

СОДЕРЖАНИЕ

Введение	7
1 Жизненный цикл ИТ-проектов.....	8
1.1 Планирование и распределение рабочего времени	2
1.2 Этап аналитики.....	3
1.3 Этап разработки	4
1.4 Этап тестирования.....	14
1.5 Этап исправления дефектов.....	7
Список использованных источников.....	17
Приложение А Сущности для работы с базой данных (модели).....	18
Приложение Б Вид страницы «Лента новостей»	20
Отзыв о прохождении практики	21

1 Жизненный цикл ИТ-проектов

Компания «000» специализируется на разработке, внедрении и поддержке собственного программного обеспечения. Для создания программного продукта организуется ИТ-проект, состоящий из нескольких фаз [2]. Фаза проекта – это набор логически взаимосвязанных работ проекта, в процессе завершения которых достигается один из основных результатов проекта, то есть создание работоспособного, отказоустойчивого программного продукта. Жизненный цикл проекта – это полный набор последовательных фаз проекта. Жизненный цикл ИТ-проектов в компании «000» состоит из следующих основных фаз:

- инициация;
- аналитика;
- разработка;
- тестирование.

На фазе инициации происходит первый контакт и налаживание связей с заказчиком.

На фазе аналитики происходит общение с заказчиком, выяснение основных требований и пожеланий к программному продукту. Основная цель данной фазы – создание документации к проекту, по которым будет вестись разработка.

В фазе разработке осуществляется написание программного кода на основе спецификаций, созданных на этапе аналитики.

В фазе тестирования выполняется тестировка программных модулей с целью выявления в них различных дефектов и уязвимостей. При обнаружении таковых, продукт отправляется на доработку, то есть переходит обратно в фазу разработки.

Стоит отметить, что при разработке программных продуктов в данной компании используется итеративный подход – выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл: Планирование – Реализация–Проверка–Оценка (англ. Plan–Do–Check–Act cycle). В конце каждой итерации, длительность которой две недели, проектная команда должна достичь запланированных на данную итерацию целей, создать или доработать проектные артефакты и получить промежуточную, но функциональную версию конечного продукта [3]. Это означает, что проектные команды, задействованные в отдельных фазах проекта, работают параллельно, постоянно тесно взаимодействуя друг с другом.

1.1 Планирование и распределение рабочего времени

Эффективность работы организации во многом определяется эффективностью повседневной работы её персонала. Решающее значение для устранения недостатков в

практике повседневной деятельности имеет рациональное использование рабочего времени [4].

Важным шагом при создании эффективной системы управления собственным временем является определение целей деятельности как на короткий, так и на длительный период. Определение целей есть лишь начальный этап. Необходимо преобразовать цель в план действий по ее достижению. Необходимо определить степени важности и решить, какие цели и задачи важнее всего, а какие могут подождать. Успешное решение задачи выделения степеней важности оказывает сильное влияние на рост эффективности работы в целом.

В компании «000» для данных целей используется Redmine – открытое серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок) [5]. Каждый сотрудник компании имеет личный профиль на данном сервисе, в котором он имеет возможность создавать индивидуальные задачи, выделять в них планируемое, а также реальное время выполнения, определять важность задачи.

Redmine идеально подходит и для командного взаимодействия. Некоторые общие задачи могут модифицировать все члены рабочей группы, и поэтому существует возможность, например, дать понять, что конкретная задача нуждается в немедленном выполнении. Немало важной особенностью является возможность руководителю проекта самому создавать и назначать задачи подчиненным, что обеспечивает более быстрое и гибкое управление деятельностью рабочей группы.

1.2 Этап аналитики

Системный аналитик – термин, который в узком смысле в сфере информационных технологий термин используется для обозначения профессиональной роли и профессии, ответственной за анализ интересов заинтересованных лиц (заказчика) создаваемой ИТ-системы на предмет возможности их удовлетворения её техническими свойствами. Также его называют «постановщик задач» [6].

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

Техническое задание, как термин в области информационных технологий – это юридически значимый документ, содержащий исчерпывающую информацию, необходимую для постановки задач исполнителям на разработку, внедрение или интеграцию программного продукта, информационной системы, сайта, портала либо прочего ИТ-сервиса [7].

