Частное учреждение образования
«Институт современных знаний имени А. М. Широкова»

Кафедра высшей математики и информатики

СОГЛАСОВАНО
Проректор по учебной и научной работе
М. И. Козлович

_____
24.10.2017 г.

ИНФОРМАЦИОННАЯ АРХИТЕКТУРА И USABILITY

Электронный учебно-методический комплекс
для студентов специальности 1-19 01 01 Дизайн (по направлениям),
направление специальности 1-19 01 01-06 Дизайн (виртуальной среды)

Составитель
Слепцов В. Ф., завкафедрой высшей математики и информатики частного учреждения образования «Институт современных знаний имени А. М. Широкова», кандидат технических наук, доцент

Рассмотрено и утверждено
на заседании Совета Института
протокол № 4 от 28.11.2017 г.

УДК 004.72(078)
ББК 32.81я73

Р е ц е н з е н т ы:

Кафедра здорового образа жизни Учреждения образования «Белорусская государственная академия связи» (протокол № 3 от 18.10.2017);

Петров В. А., кандидат физико-математических наук, доцент, доцент кафедры информационных технологий Учреждения образования «Минский инновационный университет».

Рассмотрено и рекомендовано к утверждению
кафедрой высшей математики и информатики
(протокол № 3 от 23.10.2017).

Слепцов, В. Ф. Информационная архитектура и Usability [Электронный ресурс] : учеб.-метод. комплекс для студентов специальности 1-19 01 01 Дизайн (по направлениям), направление специальности 1-19 01 01-06 Дизайн (виртуальной среды) / Авт.-сост. В. Ф. Слепцов. – Электрон. дан. (0,8 Мб). – Минск : Институт современных знаний имени А. М. Широкова, 2017. – 91 с. – 1 электрон. опт. диск (CD).

Систем. требования (миним.) : Intel Pentium (или аналогичный процессор других производителей) 1 ГГц ; 512 Мб оперативной памяти ; 500 Мб свободного дискового пространства ; привод DVD ; операционная система Microsoft Windows 2000 SP 4 / XP SP 2 / Vista (32 бит) или более поздние версии ; Adobe Reader 7.0 (или аналогичный продукт для чтения файлов формата pdf).

Номер гос. регистрации в НИРУП «Институт прикладных программных систем» 1201713896 от 28.11.2017 г.

Учебно-методический комплекс представляет собой совокупность учебно-методических материалов, способствующих эффективному формированию компетенций в рамках изучения дисциплины «Информационная архитектура и Usability».

Для студентов вузов.

ISBN 978-985-547-194-4

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебно-методический комплекс (УМК) по дисциплине «Информационная архитектура и Usability» разработан для студентов специальности 1-19 01 06 «Дизайн», направление специальности 1-19 01 01-06 «Дизайн (виртуальной среды)» Института современных знаний имени А. М. Широкова для эффективного освоения данной дисциплины. Он представляет собой совокупность учебно-методической и нормативной документации, средств обучения и контроля, а также прочих образовательных ресурсов, необходимых для полноценного обучения.

Изучение данной дисциплины позволит студентам приобрести базовые знания по разработке и оценке пользовательских интерфейсов.

Основной целью курса «Информационная архитектура и Usability» является приобретение знаний о компьютерном дизайне и графике, методах представления растровых и векторных изображений, шрифтов, технологиях их обработки, преобразования, изучение теоретических основ построения пользовательских интерфейсов для компьютерных программ и приобретение соответствующих базовых практических навыков их создания и оценки.

УМК включает в себя учебную программу дисциплины, краткий курс лекций, вопросы для самоподготовки, планы практических занятий. В УМК включен список основной и дополнительной литературы, учебно-методических материалов и ресурсов Интернет для освоения полного объема знаний, соответствующего стандартам высшей школы, приведен перечень программного обеспечения, используемого для практического освоения материала во время проведения практических работ и материально-технического обеспечения, а также методические рекомендации студентам по организации самостоятельной работы.

1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

1.1. Конспекты лекций

ТЕМА 1. Особенности взаимодействия человека с компьютерными системами

Использование машин и механизмов привело в начале XX в. к появлению исследований по изучению действий человека в скорости освоения им новой техники, затрат энергии, производительности и интенсивности при различных видах деятельности.

На научной основе началось комплексное изучение прикладных наук, которые находятся на стыке техники, психологии и физиологии труда. Этим изучением занялась наука под названием «Эргономика», которая интенсивно развивается во многих странах мира. По этой дисциплине ведется подготовка специалистов в высших учебных заведениях.

Таким образом, эргономика – это область приложения научных знаний о человеке к проектированию предметов, систем и окружений, используемых им.

При работе с компьютером возникает проблема обеспечения совместимости машины и человека. Назовем некоторые из них:

– *антропометрическая совместимость* – учет размеров тела человека, возможность обзора внешнего пространства, положение оператора при работе;

- *сенсомоторная совместимость* – учет скорости моторных операций человека и его сенсорных реакций на различные виды раздражителей;

– *энергетическая совместимость* – учет силовых возможностей человека при определении усилий, прилагаемых к органам управления;

– *психофизиологическая совместимость* – учет реакции человека на цвет, цветовую гамму, частотный диапазон, форму и другие эстетические параметры машины.

В развитии эргономики можно выделить несколько этапов:

– *повышение производительности труда.* Человек – это определенный вид ресурса, который должен быть использован наиболее полно в технологиче-

ском процессе. Основная задача эргономической работы заключается в том, чтобы определить эти возможности. Такого взгляда придерживались Ф. Тейлор, Г. Мюнстерберг, В. Штерн, И. Н. Шпилькейн, А. К. Гастев, П. М. Керженцев, В. М. Бехтерев, С. Т. Геллерштейн и др.;

– *не повышение производительности труда, а предупреждение срывов.* Речь идет о сокращении разрыва между возможностями человека и предъявляемыми к нему требованиями, о согласованности действий человека и техники. Российский инженер-железнодорожник И. И. Рихтер первым поставил такую задачу и предложил необходимую программу работ.

В результате целью эргономического анализа стало выяснение ограничений возможностей человека и закономерностей функционирования исследуемых процессов, а не возможностей того, что человек смог бы сделать.

Появление компьютерных систем позволяет автоматизировать деятельность человека по обработке информации. Плохо разработанные интерфейсы становятся причиной непредвиденных проблем. В результате возникло научное направление, существующее и развивающееся с целью совершенствования методов разработки, оценки и внедрения интерактивных компьютерных систем, предназначенных для использования человеком, а также исследования различных аспектов этого использования.

Взаимодействие между пользователями и компьютерами происходит на уровне пользовательского интерфейса (далее – интерфейса), который включает в себя программное и аппаратное обеспечение: образы или объекты, отображаемые на экранах дисплеев, данные, полученные от пользователя посредством аппаратных устройств (таких, как устройство ввода и позиционирования и кнопочный манипулятор типа «мышь». Далее – клавиатура и мышь). Взаимодействуют пользователи с крупными автоматизированными системами: это могут быть, например, воздушное судно и электростанция.

Знания, полученные в ходе изучения человеко-компьютерного взаимодействия, опираются как на человеческий, так и на компьютерный факторы.

С компьютерной стороны важны технологии: компьютерной графики, операционных систем, языков программирования и среды разработки. С человеческой стороны: теория коммуникации, графическое и производственное проектирование, лингвистика, социология, когнитивная психология, удовлетворение пользователей, инженерия и проектирование. Благодаря междисциплинарному характеру человеко-компьютерного взаимодействия люди с разным уровнем подготовки вносят вклад в его успех.

Основной задачей такого взаимодействия является улучшение взаимодействия между человеком и компьютером. Человек делает компьютеры более удобными (юзабельными) и восприимчивыми к потребностям пользователей.

Интерфейс (от англ. *interface* – «поверхность раздела», «перегородка») – совокупность средств, методов и правил взаимодействия управления, контроля между элементами системы. Этот термин используется во многих областях науки и техники. Его значение относится к любому сопряжению взаимодействующих сущностей (естественнонаучных, аппаратных и человеко-машинных). Под интерфейсом понимают не только устройства, но и правила (протокол) взаимодействия этих устройств.

При разработке пользовательских интерфейсов словом «юзабилити» обозначается общая концепция удобства использования программного обеспечения, логичность и простота расположения элементов управления. Однако при этом нередко происходит подмена понятий: утилитарных – эстетическими.

Термин «юзабилити» можно назвать синонимом слова «эргономичность». Только последнее определяет минимальность конкретных физических усилий при пользовании вещью, а первое слово – конечную суммарную степень удобства, меру интеллектуального усилия, необходимого для получения полезных качеств этой вещи, и скорость достижения положительного результата при управлении ею.

В более широком значении термин «юзабильность» употребляется как удобство пользования (например, для механических приспособлений и инструментов (дверная ручка или молоток): эргономичность формы будет повышать

юзабильность вещи (то есть «удобство применения», «дружественность и простоту при пользовании», «практичность» и вообще «применимость»)).

Международный стандарт ISO 9241-11 [1] определяет юзабилити как «…степень, с которой продукт может быть использован определенными пользователями при определенном контексте применения для достижения определенных целей с должной эффективностью, продуктивностью и удовлетворенностью» (англ.: *«…the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context ofuse»*).

Юзабилити имеет серьезное значение для показателей конверсии в электронной коммерции. Однако оно означает не только улучшенное визуальное руководство или более совершенную иерархию сайта – это также и больший контакт с потенциальным покупателем посредством профессионально выполненного серьезного дизайна, преподнесения верной информации в нужное время.

Юзабилити – это качественный признак, определяющий степень легкости использования интерфейса пользователем. Слово «юзабилити» обозначает набор методов, служащих для улучшения интерфейса во время процесса его разработки.

Назовем основные аспекты, на которых следует заострять внимание при создании хорошего юзабилити.

1) Дизайн: прежде чем делать проект, следует изучить специальные сетевые источники, которые посвящены дизайнерским премудростям. Еще один момент: дизайн сайта должен отвечать определенным стандартам.

2) Структура контента отражает элемент хорошего юзабилити. Темы, теги, разделы, выпадающие меню – эти детали помогают пользователю ориентироваться на странице, а также переходить на другие разделы сайта. Пользователь должен перемещаться по сайту плавно и интуитивно.

3) Навигация сайта должна быть профессиональной. Более того, навигация является еще и одной из главных функций любого веб-сайта. Сделав навигационную компоненту проекта на профессиональном уровне, можно ожидать

хороших временных показателей пребывания посетителей на сайте. Чтобы сделать хорошую навигацию для сайта, необходимо, прежде всего, поставить себя на место посетителя и побыть в его «шкуре».

