Государственное Учреждение Образования Республики Беларусь

«Институт Информационных Технологий Белорусского Государственного Университета Информатики и Радиоэлектроники»

Отчет по лабораторной работе № 1

По курсу ПИРИС

**«Разработка описания и анализ информационной системы «Бюро по трудоустройству»**

Выполнил студент:

Сорока А.А. гр. 481064

Минск 2017

**Цель работы**

Описать и проанализировать информационную систему, распределить роли в группе разработчиков.

Основные цели:

- создание единой базы всех вакансий и резюме в организации;

- структурирование вакансий и резюме согласно утвержденной номенклатуре;

- сокращение риска утери базы данных;

- простой и эффективный поиск вакансии;

- простой способ размещения вакансии и резюме;

- контроль исполнения.

**Введение**

Лабораторная работа направлена на ознакомление с процессом описания информационной системы и получение навыков по использованию основных методов анализа ИС.

На сегодняшний день создание информационной системы, которая упрощает доступ и позволяет предоставлять требуемую информацию конечному пользователю, является приоритетной, по сравнению с предыдущими этапами развития в бизнесе.

Отсутствие необходимости вручную отслеживать перемещение бумажных документов, контролировать порядок передачи конфиденциальной информации существенным образом снижает времязатраты и трудозатраты.

*Ограничения:*

Временные: 3 месяца

Бюджет: 20000 $

*Рекомендации к разрабатываемой системе:*

- разделение прав доступа для учетных записей пользователей;

- вход в систему после проверки логина и пароля;

- ведение картотеки резюме и вакансий;

- управление;

- резервное копирование и восстановление базы данных.

*Требования к программному обеспечению:*

- регистрация вакансий и резюме;

- электронный архив;

- автоматизация процесса связи организаций и соискателей;

- библиотека регламентов управленческих процедур.

**Описание информационной системы (ПО)**

Основная задача платформы заключается в повышении уровня абстракции при разработке и использовании прикладных решений. Это позволяет перейти от технических и низкоуровневых понятий к более содержательным и высокоуровневым. Позволяет приблизить эти понятия к языку пользователей и специалистов в предметной области. В конечном итоге это значительно ускоряет и унифицирует разработку прикладного решения и его сопровождение[1].

Одновременно с этим платформа решает и традиционные задачи, связанные с производительностью, эргономикой, функциональностью и т.д.

*Платформа и прикладные решения.*

В системе «Microsoft SharePoint»[2] существует четкое разделение на платформу и прикладное решение. Платформа представляет собой framework, в котором функционирует прикладное решение:

Платформа служит фундаментом для построения прикладных решений,

Платформа является средой их исполнения,

Платформа содержит инструментарий, необходимый для разработки, администрирования и поддержки прикладных решений.

При этом прикладное решение является самостоятельной сущностью и может выступать в качестве отдельного программного продукта. Но полностью опирается на технологии платформы.

**Анализ осуществимости**

Разрабатываемая система отвечает общим и бизнес-целям организации-заказчика и организации-разработчика. Продукт «Microsoft SharePoint» обладает всеми необходимыми инструментами и позволяет выполнить данную задачу без необходимости увеличение стоимости конечного программного решения[3].

Если по результатам эксплуатации разработанной конфигурации будет выяснено, что приходится вести часть информации в EXCEL, не формируются отчеты в требуемом виде, «программа не подходит» и др., рекомендуется еще раз проанализировать неучтенные моменты и внести корректировки в бизнес-процессы организации, в настройки программы (возможно, выполнить доработки программы)[4].

**Категории рисков**

|  |  |
| --- | --- |
| **Категория рисков** | **Примеры рисков** |
| Технологические риски | База данных, которая используется в программной системе, не обеспечивает обработку ожидаемого объема транзакций. |
| Риски, связанные с персоналом | Ведущий разработчик заболел в самое критическое время.  Работник или работники уволились  Работник ушёл в декретный отпуск |
| Организационные риски | Финансовые затруднения в организации привели к уменьшению бюджета проекта. |
| Риски, связанные с системными требованиями | Изменения требований приводят к значительным повторным работам по проектированию системы.  Первоначальная нечеткая формулировка пользовательских требований привела к значительным изменениям системных требований, проявившихся на поздних стадиях разработки проекта. |
| Риски оценивания | Недооценки времени выполнения проекта.  Скорость выявления дефектов в системе ниже ранее запланированной.  Размер системы значительно превышает первоначально рассчитанный. |

Существует три категории стратегий управления рисками.