При прохождении практики разработать технические спецификации к проекту было основным заданием на этапе аналитики. Спецификация требований ПО – это документ, который содержит полное и чёткое описание разрабатываемого продукта. В компании «000» придерживаются философии, что инструменты для непростого процесса написания спецификаций должны быть простыми и легкодоступными как для понимания, так и для командной работы. В качестве таких инструментов выступает внутренний вики-сайт компании на движке MediaWiki со множеством включенных расширений для удобной работы.

При разработке веб-приложения, спецификации должны были включать в себя следующие пункты:

- общие требования к приложению;
- прототипы (макеты) экранных форм приложения;
- описание функциональных требований к элементам формы и к форме в целом.

Создание прототипов экранных форм приложения являлось наиболее творческой и времязатратной частью этапа аналитики. Для данной цели использовался Balsamiq Mockups. Balsamiq Mockups – это сервис для построения макетов пользовательских интерфейсов для упрощения разработки веб-сайтов, мобильных и десктопных программ [8]. Balsamiq Mockups позволяет рисовать окна приложений, простые фигуры, текстовые блоки, таблицы, картинки, кнопки, стрелки и др. При редактировании элемента появляется всплывающее окно настройки. После создания макета можно сохранить его в форматах PNG или PDF, или экспортировать в виде XML-файла. Одна из созданных автором на этапе аналитики экранных форм веб-приложения – форма профиля пользователя представлена на рисунке 1. Часть функциональных требований к элементам формы профиля приведены на рисунке 2.

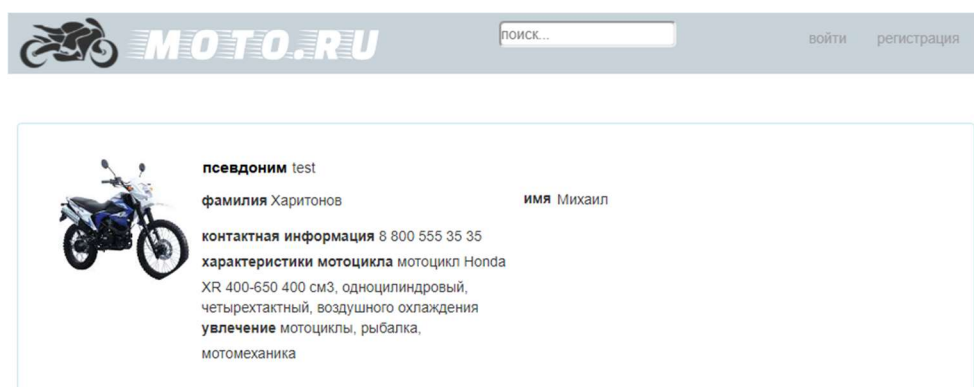


Рисунок 1 – Прототип экранной формы профиля пользователя

1.3 Этап разработки

В процессе разработки программного обеспечения так или иначе приходится вносить в него исправления (изменения), в которых, с большой вероятностью, могут

содержаться ошибки. Если проект небольшой, то, обычно, делается его резервная копия. Однако если сопровождается большой проект, состоящий из сотен или даже тысяч файлов, то копировать каждый раз весь проект и вручную описывать версию долго и трудоемко. Кроме того, такой подход имеет ещё ряд недостатков. Во-первых, файлы всех версий проекта будут занимать много места на жестком диске. Во-вторых, легко запутаться в версиях и использовать, вместо сохраненной правильной версии, промежуточную недоделанную версию с множеством ошибок. В-третьих, возможны ситуации потери единственно верных копий с последними наработками в результате случайного удаления файла с жесткого диска компьютера.