4) Функционал сайта должен обеспечивать получение информации после клика по кнопке, ссылке или «менюшке». Но часто после нажатия выдается ошибка, и пользователь остается ни с чем. Такое явление называется плохим функционалом (или проще – ошибкой в коде сайта). Своевременное устранение проблем с «глючными» скриптами, инструментами и ссылками должно стать одним из главных предметов мониторинга проекта. Поэтому следует чаще проверять сайт на работоспособность.

5) Контент сайта. То, как контент позиционируется на сайте, влияет не только на посещаемость последнего, но и на видимость поисковиками самого ресурса в целом. Проблеме «юзабилити контента» посвящено неимоверное количество материалов.

ТЕМА 2. Интерфейс пользователя

Человек непоследователен, часто отвлекается, не может продуктивно работать долгое время и воспринимает только адаптированную для него информацию. А еще люди не умеют (за редким исключением!) оперировать в уме большими числами. И при этом считают, что компьютер должен читать их мысли, быстро и точно выполнять команды, и очень расстраиваются, если этого не происходит.

Компьютер оперирует только двумя сущностями – это ноль и единица. Он работает, пока его не выключат или не прекратится подача электричества. Получив задание от человека, компьютер беспрекословно все выполняет, даже если это его повредит.

Рассмотрим, как происходит взаимодействие пользователя с какой-либо программной системой.

1. Пользователь формирует цель.

Нередко можно услышать фразы от сидящего за компьютером человека: «Он не так работает! Я хочу, чтобы он просто посчитал мне сумму, а он выдает какие-то непонятные цифры». Система не удовлетворила потребности пользователя, и он недоволен. Хотя должно быть иначе: найти телефон какой-нибудь компании в электронном справочнике, узнать расписание самолетов или время работы поликлиники – интерфейс должен быть направлен на обеспечение подобных нужд пользователя.

2. Пользователь решает, как достигнуть цели.

Выбор способа достижения цели зависит от предсказуемости системы, знаний и опыта пользователя, предвидения следующего шага и степени ее участия в работе пользователя – подсказывает ли она все это или пассивно отвечает на запросы. Пока пользователь обдумывает способы совершения последующих действий, в системе ничего не происходит. Именно человек принимает решения, что делать дальше для достижения цели.

Первый постулат о принятии решений гласит: человек не любит принимать решения. Второй: при работе с системой все-таки приходится принимать решения. Третий: человек любит, когда ему преподносят готовое решение или информацию «на блюдечке с голубой каемочкой». Однако вместо обдуманного решения предпочитает наугад щелкать «мышью».

Что можно сделать во избежание ошибок? Постоянно «подсказывать» пользователю, что он движется в верном направлении и указывать на возможные дальнейшие шаги.

3. Пользователь ищет в системе нужный элемент.

Может использоваться не только один из элементов — гиперссылка, кнопка, текст, но и несколько. Например, «тройка» – поле ввода текста – выбор из списка – кнопка».

4. Пользователь активизирует выбранный элемент.

Он щелкает по найденной ссылке (ставит «галочку» в чек-боксе, нажимает кнопку, прокручивает страницу – в зависимости от предыдущего пункта), чтобы перейти непосредственно к результату действия.

5. Пользователь получает от системы результат действия.

Пользователь доволен, потому что получил не сообщение об ошибке или проблеме на сервере, а именно результат своего запроса к системе.

6. Пользователь анализирует результат.

Принимается решение о переходе к следующему действию, если цель достигнута и не надо повторять запрос.

Таким образом, ***интерфейс призван формировать у пользователя привычки,*** складывающиеся независимо от того, думает ли об этом разработчик.

Если одно и то же действие повторяется несколько раз подряд, оно становится привычным. Привычки могут возникать на умственном и физическом уровне, постепенно переводя сознательные действия в бессознательные.

Для того чтобы интерфейс формировал привычки, он должен соответствовать следующим требованиям:

– быть довольно простым, состоящим из достаточно очевидных элементарных действий, так или иначе ведущих пользователя к цели;

– часто использоваться для формирования привычек, ведь иначе можно попросту разучиться работе с ним. Многократно повторяемые действия ведут к автоматизму и созданию привычки, а перерыв в совершении этих действий – к некоторой потере контроля над ситуацией.

Если задание становится привычным и выполняется фактически «на автопилоте», человек устает меньше. Однако иногда вследствие этого теряется контроль над ситуацией – ведь каждое свое действие человек выполняет машинально, то есть *привычно.* Заранее обозначенные действия могут играть как положительную, так и отрицательную роль.

Фокус внимания человека применительно к компьютерным системам – это место на экране, куда направлен его сосредоточенный взгляд. Цели качест-

венного, хорошо проработанного интерфейса различны: с одной стороны, надо удерживать фокус внимания пользователя в определенном месте; с другой – не заставлять перед совершением действия долго размышлять над последствиями. На фокус внимания влияет большое количество факторов, поэтому управлять им достаточно сложно.

Общий принцип управления и удержания фокуса внимания: акцентирование внимания человека к какому-либо элементу, при помощи которого фокус внимания перемещается от одного элемента к следующему – вплоть до достижения цели пользователем или системой. И произойдет это намного быстрее, если человека не отвлекать от работы.

Человек нередко отвлекается, независимо от важности и срочности того, что делает. Если «по пути» он встречает что-то более интересное, то готов на него переключиться. Фокус внимания бывает только один, и человек осознанно думает только об одной задаче, а все остальное в его голове протекает бессознательно. Если бы это было не так, мы никогда не услышали бы фраз типа «Подожди, а то я мысль потеряю», «Секундочку…». При наличии нескольких фокусов внимания человек никогда не терял бы мысль.

Для восстановления фокуса внимания пользователю необходимо сообщить достаточно большое количество информации: где он находится, с чем работает, что уже сделал, на каком шаге остановился и какой элемент является текущим, с точки зрения системы.

Существует четыре способа акцентирования внимания на текущем объекте в системе:

– ***Подсветка.*** Система подсвечивает текущий объект. Например, в браузере Internet Explorer последних версий панель инструментов представляет собой серые пиктограммы, которые становятся цветными при наведении на них курсора. В web подсветка, изменение цвета (или вида) гиперссылки появляется при наведении на нее «мыши»;

– ***Указание*** – это вариант «долговременной» подсветки. Можно указать какую-либо область экрана или объект, если это поддерживается программой. Например: конкретную строку таблицы после «клика» по ней «мышью»;

– ***Выделение*** объясняет, что пользователь обозначает объект, над которым системе нужно произвести действие: установить флажок в чек-боксе или нажать радиокнопку. Можно выделить часть текста на странице или изображение. Например, при выборе варианта дизайна бесплатной гостевой книги на сайте narod.ru пользователь выделяет из девяти представленных тот, что нужен, радиокнопкой;

– ***Активация.*** Активировать – значит сделать элемент интерфейса или объект системы доступным для преобразования. Например, выпадающее меню в некоторых случаях может быть использовано только после активации управляющего элемента, то есть наведения на него курсора. Текущий раздел меню, как правило, идентифицируется при помощи подсветки.

Имеющиеся стандарты по проектированию интерфейсов указывают на то, какими они быть не должны, что весьма незначительно уменьшает множество возможных вариантов построения интерфейса. Исследования в данной области показывают, что любой пользовательский интерфейс (далее – ПИ) должен обеспечивать выполнение четырех основных функций:

1) управление компьютером путем действий пользователя: инициация, прерывание, отмена компьютерных процессов и т. п.;

2) ввод данных, осуществляемый оператором, и отклик системы;

3) отображение данных, вводимых оператором, который управляет этим процессом;

4) поддержка пользователя в процессе деятельности, осуществляемая по каналам обратной связи, в которых циркулирует информация об ошибочных или случайных (не по алгоритму) действиях пользователя.

Эффективный пользовательский интерфейс обеспечивает всестороннее использование потенциальных возможностей пользователя, технических и информационно-программных средств компьютера, высокую безошибочность и

грамотные действия пользователя. Хорошо спроектированный пользовательский интерфейс обеспечивает максимальный комфорт деятельности пользователя и должен способствовать быстрому освоению вычислительной техники оператором, формированию у него устойчивых навыков; удовлетворять потребности человека-пользователя, а не обслуживать процесс обработки данных.

Пользовательский интерфейс состоит из трех основных частей. Это:

1) визуальное оформление, отвечающее за представление информации оператору;

2) функциональные возможности системы, включающие набор элементов для эффективного выполнения профессиональной деятельности;

3) техники взаимодействия оператора с системой.

Однако зачастую разработчики программных продуктов рассматривают функциональность системы отдельно от ее пользовательского интерфейса, не заостряя внимание на элементах взаимодействия пользователя и системы. Другими словами, пользовательский интерфейс является дополнением к функциональным возможностям системы. Пользователи программ, как правило, не разделяют функциональность и пользовательский интерфейс на две составляющие – для них программой является именно пользовательский интерфейс. Взаимодействие с программным продуктом (далее – ПП) формируется при работе с интерфейсом. Поэтапная разработка пользовательского интерфейса повышает эффективность программного продукта, уменьшает время обучения пользователей, снижает стоимость доработки системы после ее внедрения, а также позволяет полностью использовать заложенную в программное обеспечение функциональность.

ТЕМА 3. Информационная архитектура. Опыт взаимодействия

Большая часть профессионально создаваемых программных продуктов, предназначенных для работы непосредственно с пользователями, построена на базе модели MVC. Модель MVC *(model-view-controller*) – способ построения приложения с помощью трех основных компонентов:

1) модель (model) – уровень хранения данных (объектная модель, база данных);

2) представление (view) – уровень внешнего вида данных, доступных пользователю;

3) контроллер (controller) – уровень компонентов, реализующих логику приложения.

Система (например, интернет-магазин), в соответствии с моделью MVC, будет построена следующим образом:

1. В базе данных хранится вся информация о пользователях магазина, заказах, товарах, статусах и т.п. Информация пассивна: если ее отделить от остальных частей системы, то с ней можно делать всевозможные операции. На этом уровне известно, какими функциями обладает система;

2. На уровне представления (то есть на фронтальном) система выдает пользователю результаты работы контроллера и собирает данные для следующих действий. Схема следующая: показываются результаты поиска, пользователь вводит свои данные, выбирает виды доставки и оплаты, кладет товар в «Корзину». На этом уровне пользователю становятся известны результаты работы системы – интерфейс;

3. На уровне контроллера (то есть на логическом) описаны принципы работы этого интернет-магазина: четко обозначены покупатели, имеющие право делать заказы; прописаны правила поиска товаров в базе данных; обозначен путь продвижения товаров от «Корзины» до покупателя, др. На этом уровне становится известно, как система обрабатывает данные – существующие и получаемые извне.

Все уровни разделены на этапы.

