

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра системного программирования

Онлайн-анкета для юзабилити-тестирования

Курсовая работа

Выполнила: Соковикова Н.А.
Научный руководитель: Литвинов Ю.В.

Санкт-Петербург 2013

Оглавление

Введение.....	3
Реализация.....	6
Форма для опроса.....	6
Форма для просмотра результатов.....	9
Апробация WebSharper.....	11
Работа с данными.....	12
Тестирование.....	13
Заключение.....	16
Список литературы.....	17

Введение

QReal:Robots - среда визуального программирования роботов, разрабатываемая на основе metaCASE-системы QReal, предназначенной для создания специализированных сред визуального программирования. Она задумывалась как среда для обучения основам программирования и кибернетики для учеников школ и студентов вузов. QReal:Robots имеет ряд преимуществ перед самыми распространенными на данное время визуальными средами программирования NXT-G и Robolab. Однако это новый продукт, поэтому для привлечения пользователей необходимо не только предоставить им новую функциональность, но и удобный и интуитивно понятный интерфейс. То есть, следует уделить внимание юзабилити приложения.

Юзабилити - это эргономическая характеристика того, насколько продукт может быть эффективно, экономично и с удовольствием использован определенными пользователями для достижения поставленных целей в заданном контексте использования. Согласно определению, качество продукта связывается не только с его внутренними характеристиками, но также с особенностями использования продукта и субъективным отношением к нему человека. Исследование и улучшение юзабилити увеличивает шанс программы на успех.

Целью моей прошлой курсовой работы являлось нахождение приемлемого метода оценки интерфейса. Существуют два принципиально разных подхода к созданию удобного интерфейса. При первом эксперты по юзабилити подключаются в самом начале создания приложения и вносят свои коррективы в проект в процессе его разработки. При другом ведется работа с уже существующим приложением, которое уже можно тестировать на предполагаемых пользователях. Для среды визуального программирования QReal:Robots подходил только второй подход. При его применении существует множество методов оценки, например, экспертная оценка интерфейса, сбор различных статистик (наблюдение за зрачками пользователя, длина траектории мыши и количество нажатий на кнопки при выполнении базовых задач, время их выполнения и пр.), тестирование при непосредственном участии пользователей. В данном случае, по совету специалиста, был

выбран следующий способ: пользователю предлагалось решить несложную задачу, рассчитанную на понимание интерфейса, а затем ответить на вопросы опросника SUS (System Usability Scale) [2]. В тестировании должны принимать участие наиболее вероятные будущие пользователи программы. Поэтому было необходимо тщательно выбирать участников, опрашивать каждого из них индивидуально, тестирование должен был проводить человек.

Согласно многим методикам проведения юзабилити-исследований, участие специалиста в подобных случаях весьма полезно, так как, наблюдая за участниками, он может сделать полезные выводы и предложить способы улучшения юзабилити. В этом же тестировании в качестве опрашивающего выступал человек, не имеющий никакого опыта в оценке интерфейса. Проведение исследования непрофессионалом может привести к искажению результатов. В прошлом году использовались бумажные опросники, результаты которых могут быть легко утеряны, их сложно упорядочить и подробные результаты сложно сделать доступными для всех заинтересованных разработчиков.

Таким образом, была подобрана методика, но ее использование было не оптимальным, трудоемким, не вполне удобным и эффективным. Было решено создать удобную систему для прохождения тестирования возможными пользователями приложения, не требующую участия человека, которая бы обрабатывала данные анкет и сохраняла их результаты в базу данных. Информацию из которой, в свою очередь, могли бы просматривать все пользователи, имеющие соответствующие права.

Еще одной целью курсовой работы являлась апробация технологии WebSharper. WebSharper - платформа для разработки веб-приложений на языке F#. Она позволяет создавать веб приложения полностью на F# и запускать их на сервере в среде .NET, или в браузере в среде JavaScript, или и на сервере и на клиенте одновременно. WebSharper сохраняет безопасность типов F# и автодополнение кода Microsoft Visual Studio, тем самым создавая очень продуктивное окружение для JavaScript разработки.