1. *Стратегии предотвращения рисков.* Согласно этим стратегиям следует проводить меро­приятия, снижающие вероятность проявления рисков. Примером может служить стратегия исключения потенциально дефектных компонентов, описанная в таблице рисков[5].
2. *Минимизационные стратегии.* Направлены на уменьшение возможного ущерба от рисков. Примером служит стратегия уменьшения ущерба от болезни членов команды разработчиков (см. в таблице рисков).
3. *Планирование "аварийных" ситуаций.* Согласно этим стратегиям необходимо иметь план мероприятий, которые следует выполнить в случае проявления рисковой си­туации. В таблице рисков это стратегия поведения при возникновении финансовых про­блем у организации-разработчика.

**Анализ рисков**

|  |  |  |
| --- | --- | --- |
| **Риск** | **Вероятность** | **Степень ущерба** |
| База данных, которая используется в программной системе, не обеспечивает обработку ожидаемого объема транзакций. | Низкая | Катастрофическая |
| Ведущий разработчик заболел в самое критическое время | Высокая | Катастрофическая |
| Работник или работники уволились | Средняя | Катастрофическая |
| Работник ушёл в декретный отпуск | Средняя | Катастрофическая |
| Финансовые затруднения в организации привели к уменьше­нию бюджета проекта | Низкая | Серьезная |
| Изменения требований приводят к значительным повторным работам по проектированию системы | Средняя | Серьезная |
| Первоначальная нечеткая формулировка пользовательских требований привела к значительным изменениям системных требований, проявившихся на поздних стадиях разработки проекта | Средняя | Серьезная |
| Недооценки времени выполнения проекта. | Средняя | Серьезная |
| Скорость выявления дефектов в системе ниже ранее запланированной. | Низкая | Терпимая |
| Размер системы значительно превышает первоначально рассчитанный | Низкая | Терпимая |

**Роли участников группы разработки ПО.**

Системный аналитик (System analyst) — специалист по решению сложных организационно-технических проблем, имеющих междисциплинарную природу, использующий принципы общей теории систем и методы системного анализа.

В узком смысле в сфере информационных технологий данный термин используется для обозначения профессиональной роли и профессии, ответственной за анализ интересов заинтересованных лиц создаваемой IT-системы на предмет возможности их удовлетворения её техническими свойствами. Также его называют «постановщик задач».

Основным продуктом такого системного аналитика являются организационно-технические решения, оформляемые как техническое задание на систему, техническое задание на программное обеспечение[6].

Деплоймент-инженер (Build engineer / Application Support Analyst) — специалист-аналитик, планирующий и организующий работу программистов с целью упрощения и оптимизации, также реализует программные решения.

Его должностные обязанности могут включать в себя выбор методологии и среды разработки; создание или выбор фреймворков; принятие решений о том, когда следует воспользоваться готовым программным кодом вместо самостоятельной разработки; написание спецификаций и прочей документации для будущего программного продукта и планирование сроков разработки. Специалист должен быть знаком как с предметной областью, так и с процессом разработки.

SharePoint разработчик – аналитик ERP систем (ERP Developer) — в обязанности программиста входит настройка и поддержка программного продукта «Microsoft SharePoint» - написание или доработка новых конфигураций, создание и доработка новых форм, отчетов, обработок[7].

Основные обязанности программиста:

- автоматизация деятельности компании;

- настройка, сопровождение, администрирование типовых и нетиповых конфигураций;

- разработка нетиповых конфигураций под задачи компании;

- составление инструкций, оформление технической документации;

- поддержка и консультация пользователей.