1. Есть компьютер пользователя и сервер. Последний обрабатывает поступающую информацию – и на основании набора данных и инструкций возвращает HTML-страницу (независимо от языка программирования, на котором написаны серверные компоненты web-сайта).

2. Пользователь совершает какие-либо действия на странице, и та снова обращается к серверу. Что́ бы система ни «вытворяла» на сервере, пользователю всегда возвращается очередная сгенерированная HTML-страница, и так далее. Таким образом, результаты получаются одинаковыми и при использовании скриптового языка ASP в сочетании с СУБД MS SQL Server, и – при PHP с MySql.

Приведем пример. Требуется создать «Электронный телефонный справочник». Сначала о его интерфейсе ничего не известно, существуют только самые общие данные. Для того чтобы справочник был действительно полезен и удобен, необходимо точно знать: какая именно информация требуется конкретной группе пользователей; состав абонентской аудитории; принципы подачи информации, др. Поэтому в начале создания любого программного продукта необходимо обдумать и сформулировать требования к графическому интерфейсу, который адресован некоторому гипотетическому пользователю. Интерфейс, в свою очередь, должен работать «на пользователя» (в его интересах), позволяя последнему требовать от программы четкого изложения данных. Поскольку пользователи web имеют разные компьютерные комплексы с различным содержанием (так называемой «начинкой»), то интерфейс в обязательном порядке должен грамотно и правильно «организовать» взаимодействие человека и машины. То же самое касается и программного обеспечения, в частности, браузеров.

Проектирование интерфейса – довольно сложный и многоэтапный процесс, каждый этап которого состоит, в свою очередь, из отдельных ступеней. В общей сложности его можно разделить на четыре этапа. При этом достигается оптимальное взаимодействие пользователя с системой. В системе «человек – машина» главное место отводится человеку, а система, являясь подчиненным звеном, выполняет его указания, а не наоборот.

ТЕМА 4. Сценарии использования

При создании интерфейсов возникают вопросы о том, какими функциями наделена программа, и как с ней будет работать пользователь. Одним из ответов на эти вопросы является создание так называемых сценариев использования.

***Сценарий использования**, вариант, прецедент (англ. Use Case)* – в разработке программного обеспечения и системном проектировании это описание поведения системы, когда она отвечает на внешние запросы. Другими словами, сценарий использования описывает, «кто» и «что» может делать с системой.

Методика сценариев использования применяется для выявления требований к поведению системы, известных также как «функциональные требования». В системном проектировании сценарии использования применяются на более высоком уровне, чем в разработке программного обеспечения, часто представляя цели заинтересованных лиц или миссии.

Сценарии использования дают возможность совместно обсуждать функциональность и поведение системы – заказчику, конечному пользователю и разработчику.

Каждый сценарий использования сосредоточивается на описании того, как достигнуть цели или решить задачи. Для большинства программных проектов это означает, что потребуется множество сценариев использования для определения необходимого набора свойств новой системы. Степень формальности программного проекта и его стадии будут влиять на необходимый уровень детализации каждого сценария использования. Свойства системы не идентичны понятию «сценарий использования». Последний может быть связан с одним (или более) свойством системы, а свойство – с одним или более сценарием использования. Другими словами, они взаимодействуют.

Сценарии использования рассматривают систему как «черный ящик», и взаимодействия с системой (включая системные ответы) описываются с точки зрения внешнего наблюдателя. Это – преднамеренная политика, потому что «Актер» (может применяться термин «Актант») вынужден сосредоточиться на

том, что системе предстоит выполнить, а не как это будет сделано, исключая предположения о реализации возможностей.

Сценарии использования могут описываться на абстрактном уровне (деловой сценарий использования, иногда называемый ключевым сценарием использования) или на системном (системный сценарий использования). Различия между ними – лишь в деталях.

Сценарий использования должен:

– описывать, что именно система должна сделать, чтобы автор достиг своей цели;

– не затрагивать детали реализации;

– иметь достаточный уровень детализации;

– не описывать пользовательские интерфейсы и экраны. Это делается во время дизайна.

Для наглядного описания возможностей системы используется унифицированный язык моделирования отношений (UML), в котором между всеми сценариями использования (или их частью) и актерами отношения представлены в виде диаграмм, известных как диаграммы прецедентов (вариантов использования; англ. *use case diagram*). На них изображены отношения, существующие между актерами и прецедентами. Прецеденты служат для документирования функциональных требований к программным системам, описывая целостные фрагменты поведения системы, не вдаваясь при этом в особенности внутренней структуры субъекта. Определение прецедента содержит все свойственные ему виды поведения: основную последовательность, различные варианты стандартного поведения и всевозможные исключительные ситуации – с указанием ответной реакции на них. С точки зрения пользователя, некоторые из видов поведения выглядят как ошибочные. Однако для системы ошибочная ситуация является одним из вариантов поведения, который должен быть описан и обработан.

К сценариям использования в UML применяются следующие виды отношений:

– ассоциация (отношение) (англ. *Association*) указывает на то, что актер инициирует соответствующий вариант использования. В том числе между прецедентами;

– расширение (англ. *Extend*) – разновидность отношения зависимости между базовым вариантом использования и его специальным случаем;

– включение (англ. *Include*) определяет взаимосвязь базового варианта использования с другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении дополнительных условий;

– обобщение (англ. *Generalization* – «наследование») моделирует соответствующую общность ролей.

ТЕМА 5. Проектирование взаимодействия и информационная архитектура

В основе успешного проектирования опыта взаимодействия лежит четко сформулированная стратегия. Знание того, что хочет получить от сайта компания, и что он дает пользователям, позволяет принимать решения по каждому аспекту опыта взаимодействия. Однако ответить на эти простые вопросы труднее, чем кажется на первый взгляд.

Самая распространенная причина провала веб-сайта кроется не в технологии и не опыте взаимодействия. Чаще всего веб-сайты терпят неудачу потому, что перед написанием первой строчки кода, рисованием первого пикселя и установкой первого сервера никто не потрудился ответить на два принципиальных вопроса:

1. Для каких целей создается сайт?

2. Что хотят получить от него пользователи?

Ответив на эти вопросы, обозначим задачи сайта, а также узнаем потребности пользователей. Цели сайта и потребности пользователей вместе образуют уровень стратегии – основу каждого решения, принимаемого в процессе проектирования опыта взаимодействия. Тем удивительнее то, что большое количест-

во проектов по разработке опыта взаимодействия отнюдь не начинаются с четкой, сформулированной стратегии. Между тем, это важно – правильно согласовывать решения с целями.

Первый этап прояснения стратегии состоит в изучении целей, часто существующих лишь в виде туманно высказанных мыслей его разработчиков. И пока такое происходит, люди имеют смутное представление о конечных результатах проекта.

При разработке сайта есть риск попасть в ловушку, ориентируясь на одного идеализированного пользователя – разработчика. Но сайт создается не для себя – для других. Поэтому сначала надо изучить виртуальную аудиторию, понять, что ей нужно. Потратив время на исследование пользовательских потребностей, можно вырваться за рамки своих ограниченных представлений и увидеть сайт глазами пользователей. Это сложно, поскольку пользователи самые разные. Даже если сайт создается для внутреннего использования в организации, все равно придется ориентироваться на широкий диапазон потребностей. А клиентская аудитория увеличит это разнообразие [2].

Есть вещи, которые делаются ради «любви к искусству»: например, одни люди бегают трусцой по утрам, другие разучивают гаммы на фортепиано, третьи пекут пироги. Определение набора возможностей достигает двух целей: 1) процесс; 2) результат.

Процесс выявляет потенциальные противоречия и «шероховатости» конечного продукта на том этапе, когда результат существует лишь в планах. Здесь есть возможность определить, за что следует взяться прямо сейчас, а что отложить на потом. Результат дает команде точку отсчета для всей последующей работы над проектом. Определение требований убирает из процесса разработки неоднозначность. Существуют две основных причины для документирования требований к продукту:

1. необходимо знать, что создается;

2. необходимо знать, что не создается.

Многие функциональные возможности выглядят неплохо на уровне идей, однако не всегда согласуются со стратегическими целями проекта. В процесс его подготовки то и дело возникают разнообразные идеи по созданию новых аспектов. Задокументированные требования отточат механизм оценки идей при работе над проектом.

Определив конкретный набор требований и отложив на будущее реализацию всех запросов, которые им не соответствуют, можно управлять рабочим процессом осознанно и целенаправленно.

Для создания сайта надо выполнить ряд требований. Одним из таких условий являются требования, связанные с брендингом. Другой пример – технические требования (например, поддерживаемые браузеры и операционные системы). В большинстве случаев разработчики, говоря о требованиях, подразумевают краткое описание определенной возможности, которой должен обладать конечный продукт. Лучший способ узнать, чего хотят пользователи, просто спросить их об этом. А затем эти сведения разделить на категории.

Функциональные спецификации представляют собой перечень требований к тому, что должна делать система. Независимо от размера и сложности проекта при формулировании спецификаций следует руководствоваться некоторыми общими правилами.

Будьте позитивны. Вместо описания плохого или неправильного поведения системы опишите ее действия, предотвращающие нежелательный поворот событий. Например, вместо «Система не позволит купить воздушного змея без веревки», лучше написать «При попытке купить воздушного змея без веревки система направит пользователя на страницу, где он сможет приобрести веревку».

Будьте конкретны. Только оставив как можно меньше путей для неправильной интерпретации требований, можно определить, выполнены ли они.

Сравним примеры.

1. Сайт будет доступен людям с ограниченными физическими возможностями.

2. Сайт будет соответствовать разделу 508 Акта о людях с ограниченными возможностями.

На первый взгляд, первый пример производит впечатление четко сформулированного требования, но даже без глубокого анализа видно, какие пробелы он содержит. Что значит «доступен»? Если все картинки на сайте сопровождаются текстовыми описаниями, этого достаточно? Кто считается человеком с ограниченными физическими возможностями? Если на сайте не воспроизводится звук, значит ли это, что сайт доступен и для глухих? Убрав возможность различных интерпретаций, второй вариант требования исключает любые споры, которые могли бы возникнуть в течение или после работы над проектом.

Избегайте субъективных формулировок. Это просто еще один способ выражаться ясно и исключать из требований двусмысленность (а значит, возможность разночтений).

Говоря о контенте, надо иметь в виду не только текст: изображения, звук и видео тоже являются его составляющими. Эти разные виды контента могут комбинироваться: например, контент, относящийся к спортивному событию, может состоять из статьи, иллюстрируемой фотографиями и видеофрагментами. Конкретизация всех типов контента, связанных с данной возможностью системы, позволяет определить необходимые для создания этого контента ресурсы (или хотя бы понять, можно ли его создать, в принципе). Не следует путать ***формат*** элемента контента и его ***назначение.*** В ходе обсуждения с заинтересованными сторонами требований к контенту одним из первых является пожелание: «На нашем сайте должны быть ответы на часто задаваемые вопросы». Однако словосочетание «часто задаваемые вопросы» («ЧаВо», FAQ – Frequently Asked Questions) на самом деле относится к формату контента: это простая последовательность вопросов и ответов.