WebSharper включает в себя:

- компилятор для сборки в JavaScript

- поддержку большей части F# и .NET библиотек
- поддержку типобезопасного программирования со стандартной библиотекой JavaScript и DOM
- привязки F# к некоторым сторонним библиотекам, в том числе jQuery
- Formlet - инновационная библиотека для типобезопасных комбинаторов вебформ
- инструменты для использования сторонних JavaScript библиотек
- интеграцию с ASP.NET и Microsoft Visual Studio
- управление зависимостями ресурсов для CSS, рисунков и других вспомогательных файлов

Реализация

Для достижения двух главных целей курсовой работы - апробации технологии WebSharper и создания системы для юзабилити тестирования, следовало выполнить ряд задач:

- изучить F# и WebSharper
- разработать веб-интерфейс для тестирования и веб-интерфейс для удобного просмотра результатов
- организовать сбор, хранение и обработку данных
- проверить работу приложения на возможных пользователях

Форма для опроса

После изучения языка F#, прежде всего, требовалось определить, какие данные о проходящем тестировании человеке могут представлять наибольший интерес для разработчиков, которые в дальнейшем будут просматривать результаты и руководствоваться ими. При работе над интерфейсом большое внимание уделяется роли в приложении будущего пользователя. В данном случае, поскольку QReal:Robots является средой для обучения школьников и студентов основам информатики, можно выделить три основные пользовательские роли: преподаватель, ученик школы и студент.

По плану тестирования, участнику вначале предлагается решить несложную задачу при помощи QReal:Robots. Если участник справляется с ней, ему остается лишь ответить на вопросы анкеты, иначе, он может попробовать решить другую задачу или перейти к анкете. Важно знать, удалось ли ему решить хотя бы одну из этих задач.

И наконец, пользователь заполняет анкету. Использовалась анкета SUS (System Usability Scale) - это один из стандартных опросников. Он состоит из 10 утверждений, следует оценить степень своего согласия с каждым из них по пятибальной шкале. Одним из преимуществ этого опросника является небольшое, по сравнению с другими подобными, количество вопросов. Например, в опроснике QUIS (Questionnaire for User Interface Satisfaction) 27 вопросов, а в Words перечислено 118 слов, из которых нужно

выбрать наиболее подходящие к тестируемой системе. Количество вопросов - важный фактор, так как значительная часть предполагаемых пользователей - дети, они нетерпеливы, не стоит утомлять их слишком длинными анкетами.

Таким образом, нужно было узнать статус участника, успешность выполнения задачи и его ответы на вопросы. Поскольку предполагаемый пользователь, скорее всего, относится к одной из трех ранее указанных категорий, ему будет удобнее выбрать их из выпадающего списка с заранее прописанными вариантами. Чтобы отметить, получилась ли задача, нужно выбрать один из двух возможных ответов (рис. 1). А для каждого утверждения-вопроса анкеты рядом с ним расположено 5 вариантов ответа (ответ на каждый вопрос - число от 1 до 5). В опроснике нужно указать свою степень согласия с приведенными утверждениями, и для человека не естественно выражать ее цифрами, необходимо было ознакомить пользователей с правилами ее заполнения (цифра 1 означает полное несогласие участника, а 5 - абсолютное согласие, цифры от 2 до 4 - промежуточные степени соответственно) (рис. 2).

Анкетирование
Анкета для исследования юзабилити проекта QReal:Robots


Выберите Ваш статус

Задание **Напишите программу, чтобы робот доехал до стенки и остановился.**
 Получилось! Не получилось

рис. 1 выбор статуса и задача

Укажите степень Вашего согласия с утверждениями анкеты, где 1 - полностью не согласен(сна), 5 - абсолютно согласна.