Элементы формы "Профиль"



На форме "Профиль" отображаются следующие элементы:

- **Заголовок страницы.** Требования к Заголовку страницы описаны в разделе Элементы формы "Лента новостей";
- **Кнопка [Добавить пост].** По нажатию отображается форма "Добавление новой записи", аналогичную таковой на главной странице. Кнопка видна только в своём профиле.
- **Форма "Информация о пользователе".** Включает в себя:
 - Для гостя:
 - **Текст "Информация".** Содержит псевдоним, ФИО, контактную информацию, ТТХ мотоцикла, увлечения этого пользователя.
 - **Аватарка пользователя.** Отображает аватару пользователя.
 - Для владельца профиля:
 - **Текст "Псевдоним".** Содержит псевдоним своего профиля.
 - **Аватарка пользователя.** По нажатию отображается стандартное диалоговое окно для выбора файла с локального диска пользователя для загрузки на сайт;
 - **Текстовое поле "Фамилия".** Однострочное поле ввода. Допустимое количество символов для ввода 31. Максимальное количество отображаемых символов 20;
 - **Текстовое поле "Имя".** Однострочное поле ввода. Допустимое количество символов для ввода 31. Максимальное количество отображаемых символов 20;
 - **Текстовое поле "Отчество".** Однострочное поле ввода. Допустимое количество символов для ввода 31. Максимальное количество отображаемых символов 20;
 - **Текстовое поле "Псевдоним".** Однострочное поле ввода. Допустимое количество символов для ввода 31. Максимальное количество отображаемых символов 20. Обязательно для заполнения. По нажатию отображает всплывающее окно с требованиями к псевдониму. Псевдоним может содержать от 3 до 20 символов. Латиница, кириллица, цифры и подчёркивания;
 - **Текстовое поле "Пароль".** Однострочное поле ввода. Допустимое количество символов для ввода 31. Максимальное количество отображаемых символов 20. Тип текста "password" - при вводе отображаются кружки или звёздочки вместо вводимых символов. Обязательно для заполнения, если отображено.
 - **Текстовое поле "Подтвердите пароль".** Однострочное поле ввода. Допустимое количество символов для ввода 31. Максимальное количество отображаемых символов 20. Тип текста "password" - при вводе отображаются кружки или звёздочки вместо вводимых символов. Обязательно для заполнения, если отображено.
 - **Текстовое поле "Контактная информация".** Допустимое количество символов для ввода 255. Максимальное количество отображаемых символов 255. Количество отображаемых строк 4.
 - **Текстовое поле "ТТХ мотоцикла".** Допустимое количество символов для ввода 255. Максимальное количество отображаемых символов 255. Количество отображаемых строк 4.
 - **Текстовое поле "Увлечения".** Допустимое количество символов для ввода 255. Максимальное количество отображаемых символов 255. Количество отображаемых строк 4.
 - **Кнопка [Сохранить изменения].** Сохраняет введенные данные. При ошибке ведёт себя аналогично кнопке "Зарегистрироваться" в форме "Регистрация". Активна при изменении любого текстового поля на форме "Информация о пользователе" или по нажатию на кнопку [Изменить пароль].
- **Лента новостей.** Совпадает по требованиям с Лентой новостей из формы "Лента новостей", однако включает в себя только записи этого пользователя, если таковые имеются. Иначе отображается форма с заголовком "Здесь пока ничего нет..." и надписью "Вы пока не опубликовали ни одной записи. Вы можете поделиться рассказами о своём железном коне нажав на кнопку "Добавить пост" в правом верхнем углу."

Рисунок 2 – Функциональные требования к элементам формы «Профиль»

Ситуация значительно усложняется, если проект ведут несколько человек, иногда территориально удаленных друг от друга на сотни и тысячи километров. В этом случае у каждого члена команды разработки будут образовываться свои архивы версий, и командное сопровождение разработанного программного обеспечения будет крайне трудноосуществимо.

Все эти неудобства послужили толчком к созданию множества программ для контроля версий (Mercurial, Git, CVS, Subversion, Bazaar, Monotone, Aegis и др.), позволяющих решать вышеперечисленные задачи.

При разработке веб-приложения «Блог любителей мотоспорта» применялась система контроля версий Mercurial. Для более простой и удобной работы с этой системой использовалась программа TortoiseHg, представляющая собой графическую оболочку для Mercurial. Пример интерфейса TortoiseHg приведён на рисунке 3. Для хранения созданных репозиторий проекта использовался общедоступный веб-сервис Bitbucket, предназначенный для хостинга и совместной разработки проектов и основанный на системе контроля версий Mercurial и Git [9]. При непосредственном создании программного кода приложения использовался программный продукт под названием Microsoft Visual Studio. Microsoft Visual Studio – это линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

Рисунок 3 – Скриншот интерфейса TortoiseHg

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 Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight [10].