Собирать идеи относительно требований к продукту не так уж трудно. Сложность тут в том, чтобы определить, какие возможности действительно должны быть включены в проект. На практике довольно редко удается найти взаимно-однозначное соответствие между стратегическими целями и требова-

ниями к продукту. Иногда одно требование связано с несколькими целями, а иногда одна цель ассоциируется с несколькими различными требованиями.

После сбора и ранжирования требований появляется четкая картина того, что именно будет содержать конечный продукт. Однако требования не описывают, каким образом эти части формируют единое целое. Разработка концептуальной структуры сайта – задача следующего уровня.

Уровень структуры – третий из пяти уровней. Здесь осуществляется переход от абстрактных вопросов стратегии в сторону конкретных факторов, определяющих, что, в конечном счете, будет испытывать пользователь. Однако граница между абстракцией и конкретикой бывает размытой. Хотя многие решения, принимаемые на этом этапе, окажут ощутимое влияние на разработанный сайт, сами по себе они опираются на концептуальные понятия.

В традиционном подходе к разработке программного обеспечения создание структурированного опыта взаимодействия называется проектированием взаимодействия. В сфере создания контента структурирование опыта взаимодействия – это вопрос информационной архитектуры. Проектирование взаимодействия имеет отношение к реализации возможностей, позволяющих пользователю решать задачи, а информационная архитектура занимается реализацией возможностей, связанных с предоставлением пользователю информации. Эти понятия кажутся высокотехнологичными областями, доступными лишь «посвященным», однако на самом деле не имеют никакого отношения к технологиям. Они связаны, скорее, с психологией: как люди, обладающие определенными знаниями, думают и работают.

Проектирование взаимодействия – это описание возможного поведения пользователя и определение того, как система будет реагировать на его поведение, приспосабливаться к нему. Каждый раз, когда человек работает на компьютере, происходит некое подобие танца: пользователь делает движение, система на него реагирует, пользователь двигается в ответ – и танец продолжается.

При реализации проектов по разработке программных систем и моделированию бизнес-процессов встречаются ситуации, когда решение проблем в

различных проектах имеют сходные структурные черты. Попытки выявить похожие схемы или структуры в рамках объектно-ориентированного анализа и проектирования привели к появлению понятия паттерна, которое из абстрактной категории превратилось в непременный атрибут современных CASE-средств.

Паттерны различаются степенью детализации и уровнем абстракции – согласно следующей общей классификации по категориям их применения:

– архитектурные паттерны;

– паттерны проектирования;

– паттерны анализа;

– паттерны тестирования;

– паттерны реализации.

Архитектурные паттерны (Architectural patterns) – множество предварительно определенных подсистем со спецификацией их ответственности, правил и базовых принципов установления отношений между ними. Архитектурные паттерны предназначены для спецификации фундаментальных схем структуризации программных систем. Наиболее известными паттернами этой категории являются паттерны GRASP (General Responsibility Assignment Software Pattern). Они относятся к уровню системы и подсистем, но не к уровню классов. Как правило, формулируются в обобщенной форме, используют обычную терминологию и не зависят от области приложения.

Паттерны проектирования (Design patterns) – специальные схемы для уточнения структуры подсистем или компонентов программной системы и отношений между ними – описывают общую структуру взаимодействия элементов программной системы, которые реализуют исходную проблему проектирования в конкретном контексте. Наиболее известными паттернами этой категории являются паттерны GoF (Gang of Four), названные в честь Э. Гаммы, Р. Хелма, Р. Джонсона и Дж. Влиссидеса, которые систематизировали их и представили общее описание.

Паттерны тестирования (Test patterns) – специальные схемы для представления общей организации процесса тестирования программных систем. К этой категории относятся такие паттерны, как тестирование «черного ящика», «белого ящика», отдельных классов, системы. Некоторые из них реализованы в инструментальных средствах, наиболее известными из которых является IBM Test Studio. В связи с этим паттерны тестирования иногда называют стратегиями или схемами тестирования.

Паттерны реализации (Implementation patterns) – совокупность компонентов и других элементов реализации, используемых в структуре модели при написании программного кода. Эта категория паттернов делится на следующие подкатегории: паттерны организации программного кода, паттерны оптимизации программного кода, паттерны устойчивости кода, паттерны разработки графического интерфейса пользователя и др. Некоторые из них реализованы в популярных интегрированных средах программирования в форме шаблонов создаваемых проектов. В этом случае выбор шаблона программного приложения позволяет получить некоторую заготовку программного кода.

Собственное представление пользователей о поведении интерактивных компонентов называется концептуальной моделью. Возьмем, например, элемент контента. Что это: место, которое посещает пользователь, или объект, им получаемый? На сайтах применяются различные подходы. Знание концептуальной модели позволяет принимать последовательные проектные решения. Неважно, будет ли элемент контента местом или объектом. Важно, чтобы сайт вел себя последовательно, а не представлял элемент местом и объектом – попеременно.

Значительная часть любого проекта, связанного с проектированием взаимодействия, включает в себя обработку «ошибок пользователя». Что должна делать система, когда люди совершают ошибки? Что она может предпринять для предотвращения этих ошибок?

Первая и самая лучшая защита от ошибок – разработка такой системы, в которой ошибки пользователей просто невозможны. Хороший пример такого

типа защиты имеется в любом автомобиле с автоматической коробкой передач. Попытка запустить мотор при включенной трансмиссии может повредить чувствительный и сложный механизм; более того, машина не поедет, а лишь резко дернется вперед. Это плохо: для машины, водителя и невинного пешехода, случайно оказавшегося рядом.

Чтобы этого не происходило, любой автомобиль с автоматической трансмиссией оборудован стартером, который не сработает, если рычаг выбора режима не находится в положении «парковка». Поскольку запустить двигатель при включенной трансмиссии невозможно, ошибка исключена. К сожалению, большинство пользовательских ошибок не столь легко предупредить.

Следующий способ исключить ошибки – сделать их невозможными. Однако даже при самых серьезных мерах предосторожности некоторые все-таки произойдут. Тогда система должна сделать все возможное, чтобы помочь пользователю осознать ошибку и устранить ее.

Информационная архитектура связана с созданием организационных и навигационных схем, обеспечивающих экономичное и эффективное перемещение по сайту. Она имеет прямое отношение к вопросам информационного поиска – проектированию систем, позволяющих пользователям легко находить нужную информацию. Однако архитектура веб-сайтов часто призвана решать более широкие задачи, чем оказывать помощь поиска информации: во многих случаях сайтам приходится обучать, информировать и убеждать пользователей.

Обычно решение задач информационной архитектуры требует создания классификационных схем, соответствующих целям сайта, потребностям пользователей и контенту сайта. Есть два подхода к разработке такой классификации: нисходящий («сверху вниз») и восходящий («снизу вверх»).

Нисходящий подход к созданию информационной архитектуры заключается в ее построении непосредственно на основе целей сайта и потребностей пользователей. Начиная с самых общих категорий будущего контента и функциональных возможностей, необходимых для достижения этих стратегических целей, категории логически разбиваются на подкатегории. Получившаяся ие-

рархическая структура служит пустой оболочкой для контента и функциональности.

Восходящий подход к построению информационной архитектуры также состоит в выделении категорий и подкатегорий, но при этом в основу ложится анализ контента и функциональных требований. Начиная с имеющегося исходного материала (или того, который будет в наличии к моменту запуска сайта), элементы группируются в категории низшего уровня, а затем – в более крупные. Так выстраивается структура, отражающая цели сайта и потребности пользователей.

Ни один из этих подходов не лучше другого. При нисходящей разработке архитектуры можно иногда упустить из виду важные детали, касающиеся контента. Восходящий подход настолько точно подогнан под имеющийся контент, что иногда приводит к архитектуре, которую невозможно в последующем даже изменить. Единственный способ обойти эти ловушки на пути к конечному продукту – достичь баланса между нисходящим и восходящим подходами.

Веб-сайты со временем разрастаются и меняются. Однако незначительное количество новых требований, возникающих по ходу их эксплуатации, не должно приводить к полному пересмотру их структуры. Накопление новой информации может привести к пересмотру принципов организации сайта. Например, архитектура, позволяющая пользователям листать пресс-релизы в режиме «день за днем», вполне годится, когда речь идет о нескольких месяцах, однако через пару лет более удобной и практичной может стать организация пресс-релизов по темам.

Элементарной единицей информационных структур является **узел,** который соответствует фрагменту информации любого объема. Он может быть всего лишь числом (например, цена товара), или представлять собой целую библиотеку. Работая с узлами вместо страниц, документов или компонентов, можно пользоваться единым языком и набором структурных концепций при решении широкого спектра разнообразных задач [3].

Абстрактное понятие узла также позволяет детализировать уровни. Большинство проектов по созданию архитектуры веб-сайтов ограничиваются лишь упорядочением страниц. Идентифицирование страницы в виде базового узла дает возможность не заниматься информационными элементами меньшего объема. Если же страница является слишком мелкой единицей для текущего проекта, узлом можно назвать целый раздел сайта.

Узлы могут быть организованы самыми разными способами, но все эти многочисленные структуры в действительности подпадают под один из нескольких классов.

В **иерархической структуре**, иногда называемой **деревом** (или системой «узел–спица»), узлы имеют отношения типа «родитель–потомок». Узлы-потомки представляют более узкие понятия в пределах широкой категории, представленной узлом-родителем. Не каждый узел имеет «детей», но у каждого (кроме самого верхнего) есть родитель. Последовательно переходя от потомка к его родителю, можно в конце концов дойти до родителя всей структуры (или «корня дерева»). Поскольку концепция иерархических отношений хорошо понятна пользователям, а компьютеры все равно работают с иерархиями, это самый распространенный тип структур.

Матричная структура позволяет пользователю перемещаться от узла к узлу в двух и более «измерениях». Матричные структуры уместны, когда нужно обеспечить навигацию по одному и тому же контенту пользователям с разными потребностями, поскольку каждая пользовательская потребность может быть ассоциирована с некоторой «осью координат» в матрице. Например, если одни пользователи предпочитают искать товар по цвету, а другие – по размеру, то матрица поможет обслужить обе группы. Матрица с числом измерений более трех превратится в источник проблем, если считать, что пользователи станут применять ее в качестве основного инструмента навигации: человеческий мозг слабо приспособлен к визуализации перемещений в четырех и более измерениях.