1) Я бы хотел(а) еще поработать с этой программой.

1 2 3 4 5 


2) Программа слишком сложная.

1 2 3 4 5 

3) Этой программой легко пользоваться.

1 2 3 4 5 

4) Мне понадобится помощь, чтобы научиться пользоваться этой программой.

1 2 3 4 5 

5) Разные функции в этом приложении правильно сгруппированы.

1 2 3 4 5 

6) В приложении слишком много несоответствий.

1 2 3 4 5 

7) Большая часть людей очень быстро научится пользоваться этой программой.

1 2 3 4 5 

8) Это приложение очень трудно использовать.

1 2 3 4 5 

9) Я уверенно себя чувствовал(а), используя это приложение.

1 2 3 4 5 

10) Мне пришлось многому научиться, прежде чем я смог(ла) работать с приложением.

1 2 3 4 5 

рис.2 опросник

Форма для просмотра результатов

Имея хранящиеся где-то все необходимые данные о людях, которые проходили тестирование, следовало предоставить эту информацию в удобной для восприятия форме. Поскольку для каждого пользователя набор данных по своей структуре одинаков (статус - одна из 3 строк, успешность выполнения задачи - битовое значение и средний бал анкеты - число от 1 до 100), удобно расположить их в таблице с аналогичными названиями столбцов. Кроме того, важно знать средний балл всех анкет, то есть нужно отобразить и это число.

Вполне вероятно, что зашедший на сайт человек интересуется уровнем юзабилити, достигнутым после внесения каких-то изменений в интерфейс и до следующих серьезных изменений, так что нужно предоставить возможность выбирать даты начала и конца исследования. Для дальнейшей разработки нужно знать оценку интерфейса после последних изменений, поэтому они имеют фиксированную дату окончания тестирования, а именно, текущую дату.

Возможно, внесенные изменения, о влиянии которых на интерфейс необходимы сведения, должны были повлиять в более значительной мере на одну из трех ролей (ученик, студент, учитель). Соответственно, нужно выбрать только анкеты, заполненные этой категорией людей. Поэтому был добавлен и такой параметр.

Кроме того, пользователь может просмотреть только результаты анкет, заполненных теми, кто справился с задачей или наоборот.

Желательно чтобы доступ к данным о юзабилити проекта имели только разработчики QReal:Robots, поскольку, например, просмотр этих сведений респондентами может повлиять на результаты тестирования. С этой целью страницы со статистикой защищены авторизацией. Логин и пароли сверяются с таблицей, и лишь после этого, при корректных введенных данных, пользователю разрешен просмотр результатов тестирования.

Таким образом, вначале пользователь проходит авторизацию. При некорректных данных, ему предложат авторизоваться заново, если же логин и пароль оказались

правильными, пользователь увидит общую статистику, то есть информацию по всем когда-либо заполненным анкетам в виде таблицы, где о каждой анкете дается информация о статусе заполнившего ее, успешности выполнения задачи и средний бал анкеты. Также выводится средний бал по всем данным (просто среднее арифметическое баллов всех анкет). Он так же может выбрать параметры интересующих его анкет (рис. 3)

Статистика с параметрами

Выберите статус интересующих Вас участников

все категории ученики студенты преподаватели

Успешность выполнения задачи:

не имеет значения справился не справился

Укажите дату начала

Укажите дату окончания

Статус	Успешность	Бал анкеты
Студент	решил	60
Студент	решил	62.5
Студент	решил	75
Студент	решил	77.5
Студент	решил	90
Средний бал:		73

