где – время занятости компьютера решением задач с помощью программы, ч.;

– стоимость одного часа эксплуатации компьютера определенной модели (без учета амортизационных отчислений от стоимости приобретенного ПО);

– рыночная цена купленного программного обеспечения;

– планируемый срок использования приобретенной программы (с учетом морального износа не более 5 лет).

Затраты на решение задач без применения программы определяются по формуле (27):

, (27)

где Р – число работников, участвующих в решении задач ручным способом;

– время участия каждого работника в решении задач ручным способом в течение года, в часах.

Годовую экономию на текущих расходах, которую может получить фирма от применения программного обеспечения, определяют по формуле (28):

. (28)

Срок окупаемости капитальных затрат на покупку и внедрение программы рассчитывается по формуле (29):

. (29)

Произведём расчеты по формулам (26) – (29).

Рассчитаем текущие годовые затраты, связанные с применением программы, по формуле (26):

Рассчитаем затраты на решение задач без применения компьютерной программы по формуле (27):

Рассчитаем годовую экономию на текущих расходах, которую может получить фирма от применения программного обеспечения, по формуле (28):

Рассчитаем срок окупаемости капитальных затрат на покупку и внедрение компьютерной программы по формуле (29):

Финансово-экономические показатели создания и использования ПО представлены в таблице 15.

Таблица 15 – Финансово-экономические показатели создания и использования ПО

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Единица измерения | Значение показателя |
| **1 Показатели фирмы-разработчика ПО** | | |
| 1.1 Число специалистов, участвующих в разработке компьютерной программы | человек | 2 |
| 1.2 Время создания ПО | час | 345 |
| 1.3 Число копий ПО | штук | 1 |
| 1.4 Затраты на создание ПО | рублей | 209377,1 |

Продолжение таблицы 15

|  |  |  |
| --- | --- | --- |
| 1.5 Розничная цена одной копии | рублей | 326628,3 |
| 1.6 Уровень рентабельности | % | 20 |
| 1.7 Балансовая прибыль от продажи ПО | рублей | 36148 |
| 1.8 «Чистая» прибыль | рублей | 28918,4 |
| **2 Показатели фирмы-покупателя ПО** | | |
| 2.1 Капитальные затраты на покупку и внедрение ПО | рублей | 379737,5 |
| 2.2 Годовые текущие расходы, связанные с использованием ПО | рублей | 144181,3 |
| 2.3 Годовая экономия от применения ПО | рублей | 120014,2 |
| 2.4 Расчетный срок окупаемости капитальных затрат | год | 3,2 |

В ходе выполнения данного подраздела был выполнен расчёт затрат, связанных с покупкой, внедрением и использованием программного обеспечения. Капитальные затраты на покупку и внедрение ПО составили 379737,5 руб.

Покупка и применение программы будет экономически выгодна, т.к. расчетный срок окупаемости не превышает 5 лет, что, в свою очередь, является рекомендуемым сроком окупаемости в условиях рынка.

**4.4 Выводы к разделу 4**

В четвертом разделе было произведено экономическое обоснование разработки конструктора интернет-магазинов. В ходе выполнения данного раздела был произведён расчёт затрат на разработку программного обеспечения, вычислены выручка и прибыль от реализации ПО, а также произведён расчёт затрат на покупку и использование ПО.

Затраты на создание ПО составили 209377,1 руб., розничная цена составила 326628,3 руб., «чистая прибыль» составила 28918,4 руб.

Годовая экономия от внедрения ПО составит 120014,2 руб., при этом срок окупаемости капитальных затрат равен 3,2 года, что не превышает 5 лет. Таким образом, можно сделать вывод, что разработка данного проекта положительно скажется на финансовом состоянии фирмы-покупателя.