Органические структуры не пытаются следовать какому-либо регулярному шаблону. Узлы соединяются произвольным образом, четкое понятие «разделов» в архитектуре отсутствует. Органические структуры подходят для работы с набором элементов, связи между которыми неясны или подвержены изменениям. Однако такие структуры не позволяют пользователю понять, в какой точке архитектуры он находится. Если пользователь хочет чувствовать себя свободным исследователем (например, в развлекательных или просветительских целях), то органическая структура для этого станет подходящим решением. В то же время она может вызвать трудности у пользователя, который захочет быстро вернуться к уже рассмотренному элементу контента.

Последовательные структуры знакомы всем по традиционным источникам информации. Пример: это книга. Последовательный языковой поток является основным среди имеющихся видов информационной архитектуры, и мы обладаем врожденными способностями к его обработке. Книги, статьи, звуко- и видеозаписи созданы специально для последовательного восприятия. Во Всемирной паутине последовательное представление используется для мелкомасштабных структур – отдельных статей или разделов. Более крупные структуры с последовательной организацией обычно применяются в приложениях, в которых порядок представления содержимого является существенным для удовлетворения пользовательских потребностей, например, в инструкциях.

Даже если структура безупречно отражает то, как пользователи представляют себе тематику сайта, они не смогут ориентироваться в архитектуре, если не в состоянии понять классификационную номенклатуру – описания, заголовки и прочую терминологию, используемую на сайте. Поэтому необходимо говорить на языке пользователей, причем употреблять его правильно. Средство, используемое для этой цели, называется словарем нормативной лексики.

Состав документов, необходимых для описания структуры сайта (от конкретных деталей номенклатуры и метаданных – до общей картины информационной архитектуры и конкретной организации взаимодействия), может варьироваться в зависимости от сложности проекта.

В проектах с большим объемом иерархически организованного контента эффективным способом документирования архитектуры могут оказаться простые многоуровневые списки. В некоторых случаях для отражения нюансов сложной архитектуры потребуются электронные таблицы и базы данных.

Однако самым главным инструментом документирования информационной архитектуры и проектирования взаимодействия является следующая схема:

«Визуальное представление структуры – наиболее эффективный способ разобраться в ответвлениях, группах и взаимосвязях компонентов сайта». Структура веб-сайта достаточно сложна, и ее описание вряд ли кто-нибудь станет читать.

Удачными интерфейсы можно назвать, если пользователи сразу замечают важную информацию (а не столь важная остается незамеченной). Самая большая трудность в разработке интерфейса сложных систем – определить, какие аспекты не нужны пользователям, и перевести их в разряд неприметных (или исключить совсем). Интерфейс, в котором маловероятным ситуациям придается такой же вес, как и потребностям огромного количества пользователей, обречен на недовольство со стороны любой аудитории.

Дизайн навигации кажется простым делом: нужно всего лишь расставить на каждой странице ссылки, чтобы пользователь мог ориентироваться на сайте. Однако трудностей навигационного дизайна здесь довольно много.

Дизайн навигации на любом сайте должен одновременно решать три задачи:

1. Предоставлять пользователям способ, чтобы попадать из одной точки сайта в другую. Поскольку во многих случаях связать каждую страницу со всеми остальными невозможно, приходится подбирать навигационные элементы так, чтобы они упрощали реальные передвижения пользователя; в числе прочего это подразумевает, что ссылки должны быть рабочими;

2. Отражать взаимоотношения между внутренними элементами навигации. Просто предоставить список ссылок недостаточно. Как эти ссылки соотносятся друг с другом? Являются ли одни более важными, чем другие? Какая ме-

жду ними разница? Эта информация необходима пользователю для определения выбора;

3. Отражать связь между содержательной стороной элементов навигации и страницей, которая находится перед глазами пользователя. Какое отношение имеет вся эта куча ссылок к странице, на которую я сейчас смотрю? Эта информация поможет пользователю понять, какой выбор ему следует сделать, чтобы наилучшим образом достичь своей цели или решить стоящую перед ним задачу.

Информационный дизайн не всегда можно «потрогать руками». Часто он является связующим материалом, который скрепляет другие компоненты дизайна. Однако во всех случаях информационный дизайн сводится к принятию решений о том, как представить информацию, чтобы людям было легче воспринимать и использовать ее.

Порой информационный дизайн принимает визуальную форму. Будет ли секторная диаграмма оптимальной для представления этих данных или нашим пользователям лучше подойдет гистограмма? Сможет ли пиктограмма с биноклем адекватно передать понятие «поиск на сайте» или пиктограмма увеличительного стекла будет понятнее?

Макет страницы – это место встречи информационного дизайна, дизайна интерфейса и дизайна навигации, которые совместно формируют единую конструкцию, связывающую части в целое. Макет страницы должен включать в себя все навигационные системы, имеющиеся на сайте и отражающие разные взгляды на архитектуру сайта; все элементы интерфейса, необходимые для использования функциональности этой страницы; а также информационный дизайн, поддерживающий как вышеупомянутые элементы, так и собственно контент страницы.

Дизайн интерфейса – это организация элементов, облегчающая взаимодействие, дизайн навигации – организация элементов, упрощающая передвижение по сайту, а информационный дизайн – организация элементов для донесения информации до пользователя.

Визуальное представление логического порядка элементов, образующее компоновку сайта, расположено на уровне поверхности. Информационный дизайн группирует информационные элементы страницы, а визуальный дизайн определяет, как все это представить визуально.

Один из самых простых способов оценить визуальный дизайн страницы – ответить на следующие вопросы: «Куда в первую очередь направляется взгляд?»; «Какой элемент дизайна первым притягивает внимание пользователя?». Исследователи применяют сложное оборудование, отслеживающее движение зрачков (eyetracking), когда хотят точно знать, на что смотрят испытуемые, и как их взгляд перемещается по странице. Но в целях оптимизации визуального дизайна страницы можно просто спросить других людей или даже самого себя. Такой подход вряд ли сумеет отразить все те нюансы, которые можно выявить с помощью специального оборудования, но в большинстве случаев простого опроса будет вполне достаточно.

В визуальном дизайне основным инструментом привлечения внимания пользователя является контраст. Дизайн без контраста воспринимается как серая невыразительная масса, по которой взгляд пользователя бесцельно блуждает, не понимая, где ему задержаться.

Контраст жизненно важен для привлечения внимания пользователя к существенным аспектам интерфейса; он помогает пользователю разобраться в отношениях между навигационными элементами на странице; и, наконец, служит основным средством обозначения концептуальных групп в информационном дизайне.

Цвет – один из самых эффективных способов передачи идентичности бренда. Некоторые бренды настолько тесно ассоциируются с цветом, что при одной лишь мысли о компании на ум автоматически приходит определенный цвет. Такие фирмы, как Coca-Cola, Kodak или «Велком» в течение многих лет систематически используют конкретные цвета (красный, коричневый, желтый), чтобы создать в общественном сознании устойчивое представление о своей идентичности.

Использование вышеназванных цветов исключает из репертуара дизайнера все остальные. Основные цвета бренда обычно являются частью более широкой цветовой палитры, применяемой во всех материалах компании.

ТЕМА 6. Критерии оценки качества пользовательских интерфейсов

Существует четыре основных критерия качества любого интерфейса:

– скорость работы пользователей;

– количество человеческих ошибок;

– скорость обучения;

– субъективное удовлетворение пользователей (подразумевается, что соответствие интерфейса задачам пользователя является неотъемлемым свойством интерфейса).

Скорость выполнения работы является важным критерием эффективности интерфейса. В чистом виде этот критерий ценят довольно редко, но почти всегда он является крайне желательной составляющей целого. Любая попытка как-то увеличить производительность труда всегда встречается положительно.

Длительность выполнения работы пользователем состоит из следующих составных частей:

– длительности восприятия исходной информации;

– длительности интеллектуальной работы;

– длительности физических действий пользователя;

– длительности реакции системы.

Длительность восприятия исходной информации в комментариях не нуждается. Пользователь должен иметь представление, какая информация о выполняемой задаче у него существует, и в каком состоянии находятся средства, с помощью которых он будет решать данную задачу. Основное время здесь пойдет на считывание показаний системы.

Длительность интеллектуальной работы оценивается тем, что взаимодействие пользователя с системой состоит из семи шагов:

1) формирование цели действий;

2) определение общей направленности действий;

3) Определение конкретных действий;

4) выполнение действий;

5) восприятие нового состояния системы;

6) интерпретация состояния системы;

7) оценка результата.

Из этого списка становится видно, что процесс размышления занимает почти все время, в течение которого пользователь работает с компьютером, во всяком случае, шесть из семи этапов полностью заняты умственной деятельностью. Соответственно, повышение скорости этих размышлений приводит к существенному увеличению скорости работы. К сожалению, существенно повысить скорость собственно мышления пользователей невозможно. Но уменьшить влияние факторов, усложняющих и, соответственно, замедляющих процесс мышления, можно.

Длительность физических действий пользователя прежде всего зависит от степени автоматизации работы и степени необходимой точности работы. Об автоматизации что-либо конкретное сказать сложно. Понятно, что чем больше работы делает компьютер, тем лучше. Непонятно только, как это можно универсально описать, поскольку степень автоматизации сильно зависит от автоматизируемого процесса.

Важным критерием эффективности интерфейса является количество человеческих ошибок. В некоторых случаях одна или две ошибки погоды не делают, но при условии, что их легко исправить. Однако часто минимальная ошибка приводит к совершенно катастрофическим последствиям, например, за одну секунду оператор в банке может сделать кого-то богаче, а банк, в свою очередь, беднее (впрочем, обычно беднее становятся все). Классический сюжет из жизни летчика, который после взлета хотел убрать шасси, но вместо этого включил систему аварийного катапультирования, возник отнюдь не на пустом месте.

Человек при работе с компьютером постоянно совершает ошибки. Дело в том, что компьютеры (как и все сложные технические системы) вообще не могут быть используемы человеком без совершения ошибок. Они требуют от человека точности, логического мышления, способности абстрагироваться от идей реального мира. Человек же на это не способен. Он – не цифровая система, неспособная на ошибку, но система аналоговая. Именно благодаря этому он слаб в логике, зато имеет интуицию, не приспособлен к точности, однако может подстраиваться к ситуации, плохо абстрагируется, но хорошо разбирается в реальном мире.

Человеческие ошибки естественны, следовательно, система, неспособная сама их обнаружить и исправить, порочна. Таким образом, человеческих ошибок не бывает; бывают ошибки в проектировании систем. Сам термин «человеческая ошибка» до сих пор существует только по двум причинам. Во-первых, люди в ошибках системы склонны винить себя, поскольку по собственному эгоцентризму полагают, что подобные вещи происходят только с ними. Во-вторых, существующее положение вещей очень выгодно всякому руководству: гораздо легче уволить кого-либо, нежели признать, что система спроектирована плохо.