рис. 3 статистика базы данных по выбранным параметрам

Апробация WebSharper

Располагая средствами WebSharper(a), все необходимые графические компоненты можно реализовать удобно и компактно при помощи встроенных функций и элементов HTML, таких как Select, поля для ввода текста, специальное поле ввода для пароля, функции для построения таблиц и всевозможные кнопки. Библиотека WebSharper(a) Formlet позволяет расположить элементы в окне браузера, применить к ним стандартные стили, организовать сбор и обработку данных из различных форм. Она также содержит функции, определяющие порядок и вид форм, например, сменяющие друг друга формы - Flowlet. Зависимые формы - при разных результатах заполнения предыдущей формы, после нее отображают разные формы (например, в данном случае если пользователь не справляется с задачей, ему предлагается выбор - либо попробовать решить другую задачу, либо перейти сразу к опроснику, а если справляется - сразу отображается опросник). Однако, возникали определенные сложности в реализации, связанные с отсутствием подробной документации и пользовательских форумов в интернете. Было сложно найти информацию о конкретных функциях и их применении. Не всегда была понятна причина ошибки, которая часто заключалась в отсутствии перед функцией указания, должна ли она компилироваться в JavaScript (такое указание должно быть перед каждой функцией, которая исполняется на клиентской части) или вызываться удаленно. Уведомления о причинах других ошибок тоже часто не давали полезной информации. Кроме того, при внесении в программу изменений, их не было сразу заметно в открывшемся в браузере приложении. Вначале из-за этого начинался спешный поиск причины, по которой не произошло никаких изменений. Однако потом оказалось, что дело в кэше браузера, и приложение будет запускаться оттуда, пока он не будет очищен. Существуют библиотеки, названия которых одинаковы, но пути отличаются несколькими разделами. Причем, библиотека с таким же названием может находиться на несколько уровней раньше на том же пути. Из-за этого возникает путаница и ошибки.

Несмотря ни на что, многие из возникших трудностей удалось преодолеть. После некоторой практики программирования на WebSharper некоторые новые функции легче

воспринимаются по аналогии с теми, которые были изучены раньше. Благодаря тому, что система написана на F# в сочетании с WebSharper, код смотрится удивительно компактно.

Работа с данными

Как уже было отмечено выше, сбор данных удобно реализуется с помощью библиотеки Formlet. Далее, обработка, в случае с опросником заключалась в подсчете среднего бала анкеты по ответам. Он рассчитывается по следующему алгоритму: для вопросов с нечетными номерами из ответа опрашиваемого вычитается 1, для вопросов с четными номерами, ответ вычитается из 5. Затем все 10 цифр складываются и домножаются на 2,5, и в результате имеем число в промежутке от 0 до 100. Интерфейс считается удовлетворительным, если результат больше или равен 68. После подсчета, вся информация заносится в базу данных для хранения. Для второго интерфейса логин и пароль использовались для авторизации пользователя и данные, вводимые пользователем относительно статусов участников, успешности выполнения задачи и дат являлись параметрами для запросов к базе данных, результаты которых потом выдавались пользователю.

Для хранения всей информации, была сделана база данных, состоящая из следующих таблиц: в главной таблице хранится номер статуса участника, информация о том, удалось ли ему решить задачу, средний балл анкеты и дата заполнения. В двух вспомогательных таблицах хранятся расшифровки статусов и вопросы анкеты соответственно (рис. 4)