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 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, добавление новых наборов

инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения.

Также во время проектирования приложения использовались принципы объектно-ориентированного программирования и архитектурный паттерн Model-View-Controller (MVC).

Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента [11]:

- модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние;
- представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели;
- контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

База данных разрабатываемого веб-приложения «Блог любителей мотоспорта» состоит из двух таблиц:

- Пользователи (Users). Содержит поля: Id (идентификатор пользователя в базе данных), Nickname (никнэйм), Password (пароль авторизации), Email (электронная почта), PhotoUrl (URL-ссылка на изображение профиля пользователя – аватарку), FirstName (имя), LastName (фамилия), ContactInfo (контактная информация), BikeSpecifications (ТТХ мотоцикла), Hobbies (увлечения);
- Посты (Posts). Содержит поля: Id (идентификатор публикации в базе данных), Date (дата публикации поста), Text (текст поста), PhotoUrl (URL-ссылка на изображение поста), Author (ссылка на автора).

Для каждой из двух таблиц была описана объектная модель, или сущность. Это позволило с легкостью оперировать базой данных в рамках ASP.NET MVC и Entity Framework. Код описанных классов-сущностей представлен в приложении А.

Также были созданы представления (View), являющиеся HTML-разметкой для каждой страницы блога, соответствующие требованиям спецификации, реализующие интерфейс приложения.

Для каждой View был создан контроллер, обрабатывающий запросы пользователя и обеспечивающий изменение моделей, вид View и осуществляющий переходы между страницами блога.

1.4 Этап тестирования

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом [12].

Данный этап включал в себя ознакомление с различными видами тестирования:

- функциональное тестирование;
- нефункциональное тестирование;
- тестирование безопасности.

Нефункциональное тестирование подразделяется на:

- тестирование пользовательского интерфейса на соответствие спецификации;
- тестирование пользовательского интерфейса на удобство использования;
- инсталляционное тестирование – выявление дефектов программного обеспечения, влияющих на протекание стадии инсталляции (установки) и деинсталляции (удаления) приложения;
- конфигурационное тестирование – проверка совместимости с различными версиями браузера;
- графическое тестирование;
- тестирование совместимости – проверка совместимости с различными устройствами;
- тестирование надёжности и восстановления – возможность восстановить несохраненные данные;
- локализованное тестирование – проверка отображения иностранных языков, различных шрифтов;
- тестирование стабильности – проверка работоспособности приложения при длительном тестировании с ожидаемым уровнем нагрузки.

На практике осуществлялось функционально тестирование, которое включало в себя: написание «Check-листа» – документации, в которой описываются проверки, используемые для тестирования функционала приложения, описанного в спецификации и непосредственное тестирование веб-приложения, разработанного другим практикантом на основе «Check-листа», а также тестирование соответствия внешнего вида приложения исходной спецификации. На рисунке 4 представлен скриншот «Check-листа».

1.5 Этап исправления дефектов

Данный этап заключался в исправлении всех выявленных тестировщиком дефектов в разработанном автором приложении путём исправления ненадёжных и содержащих ошибки участков кода.

	Ссылки на спецификации						
2							
3	Наименование спецификации		Ссылка				
4	Web приложение для любителей байков						
5							
6	Модуль	Название группы тестов	Название теста	Номер теста	Чек-лист Шаги теста / Проверки	Ожидаемый результат	Комментарии
7	Лента новостей	Заголовок	Логотип	1	Проверить отображение логотипа Проверить переход к ленте по нажатию на логотип		
8			Поле поиска	2	Проверит: - ввод текста/цифр/спец символов - ввод минимум/максимум/обычное кол-во символов - отображение минимум/максимум символов - фильтрацию ленты по введённому значению по кнопке [Поиск] - фильтрацию ленты по введённому значению по клавише [Enter] - очистку поля по кнопке/клавише - уязвимости		
9			Кнопки	3	Проверить, что кнопки отображаются в соответствии со спецификацией для: - авторизованного пользователя - неавторизованного пользователя Проверить, что действие, реагирующее на нажатие элементов, соответствует спецификации. Нажать каждую кнопку: - один раз - несколько раз быстро - с помощью табличной/горячих кнопок.		
10			Аватар/псевдоним пользователя	4	Отображение аватар/псевдонима соответственно пользователю. Переход к профилю по нажатию на аватар/псевдоним		
11			Аватар/псевдоним автора	5	Отображение аватар/псевдонима соответственно автору. Переход к профилю по нажатию на аватар/псевдоним		
12			Дата публикации	6	Проверить отображение даты, если: - прошло меньше суток со времени публикации поста - прошло больше суток со времени публикации поста		
13			Тайтл	7	Проверить отображение: - короткого текста - очень длинного текста - очень длинного слова - цифр/спец символы		