Под словосочетанием «человеческая ошибка» нужно понимать «действие пользователя, не совпадающее с целью действий этого пользователя».

Помимо классификации человеческих ошибок, существует еще одна классификация. В ней ошибки расставлены по уровням их негативного эффекта:

1) ошибки, исправляемые во время совершения действия (например, пользователь удаляет файл, перетаскивая его в «Корзину», и замечает, что пытается стереть не тот файл);

2) ошибки, исправляемые после выполнения действия (после ошибочного удаления файла его копия переносится из «Корзины»);

3) ошибки, которые исправить можно, но с большим трудом (реальное стирание файла, при котором не остается копии в «Корзине»);

4) ошибки, которые невозможно исправить или обнаружить формальной проверкой (смысловая ошибка в тексте, удовлетворяющая правилам языка).

Последних ошибок нужно стараться избегать, не считаясь с потерями, поскольку каждая такая неточность обходится гораздо дороже, чем любая ошибка из пункта № 3. Но не каждый хороший дизайнер интерфейса, умеющий мыслить системно, знает, что ошибок из пункта № 2 нужно также не делать, поскольку они тоже дороже тех, что расположены в № 1. С ошибками из пункта № 4 все ясно. Всякий раз, когда мы теряем возможность (например, проверить корректность данных или самой системы), то вступаем на слишком уж «скользкий» путь. Межпланетные зонды из-за ошибок в ПО улетают не туда, куда надо; коммерческие договоры, в которых обнаруживаются ошибки, приносят много неприятностей; ошибочные номера телефонов в записной книжке не дают возможности найти абонента… Это – примеры неисправляемых ошибок**.** Именно поэтому они гораздо хуже ошибок, которые исправить трудно, но они, по крайней мере, сразу видны.

Исправляемые ошибки снижают производительность работы. Любое действие пользователя состоит из семи шагов. Всякий раз, когда пользователь обнаруживает свою ошибку, ему приходится возвращаться назад на несколько этапов. В результате значительный процент времени уходит не на действие (то есть продуктивную работу), а на исправление ошибок.

Наличие человеческих ошибок, которые нельзя обнаружить и исправить до окончательного совершения действия, всегда свидетельствует о недостаточно хорошем дизайне.

В традиционной науке о человеко-машинном взаимодействии роль обучения операторов чрезвычайно велика. Дополнением к самой системе является методология обучения ее будущих пользователей: если, согласно нормативам, человек будет признан неподходящим элементом, к системе его просто не допустят. Напротив, с ПО и сайтами ситуация принципиально иная: как цель, ставится возможность работы с системой любому человеку, независимо от его свойств и навыков. При этом целенаправленное обучение пользователей, как

правило, не производится. Все это делает проблему обучения пользователей работе с компьютерной системой чрезвычайно важной. Есть непреложный закон природы: люди делают что-либо только при наличии стимула, при этом тяжесть действия пропорциональна его силе. Применительно к компьютерным системам этот закон действует без каких-либо исключений. Обучение есть действие: если изучать легко, пользователям будет достаточно слабого стимула; если тяжело – стимул придется увеличивать**.**

Обычно считается, что в случае ПО есть два способа повысить эффективность обучения (помимо метода «обучения плаванию посредством выбрасывания из лодки»): бумажная документация и «оперативная справка». Но существуют другие и более эффективные способы. Рассмотрим некоторые их них:

– общая «понятность» системы;

– обучающие материалы.

Термин «понятность» включает в себя четыре составляющих: ментальную модель, метафору, аффорданс, стандарт.

Ментальная модель. Чтобы успешно пользоваться какой-либо системой, человеку необходимо однозначно понимать, как она работает. При этом необязательно точно понимать сущность происходящих процессов. Это понимание сущности системы называется ментальной моделью.

Метафора. Чтобы научиться пользоваться системой, пользователю нужно построить ментальную модель этой системы. Избавить его и от этой работы можно применением метафоры, которая позволяет пользователю не создавать новую модель, а воспользоваться готовой, построенной ранее по другому поводу. Анализируя опыт применения метафор, можно вывести следующие правила:

– опасно полностью копировать метафору, достаточно взять из нее самое лучшее;

– не обязательно брать метафору из реального мира, ее смело можно придумать самому;

– эффективнее всего метафорически объяснять значение отдельных объектов: например, для графической программы слои можно представлять как по-

ложенные друг на друга листы стекла (этот пример подходит и для предыдущего пункта);

– если метафора хоть как-то ограничивает систему, от нее необходимо немедленно отказаться.

Аффордансом называется ситуация, при которой объект показывает субъекту способы использования своими неотъемлемыми свойствами. Например, надпись «На себя» на двери не является аффордансом, а облик двери, который подсказывает человеку, что она открывается движением «внутрь», несет в себе аффорданс. Польза аффорданса заключается в том, что он позволяет пользователям обходиться без какого-либо предварительного обучения, что является самым эффективным и надежным средством обеспечения понятности.

В приложении могут использоваться различные справочные материалы. Приведем достоинства и недостатки некоторых таких материалов. Справка состояния отвечает на вопрос: «Что происходит в настоящий момент?». Поскольку она требуется именно в данный момент, то не может быть вынесена из интерфейса.

Бумажная книга обычно в веб-приложениях не применяется. Справочная карта – отдельная краткая бумажная документация, демонстрирующая основные способы взаимодействия с системой (quick reference card). Структурированная электронная документация плохо предназначена для чтения больших объемов материала, зато обеспечивает легкий поиск и не имеет ограничений на объем.

Фрагменты пространства интерфейса показывают справочную информацию. Занимают пространство экрана, но пространство ограниченное. Отвлекают внимание, но, как минимум, один раз воспринимаются всеми пользователями. Всплывающие подсказки хорошо справляются с ответом на вопросы «Что это такое?» и «Зачем это нужно?», при условии, что объем ответов сравнительно невелик.

Пользователи выбирают не оптимальный путь в поисках необходимой информации. Им требуется не самое лучшее и надежное решение. Напротив –

часто они готовы удовлетвориться быстрым, то есть «вполне приемлемым» решением.

При разработке проекта пользовательского интерфейса учитываются мнение художника, человеческий фактор и интуиция потенциального пользователя. Некоторые основные принципы построения оконных интерфейсов проясняются после длительной работы с ними.

Якоб Нильсен – успешный программист и инженер, работающий в сфере веб-дизайна. Главным его достижением является основание компании «Nielsen Norman Group» совместно с бывшим вице-президентом фирмы «Apple Computer» Дональдом Норманном. Наибольший опыт Я. Нильсен получил во времена работы в компании «Sun Microsystems» в должности ведущего инженера. Также он являлся специалистом по юзабилити – теории о целесообразности и удобстве использования предмета по отношению к заявленной цели. Вся деятельность Якоба Нильсена направлена, в первую очередь, на то, чтобы сделать работу в сети интернет для пользователей максимально удобной и доступной. Он – автор множества полезных изобретений и владеет 31-ю патентом. Также в арсенале Якоба Нильсена немало книг на темы веб-дизайна, пользовательских интерфейсов, структуры сайтов и многое другое. Одна из самых известных его книг – «Веб-дизайн». За этим высококлассным специалистом давно закрепилось прозвище «адвокат пользователей», коим он сам себя окрестил. Ввиду тех достижений, которых он добился в своей работе по борьбе с «темной стороной интернета», это не оспаривается никем. Именно Я. Нильсен определил десять основных принципов, благодаря которым может быть создан успешный интерфейс сайта:

1. Главное в интерфейсе – его простота. Упрощение навигации и фильтрация от ненужной информации сделают сайт намного привлекательнее;

2. Большое значение имеет скорость загрузки сайта. Достичь этого можно при помощи совмещения небольших страниц с использованием быстрых серверов;

3. Не последнюю роль в наполнении сайта играет лаконичность используемого контента. Вся информация, представленная на веб-страницах, должна строго соответствовать тематике самого сайта. Ничего лишнего!

4. Важно сделать сайт удобным для пользователей со всех уголков планеты. Интернет – по определению – интернационален. В этой связи нежелательно использовать на страницах выражения из местных диалектов или термины, понятные узкому кругу лиц;

5. Для хорошего сайта необходим удобный поисковик, позволяющий легко ориентироваться посетителю. Его рекомендуется разместить в правом верхнем углу главной страницы;

6. Целесообразнее использовать программное обеспечение, уже проверенное и применяемое, как минимум, два года. Самые последние технологии зачастую несовершенны и могут повредить работе сайта;

7. Информация о самом сайте непременно должна быть размещена на главной странице. Не стоит заставлять пользователя искать это вручную;

8. Необходимо на видном месте разместить расценки на предлагаемые сайтом товары или услуги в том случае, если сайт имеет коммерческую направленность;

9. Обязательно стоит позаботиться о том, чтобы в браузере можно было увеличить размер шрифта. Многие пользователи имеют неважное зрение;

10. Регулярное обновление сайта позволит удержать на нем большинство постоянных посетителей, а также привлечь внимание новых.

ТЕМА 7. Тестирование пользовательских интерфейсов

Юзабилити-тестирование – это эксперимент, выполняемый с целью определения того, насколько хорошо люди могут использовать некий искусственный объект для его предполагаемого применения, то есть юзабилити-тестирование измеряет полезность объекта. Юзабилити-тестирование – это метод оценки удобства использования продукта, основанный на привлечении пользователей в качестве тестирующих.

В ходе эксперимента участники выполняют серию заранее подготовленных заданий, за ходом выполнения которых следят наблюдатели, которые отмечают, с какими сложностями сталкивается каждый из участников.

Юзабилити-тестирование сайтов позволяет выявить, отвечает ли сайт поставленным целям, а также дает ответы на следующие вопросы:

– Удачно ли пользователи решают поставленные перед ними задания?

– Насколько быстро было выполнено каждое из них?

– Сколько страниц (кликов) было использовано пользователями для выполнения каждого задания?

– Насколько удовлетворены пользователи сайтом?

– Какие изменения нужно внести, чтобы сделать сайт более успешным?

Исследование и оценку сайтов можно проводить разными методами, разработанными экспертами по юзабилити. Общим для всех методов является постановка реальных задач перед пользователями, а также фиксирование результатов тестирования для дальнейшего анализа**.** Для участия в юзабилити-тестировании отбираются пользователи, соответствующие целевой аудитории сайта, они также не должны быть слишком знакомы с разработкой сайта, так как быстрее других смогут разобраться с его устройством, а это может создать иллюзию, что сайт понятен и удобен для целевых посетителей. Для того чтобы выявить и оценить наибольшее количество имеющихся на сайте проблем, для тестирования необходимо привлекать реальных пользователей.

Для того чтобы провести качественное юзабилити-тестирование, необходимо подготовить соответствующие вопросы. Например, если надо протестировать, насколько удобна функция поиска информации на сайте, следует сконцентрировать внимание на ответах на следующие вопросы:

– Используют ли пользователи поиск или просто «кликают» по страницам?