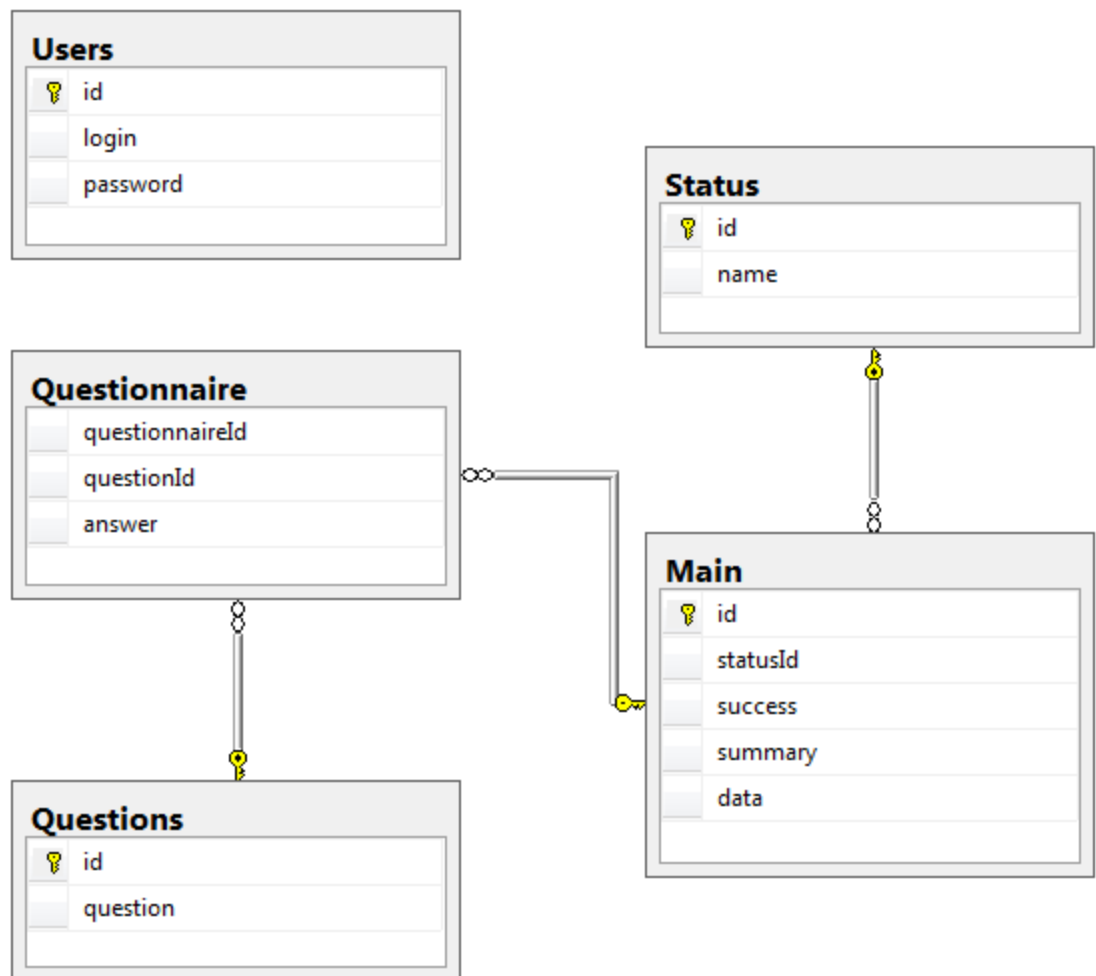


рис. 4 схема базы данных

Тестирование

Мною было проведено тестирование, которое, с одной стороны, должно было показать, насколько удобной для пользователей получилась онлайн система анкетирования, а с другой, выявить уровень юзабилити QReal:Robots. В тестировании приняли участие 12 человек, в основном это были студенты, один из них имел опыт преподавания в визуальной среде программирования. Еще один респондент ранее увлекался программированием роботов. То есть только 2 человека из 12 были знакомы с программами, подобными QReal:Robots. Как было замечено раньше, проведение