**Программно-аппаратные средства, используемые при выполнении работы.**

- платформа разработки: «Microsoft SharePoint» версия 2016;

- среда разработки: «Microsoft Visual Studio» версия 2016;

- платформа БД : Microsoft SQL server 2016.

MS SQL Server – является продуктом, ориентированным на операционные системы семейства Windows. Системные требования MS SQL Server 2016 по современным меркам не велики и определяются скорее нагрузкой на базу данных, при этом сервер обладает высокой производительностью и легкостью администрирования[8].

Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL[9] является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

Преимуществами MS SQL Server являются:

- простота проектирования;

- простота управления базами данных;

- наглядность;

- безопасность.

Основным инструментом MS SQL Server 2008 является среда SQL Server Management Studio - утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

ПО для написания, документирования и регистрации работ: Microsoft Word, Microsoft Excel, Microsoft Visio.

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

Основные преимущества языка C#:

- простота, он относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java;

- объектно-ориентирован, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы;

- поддержка Visual C# в Visual Studio реализована в виде полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и к другим полезным, хорошо спроектированным классам, что существенно ускоряет цикл разработки.

- простота подключения к базам данных, представления, изменения данных и структур таблиц с помощью встроенных библиотек;

- кроссплатформенность, благодаря проектам Mono и Portable.NET, позволяющие запускать программы .NET, в том числе и написанные на C# на некоторых других операционных системах.

**ЗАКЛЮЧЕНИЕ**

Развитие информационных технологий учета за последние годы идет все возрастающими темпами. Любой пользователь работает на компьютере и даже не задумывается о том, сколько процессов, которыми он каждодневно пользуется, автоматизировано.

Модернизация и смена поколений вычислительной техники, переход на новые операционные системы, обновление версий прикладных программ и т.п. является характерной особенностью современной компьютеризации. в настоящее время повсеместно внедряются компьютерные сети (локальные, региональные, глобальные), благодаря которым обеспечен удаленный доступ к общесетевым ресурсам (базам данных, компьютерам, принтерам, факс-модемам и др. )

Разработка программного обеспечения представляет собой сложный процесс, в котором принимает участие вся команда (руководитель, тестировщик, разработчик, аналитик). Каждый член команды должен выполнять свою часть работы, при этом учитывая цели проекта, требования к разрабатываемому программному продукту. Немаловажную роль играет планирование бюджета и времени. Это позволяет максимально точно определить возможность разработки, используемые технологии и т.д.

Важно грамотно распределить роли в команде, обеспечить эффективное общение внутри коллектива, вести мониторинг текущей ситуации и отслеживать, успевает ли команда завершить проект вовремя. При выполнении всех условий программный продукт будет соответствовать требованиям и будет готов к использованию в установленный заказчиком срок при изначально заданном бюджете.

**Список используемой литературы:**

1. Соммервиль Иан. Инженерия программного обеспечения, 6-е издание: Пер. с англ. – М.: Издательский дом “Вильямс”, 2002. – 624 с.
2. Кирьянов Д. Microsoft SharePoint Foundation 2010 и SharePoint Online 365: С-П. Издательство «Питер», 2011. – 524 с.
3. Константайн Л., Локвуд Л. Разработка программного обеспечения. – СПб.:Питер, 2004. – 592 с.
4. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя.: Издательство «Питер», 2003. – 432 с.
5. Конвертация данных. Методика работы и примеры. О. Кузнецова. 2008г.
6. Л. Басс, П. Клементс, Р. Кацман Архитектура программного обеспечения на практике (NFR)
7. Рамбо Дж. Microsoft SharePoint 2016. Полное руководство: С-П. Издательство «Питер», 2016. – 863 с.
8. Д. Петкович Microsoft SQL Server 2012. Руководство для начинающих: С-П.: Издательство «БХВ-Петербург», 2013. – 816 с.
9. И. Бен-Ган Microsoft SQL Server 2012. Основы T-SQL: Издательство «Питер», 2015. – 400 с.
10. Г.Шилдт C# 4.0. Полное руководство: Издательство «Вильямс», 2015. – 1000 с.