– Преимущественно какие слова и фразы используются для поиска?

– Расположена ли строка поиска в удачном месте и соответствуют ли размеры поля ввода поисковым запросам, которые пользователи используют чаще всего?

– Насколько быстро пользователь получает ответы на свои вопросы, используя поиск?

– Насколько релевантны результаты поиска поисковому запросу пользователя? Если «да», то упорядочиваются ли они с уменьшением релевантности?

– Помогает ли поисковый робот справиться с ошибками при вводе текста?

При проведении юзабилити-тестирования сайтов необходимо помнить, что тестируются возможности сайта, а не способности пользователей. Следует обращать внимание на то, как пользователи выполняют задания, а не как они оценивают сайт, чему отдают предпочтения. Результаты юзабилити-тестирования являются основой для рекомендаций по улучшению сайта.

Необходимо искать наилучшее решение проблем, отмеченных разными пользователями. Для многих людей термин «тестирование» ассоциируется с проверкой их знаний, что очень часто негативно ими воспринимается. Нужно убедить пользователей в том, что тестируются не их способности, а возможности сайта, объяснить, что они помогают найти все преимущества и недостатки. Когда у пользователей возникают трудности с выполнением задания, необходимо фиксировать это как проблему на сайте, а не ошибку пользователя.

Юзабилити-тестирование – это не конечный этап проектирования сайта, оно не заканчивается, когда расходятся его участники. Команда, которая занимается анализом, должна обсудить полученные данные, выделить преимущества и составить рекомендации по изменению прототипа или основы сайта в соответствии с тем, что было выявлено в процессе тестирования.

Юзабилити-тестирование рекомендуется проводить:

1. Перед началом разработки нового дизайна. В прежнем дизайне следует выявить хорошие моменты, которые надо оставить или усилить, и исключить плохие моменты, которые вызывают затруднения у пользователей;

2. Исследуйте дизайн ваших конкурентов, чтобы недорого получить данные о возможных вариантах интерфейса;

3. Проведите тестирование в реальных условиях, чтобы оценить, как работают с интерфейсом пользователи;

4. Создайте бумажные прототипы одного или нескольких вариантов дизайна и испытайте их на пользователях. Чем меньше времени потратите на воплощение той или иной идеи, тем лучше: все равно придется менять дизайн по результатам тестов;

5. Отшлифуйте идеи, которые дают лучшие результаты, в нескольких тестах. Постепенно продвигайтесь от схематичного дизайна на бумаге к более подробному и детальному – на компьютере. Проводите тесты на каждом этапе внесения изменений;

6. Проверьте дизайн на соответствие общепринятым правилам юзабилити;

7. После того, как Вы примете решение по окончательному варианту и реализуете его, протестируйте его еще раз. На этапе реализации всегда закрадываются мелкие ошибки.

Не откладывайте тестирование дизайна на пользователях до самого последнего момента. Если поступите так, большинство критических ошибок, обнаруженных во время теста, будет невозможно исправить. Многие из этих ошибок будут носить структурный характер, и для их исправления придется переделывать все заново.

Единственный способ получить качественный пользовательский интерфейс – начать тестирование с самого начала развития проекта и продолжать тесты на каждом этапе. Попытайтесь найти лучшее решение. При анализе любого проекта нужно учитывать разные способы работы пользователей с ним, различный опыт, цели и задачи. Большинству проектов приходится бороться с ограничениями по времени, бюджету и ресурсам. Баланс этих составляющих является одной из важных проблем. Только взвесив все ограничения и найдя компромиссы, можно разработать веб-проект или приложение, которое будет соответствовать требованиям большинства пользователей. Рассмотрев поведе-

ние всех участников, сценарии и то, что было выделено в процессе юзабилити-тестирования, нужно найти идеальные решения проблем сайта в соответствии с требованиями и предпочтениями пользователей, которые принимали участие в тестировании.

После долгосрочного использования неудобного в работе продукта пользователи никогда не достигают тех же успехов, которых достигли бы, изначально работая с лучшим интерфейсом.

В результате исследований взаимодействия человека и компьютера американский программист Бен Шнейдерман составил набор правил, которые могут быть использованы при разработке многих типов интерфейсов. Эти принципы актуальны как для разработчиков интерфейсов, так и веб-дизайнеров:

1. Стремитесь к логичности.

2. Для опытных пользователей должен быть быстрый способ (сокращения, «горячие» клавиши, макросы).

3. Должна быть информативная обратная связь.

4. Диалог должен быть законченным.

5. Обработка ошибок должна быть простой.

6. Должен быть простой способ отмены действий.

7. Пользователь должен чувствовать, что все находится под его контролем.

8. Как можно меньше загружайте кратковременную память.

Группа участников тестирования должна соответствовать целевой аудитории веб-сайта, иначе исследование будет неэффективным. Напомним, что юзабилити- тестирование должно ответить на следующие вопросы:

– Понимают ли пользователи основное назначение сайта?

– Могут ли пользователи найти на сайте нужную информацию и услуги?

– Насколько простым они находят использование этих услуг?

– Насколько простым является для них заполнение форм (при регистрации, авторизации, подаче всевозможных запросов)?

– Достигаются ли цели, которые определили владельцы сайта?

В процессе юзабилити-тестирования перед пользователями должны быть поставлены реальные задачи. За действиями пользователей необходимо внимательно наблюдать, тщательно их документировать и анализировать. Наблюдение за пользователями при тестировании даст больше детальной информации, чем просто ответы на вопросы при анкетировании.

Юзабилити-тестирование с привлечением реальных пользователей может обнаружить, что некоторые части сайта пользователи находят сложными и непонятными. Предлагается оценить проблемы юзабилити сайта, исходя из следующих факторов:

– Насколько часто возникает эта проблема у разных пользователей?

– Влияние проблемы, если она возникает: сложна или проста она для пользователя?

– Как часто возникает данная проблема у одного и того же пользователя: он учится с первого раза решать эту проблему или она возникает всякий раз, когда он попадает в ту же ситуацию?

В ходе юзабилити-тестирования внимание надо обращать на действия пользователя, а не на его речь. Обычно имеется существенная разница между тем, что пользователь говорит, «что он хочет», и тем, что в действительности будет использовать. Единственный способ отличить одно от другого – тщательное юзабилити-тестирование.

В ходе тестирования необходимо сообщить пользователям и убедиться в том, что они это четко понимают, что тестируются возможности сайта, а не их способности. Пользователи должны понимать, что тестирование проводится с целью выяснения, насколько легко они могут выполнять те или иные задачи. Перед проведением юзабилити-тестирования нужно позаботиться о договоре неразглашения информации для защиты своей работы, а также чтобы испытуемые понимали, как именно будет использоваться и защищаться их персональная информация, полученная в ходе тестирования.

Оптимальное количество для тестирования: 6-9 пользователей в каждой группе. Если их больше 9-ти, то срабатывает закон убывающей приростной от-

дачи, то есть затрачиваемые усилия не будут оправдываться повышением точности результатов.

Существует множество методов изучения юзабилити, но самый основной и полезный – это метод тестирования дизайна пользователями. Он включает три компонента:

1) выберите пользователей, наиболее типичных для Вашего проекта/продукта, например, посетителей коммерческих сайтов или работников Вашей компании, часто пользующихся интернетом (в последнем случае должны быть выбраны работники, которые не работают в Вашем отделе);

2) попросите их выполнить наиболее типичные задачи;

3) следите за тем, что делают пользователи, где у них все получается, и где – возникают трудности с интерфейсом. Молчите и слушайте, о чем они говорят.

Самое важное – проводить тестирование отдельно с каждым пользователем. Пусть каждый сам решает вставшие перед ним проблемы. Если ему помогать или привлекать внимание к какому-то определенному элементу на экране, результаты теста окажутся недостоверными.

Для того чтобы обнаружить наиболее серьезные проблемы с юзабилити, вполне достаточно привлечь к тестированию пятерых пользователей. Лучше провести несколько кратких тестов и вносить изменения в дизайн после каждого из них: так можно сразу исправить обнаруженные ошибки. Этот метод последовательного дизайна позволяет улучшить качество конечного продукта. Чем больше окажется версий и идей, тем лучшим будет результат.

2. ПРАКТИЧЕСКИЙ РАЗДЕЛ

2.1. План практических занятий

Практическое занятие № 1. **Знакомство с уровнями опыта взаимодействия**

1. Цель работы:

1.1. Закрепить теоретические знания по разработке пользовательского интерфейса.

1.2. Получить практические навыки по проведению этапов предварительного и высокоуровневого проектирования интерфейса пользователя.

Теоретические сведения

Опыт взаимодействия условно можно разделить на пять уровней:

1) Уровень стратегии

Стратегические задачи у программных продуктов и у информационных пространств одни и те же. Потребности пользователей – это цели сайта, источник которых находится за границами нашей организации. Они определяются людьми, которые будут пользоваться нашим сайтом. Мы должны понимать, чего хочет от нас наша аудитория, и как эти пожелания согласуются с другими ее потребностями.

Противовесом пользовательским потребностям являются наши собственные цели. Эти цели сайта могут быть бизнес-целями («Заработать миллион долларов на интернет-продажах в этом году») или какими-либо иными («Информировать избирателей о кандидатах, участвующих в ближайших выборах»).

2) Уровень набора возможностей

На программной половине стратегия преобразуется в набор возможностей путем создания функциональной спецификации – подробного описания функциональных возможностей процесса.

На информационной половине возможности равнозначны требованиям к контенту – это описание различных элементов содержимого, которые необходимо создать.

3) Уровень структуры

На «программной» половине этого уровня возможности приобретают структуру благодаря проектированию взаимодействия, в процессе которого можно определить, как система будет вести себя в ответ на действия пользователей.

У информационных пространств структура определяется информационной архитектурой – организацией элементов содержимого в пределах информационного пространства.

4) Уровень компоновки

Уровень компоновки включает в себя три компонента. На обеих его половинах имеем дело с информационным дизайном – представлением информации в таком виде, который облегчает ее восприятие. У программных продуктов компоновка также включает в себя дизайн интерфейса, то есть организацию элементов интерфейса, позволяющую пользователям взаимодействовать с функциями системы.

Интерфейс для информационного пространства появляется в ходе разработки дизайна навигации, в результате чего будет набор экранных элементов, позволяющих пользователю перемещаться по информационной архитектуре.

5) Уровень поверхности

Наконец, мы поднялись на поверхность. Независимо от того, находится ли перед нами программный продукт или информационное пространство, имеем дело с визуальным дизайном – внешним видом конечного продукта. Этот элемент сложнее, чем может показаться на первый взгляд.