тестирования непрофессионалом может повлиять на объективность и без того довольно субъективного результата. Тем не менее, поскольку система только тестировалась, она не была никуда выложена, поэтому участия человека было не избежать. Например, многие люди нервничают и смущаются, когда при попытках быстро освоить новую систему и выполнить в ней какое-то задание, за ним наблюдают, другие подсознательно ждут от наблюдателя помощи. В результате, первые долго не могут решить задачу, и это негативно сказывается на результатах анкеты. Вторые часто совсем не могут с ней справиться, так как даже не пытаются этого сделать, а ждут, когда им все объяснят, поэтому их анкета оказывается заполненной “наобум”. Чтобы уменьшить возможное влияние, в большинстве случаев опрашиваемый находился в соседней комнате. Одно из заданий онлайн анкеты заключается в том, что респонденту предлагается написать программу, при исполнении которой робот бы доехал до стены и остановился. В случае неудачи, опрашиваемый мог попробовать решить другую задачу. Однако ее никто не смотрел, все переходили сразу к опроснику. Половина участников успешно справилась с задачей. У некоторых остальных интерфейс QReal:Robots вызывал недоумение, и они не могли сделать решительно ничего. В среднем, опрашиваемые тратили на попытки справиться с задачей 7-10 минут.

Интерфейс онлайн анкеты показался всем участникам довольно удобным. Некоторые отметили, что хотели бы видеть информацию о QReal:Robots и о том, как с ним работать, а также правильный ответ к задаче. Отмечалось удобство иконок валидации рядом с ответами на вопросы.

Несмотря на сложности, возникшие у многих при решении задачи, результаты анкеты показали, что интерфейс QReal:Robots вполне пригоден. Интерфейс считается удовлетворительным, если результат анкет не меньше 68 баллов. В данном случае, на балл анкеты немного повлияла успешность выполнения задачи. В среднем, респонденты, которые справились с задачей, положительнее оценили приложение, чем те, кто не справился.

Общая статистика

Статус	Успешность	Балл анкеты
Студент	решил	77.5
Студент	решил	62.5
Студент	не решил	52.5
Студент	не решил	57.5
Студент	решил	60
Студент	решил	75
Студент	не решил	67
Студент	решил	90
Ученик	не решил	72.5
Студент	не решил	70
Учитель	решил	72.5
Студент	не решил	67.5
Средний балл:		68.7

рис. 5 результаты всех анкет

Заключение.

Таким образом, в ходе курсовой работы был изучен F# и WebSharper - HTML5 ориентированный фреймворк, позволяющий создавать клиент-серверные веб приложения полностью на F#. При помощи WebSharper'a был реализован веб интерфейс для тестирования юзабилити проекта QReal:Robots и для просмотра результатов такого тестирования. В ходе работы возникли некоторые сложности с WebSharper связанные, в основном, с недостаточно подробной пользовательской документацией и отсутствием пользовательских форумов в интернете. Большую часть таких сложностей удалось преодолеть.

Для апробации получившейся системы и юзабилити QReal:Robots было проведено тестирование. В ходе него не возникло никаких сложностей с интерфейсом анкеты. В тестировании участвовали люди, большинство из которых никогда не работало со средами визуального программирования и, несмотря на некоторые трудности при работе с QReal:Robots, в целом, согласно данным анкет, данное приложение имеет удовлетворительный интерфейс.

Список литературы

1. Thomas S. Tullis and Jacqueline N. Stetson: A Comparison of Questionnaires for Assessing Website Usability, <http://home.comcast.net/~tomtullis/publications/UPA2004TullisStetson.pdf> [дата просмотра 23.05.2013]
2. Jeff Sauro, Measuring Usability With The System Usability Scale (SUS), <http://www.measuringusability.com/sus.php> [дата просмотра 24.04.2013]
3. Якоб Нильсен (перевод - Александр Качанов), Элементарные основы юзабилити, <http://www.webmascon.com/topics/testing/14a.asp> [дата просмотра 24.04.2013]
4. Брыксин Т.А., Литвинов Ю.В., Среда визуального программирования роботов QReal:Robots // Материалы международной конференции "Информационные технологии в образовании и науке". Самара. 2011. С. 332-334
5. WebSharper manual
6. Don Syme, Adam Granicz and Antonio Cisterino "Expert F# 3.0"
7. Психологическая энциклопедия, http://enc-dic.com/enc_psy/JUzabiliti-32262.html [дата просмотра 29.05.2013]