По завершению этапа исправления ошибок было получено работающее веб-приложение, отвечающее заявленным в спецификации функциональным требованиям.

На странице «Профиль» авторизованный пользователь имеет возможность:

- На странице «Авторизации» любой пользователь имеет возможность:

- | № | Тема | Описание | Статус | Приоритет | Назначено | Категория | Серьезность | явлено в сборе | иснено в сборе | Вложения | Комментарий |
|---|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------|-------------|-------------------------|-------------|----------------|----------------|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 | Не работает редактирование в своего профиля | Предусловие: Пройти авторизацию. Зайти на страницу своего профиля.

1. Открыть форму авторизации
2. Провести действия в предусловии
3. Нажать кнопку "Сохранить"

Результат: Необработанное исключение при выполнении текущего веб-запроса
Ожидаемый результат: Изменения сохраняются | Закрыта | НЕМЕДЛЕННО | Разработчик | Профиль | Блокирующий | 10.07.13.37 | | | Сведения об исключении: System.InvalidOperationException: He удалось най... представление "UpdateProfile" или его образец, либо ни один обработчик, представленный не поддерживает места поиска. |
| 3 | Не выдается сообщение о том, что необходимо добавить картинку | Предусловие: Пройти авторизацию. Зайти на страницу своего профиля. Попыаться добавить пост без картинки

1. Открыть форму авторизации
2. Провести действия в предусловии
3. Нажать кнопку "Сохранить"

Результат: Необработанное исключение при выполнении текущего веб-запроса
Ожидаемый результат: Сообщение о том, что не добавлена картинка | Закрыта | НЕМЕДЛЕННО | Разработчик | Профиль | Блокирующий | 10.07.13.40 | | | Сведения об исключении: System.InvalidOperationException: Элемент модели, переданный в словарь, имеет тип "RealBlog.ModeIsPost", но для этого словаря требуется элемент модели типа "RealBlog.ModeIsUser". |
| 4 | Отображение страниц не соответствует спецификации | | Закрыта | Желательный | Разработчик | Авторизация/Регистрация | Критичный | 10.07.14.45 | | https://imgur.com/8QmudLpn | Смотря спецификацию |

9

На форме «Регистрация» любой пользователь имеет возможность создать новый аккаунт.


На форме «Лента новостей» авторизованный пользователь, дополнительно имеет возможность:

- просматривать публикации пользователей блога;
- добавлять новые публикации;
- перейти на страницу своего профиля;
- перейти в профиль автора любой публикации.

В приложении Б приведён вид страницы «Лента новостей».


ПРИЛОЖЕНИЕ Б

Вид страницы «Лента новостей»

 **MOTO.RU**

ПОИСК...

test



лента новостей

добавление нового поста

введите текст

добавьте изображение

Выберите файл

Файл не выбран

добавить

test 14.07.2019 в 08:37

Мотоцикл (фр. motorcycle, от лат. mōtor — приводящий в движение и греч. κύκλος — круг, колесо) — двухколёсное (либо трёхколёсное) транспортное средство с механическим двигателем (двигатель внутреннего сгорания, электрический, пневматический) главными отличительными чертами которого являются: вертикальная посадка водителя (мотоциклиста), наличие боковых ножных упоров (площадок, подножек), прямое (безредукторное) управление передним поворотным колесом.




Рисунок 7 – Вид страницы «Лента новостей»