Задание

С помощью поисковой системы найдите какой-нибудь интернет-сайт (магазина или фирмы).

Походите по сайту и попытайтесь определить его цели, функциональные возможности пользователя, информационную структуру, компоновку. Оцените дизайн. Попытайтесь определить слабые (сильные) стороны данного сайта и

сделать вывод о его полезности (при необходимости ознакомьтесь с HTML-кодом страницы).

Оформите отчет. В отчет поместите краткое описание Вашего представления, как реализован каждый уровень, и снимки экранов сайта.

Практическое занятие № 2. Составление сценариев использования

1. Цель работы:

1.1. Закрепить теоретические знания по созданию сценария использования при разработке программного обеспечения.

1.2. Получить практические навыки по проведению этапов предварительного и высокоуровневого проектирования интерфейса пользователя.

Теоретические сведения

Сценарий использования, вариант использования, прецедент (англ. *Use Case*) – в разработке программного обеспечения и системном проектировании – это описание поведения системы, которым она отвечает на внешние запросы. Другими словами, сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой. Методика сценариев использования применяется для выявления требований к поведению системы, известных также как «функциональные требования».

«Каждый сценарий использования сосредоточивается на описании того, как достигнуть цели или задачи. Для большинства программных проектов это означает, что потребуется множество сценариев использования, чтобы определить необходимый набор свойств новой системы. Степень формальности программного проекта и его стадии будет влиять на необходимый уровень детализации, для каждого сценария использования».

Сценарий использования определяет взаимодействия между внешними агентами и системой, направленные на достижение цели. Актер (англ. *Actor*) представляет собой роль, которую играет человек или вещь, взаимодействуя с системой. Тот же самый человек, использующий систему, может быть представлен как различные актеры, потому что они играют различные роли. Напри-

мер, «Джек» может играть роль Клиента, использующего Банкомат, чтобы забрать наличные деньги, или играть роль Работника Банка, использующего систему для пополнения банкомата купюрами.

Шаблоны сценариев использования

Нет никакого стандартного шаблона для документации детальных сценариев использования. Существует много конкурирующих схем, но лучше всего использовать те шаблоны, которые лучше подходят для проекта. Есть, однако, смысл упомянуть об основных моментах, на которые стоит обратить внимание.

Имя сценария стоит писать в формате глагол-существительное (например, Заимствовать Книги, Забрать Наличные Деньги). Оно должно описывать достижимую цель (например, Регистрация Пользователя лучше, чем Регистрирующийся Пользователь) и объяснять, о чем сценарий использования.

Неплохо использовать как имя сценария цель актера, гарантируя таким образом внимание к потребностям пользователя. Два-три слова – оптимум. Если слов в названии больше, то обычно есть более короткое и более информативное имя.

Цель. Без цели сценарий бесполезен. Нет никакой необходимости в сценарии использования, когда нет потребности ни в каком актере, чтобы достигнуть цели. Цель кратко описывает то, чего пользователь намеревается достигнуть с этим сценарием.

Актеры. Актер (Актор) – это кто-то или что-то вне системы, влияющий на систему или находящийся под ее влиянием. Актер может быть человеком, устройством, другой системой или подсистемой, или временем. Человек в реальном мире может быть представлен несколькими актерами, если у них есть несколько различных ролей и целей по отношению к системе. Они взаимодействуют с системой и производят над ней некоторые действия.

Заинтересованные лица. Заинтересованное лицо – человек или отдел, которых затрагивает сценарий использования. Обычно это работники организации или отдела, для которых создается сценарий. К заинтересованному лицу

можно обратиться с просьбой предоставить информацию, отзыв или разрешение для сценария использования.

Предварительные условия. Стоит определить все условия, которые должны быть истиной (то есть описать состояние системы), при которых исполнение сценария имеет смысл. Таким образом, если система не находится в состоянии, описанном в предварительных условиях, поведение сценария неопределенно. Заметьте также, что предварительные условия – это не «активаторы».

Активаторы. Активатор – это событие, инициирующее выполнение сценария. Он может быть внешним, временным или внутренним. Если активатор – не реальное событие (например, клиент нажимает кнопку), но ряд сложных условий, тогда стоит описать процесс активации. Этот процесс должен периодически или постоянно проверять условия активации и инициировать сценарий.

Есть несколько вариантов, как описать ситуацию, когда активация происходит, но предварительные условия не удовлетворены.

Один путь состоит в том, чтобы обработать «ошибку» в пределах сценария (как исключение). В принципе, такой подход нелогичен, потому что «предварительные условия» – теперь не истинные предварительные условия вообще (потому что поведение сценария определено, даже когда предварительные условия не встречены).

Другой путь – поместить все предварительные условия в активатор (так, чтобы сценарий не выполнялся, если предварительные условия не встречены) и создать отдельный сценарий, описывающий проблему.

Порядок событий. Как минимум, каждый сценарий использования должен описать типичный ход событий, часто представленный как ряд пронумерованных шагов.

Альтернативные пути и дополнения. Сценарии использования могут содержать вторичные пути или альтернативные сценарии, которые являются вариантами основного. Каждое проверенное правило может привести к альтернативному пути, и когда правил много, количество альтернативных путей возрастает, приводя к очень сложным документам. Иногда лучше использовать ус-

ловную логику или диаграммы, чтобы описать сценарии со многими правилами и условиями.

Бизнес-правила определяют, как организация ведет свой бизнес относительно сценария использования. *Бизнес-правила* – специальный вид требований. Они могут касаться одного сценария использования, применяться ко всем сценариям или ко всему бизнесу. Сценарии использования должны ясно ссылаться на бизнес- правила, которые для них применимы и используются.

Задание

С помощью поисковой системы найдите какой-нибудь интернет-сайт (магазина или фирмы).

Походите по сайту и попытайтесь определить варианты использования.

Выберите какой-нибудь вариант и составьте сценарий использования в соответствии с приведенным шаблоном.

Оформите отчет. В отчет поместите название сайта, сценарий использования, снимки экрана, подтверждающие Ваш сценарий.

Практическое занятие № 3. Разработка пользовательского интерфейса: этапы предварительного и высокоуровневого проектирования

1. Цель работы:

1.1. Закрепить теоретические знания по разработке пользовательского интерфейса.

1.2. Получить практические навыки по проведению этапов предварительного и высокоуровневого проектирования интерфейса пользователя.

Теоретические сведения

На практике высокоуровневое проектирование пользовательского интерфейса предваряет первоначальное проектирование, которое позволяет выявить требуемую функциональность создаваемого приложения, а также особенности его потенциальных пользователей. Указанные сведения можно получить, анализируя информацию, поступающую от пользователей. С этой целью произво-

дят опрос целевой аудитории и формируют профили пользователей. Профилями называют описания главных категорий пользователей. Одна из таких категорий может быть принята за основной профиль. Следует отметить, что набор характеристик, подробно описывающий пользователя, зависит от предметной области и контекста решаемых им задач. Поэтому работа по определению целей и задач пользователей и работа по формированию их профилей ведется параллельно.

Наиболее общий шаблон профиля содержит в себе следующие разделы:

– социальные характеристики;

– навыки и умения работы с компьютером;

– мотивационно-целевая среда;

– рабочая среда;

– особенности взаимодействия с компьютером (специфические требования пользователей, необходимые информационные технологии и др.).

Профили пользователей могут по необходимости расширяться за счет добавления других (значимых с точки зрения проектировщика) характеристик пользователей.

После выделения одного или нескольких основных профилей пользователей и после определения целей и задач, стоящих перед ними, переходят к следующему этапу проектирования. Этот этап связан с составлением пользовательских сценариев. Как правило, начинают с персонификации профилей (присваивания каждому профилю условного имени), затем формулируют сценарии. Сценарий – это описание действий, выполняемых пользователем в рамках решения конкретной задачи на пути достижения его цели. Очевидно, что достигнуть некоторой цели можно, решая ряд задач. Каждую их них пользователь может решать несколькими способами, следовательно, должно быть сформировано несколько сценариев. Чем больше их будет, тем ниже вероятность того, что некоторые ключевые объекты и операции будут упущены.

В то же время у разработчика имеется информация, необходимая для формализации функциональности приложения. А после формирования сцена-

риев становится известным перечень отдельных функций. В приложении функция представлена функциональным блоком с соответствующей экранной формой (формами). Возможно, что несколько функций объединяются в один функциональный блок. Таким образом, на этом этапе устанавливается необходимое число экранных форм. Важно определить навигационные взаимосвязи функциональных блоков. На практике установлено: наиболее подходящим является число связей для одного блока, равное трем. Иногда, когда последовательность выполнения функций жестко определена, между соответствующими функциональными блоками можно установить процессуальную связь. В этом случае их экранные формы вызываются последовательно одна из другой. Такие случаи имеют место не всегда, поэтому навигационные связи формируются либо исходя из логики обработки данных, с которыми работает приложение, либо основываясь на представлениях пользователей (карточная сортировка). Навигационные связи между отдельными функциональными блоками отображаются на схеме навигационной системы. Возможности навигации в приложении передаются через различные навигационные элементы.

Основным навигационным элементом приложения является главное меню. Роль главного меню велика еще и потому, что оно осуществляет диалоговое взаимодействие в системе «пользователь – приложение». Кроме того, меню косвенно выполняет функцию обучения пользователя работе с приложением.

Формирование меню начинается с анализа функций приложения. Для этого в рамках каждой из них выделяют отдельные элементы: операции, выполняемые пользователями, и объекты, над которыми осуществляются эти операции. Следовательно, известно, какие функциональные блоки должны позволять пользователю осуществлять какие-то операции над какими-то объектами. Выделение операций и объектов удобно проводить на основе пользовательских сценариев и функционала приложения. Выделенные элементы группируются в общие разделы главного меню. Группировка отдельных элементов происходит в соответствии с представлениями об их логической связи. Таким образом, главное меню может иметь каскадные меню, выпадающие при выборе какого-

либо раздела. Каскадное меню ставит в соответствие первичному разделу список подразделов.

Одним из требований к меню является их стандартизация, целью которой выступает формирование устойчивой пользовательской модели работы с приложением. Существуют требования, выдвигаемые с позиций стандартизации, которые касаются места размещения заголовков разделов, содержания разделов, часто используемых в разных приложениях, формы заголовков, организации каскадных меню и др. Наиболее общие рекомендации стандартизации – следующие:

– группы функционально связанных разделов отделяют разделителями (черта или пустое место);

– не используют в названиях разделов фраз (желательно не больше 2-х слов);

– названия разделов начинают с заглавной буквы;

– названия разделов меню, связанных с вызовом диалоговых окон, заканчивают многоточием;

– названия разделов меню, к которым относятся каскадные меню, заканчивают стрелкой;

– используют клавиши быстрого доступа к отдельным разделам меню. Их выделяют подчеркиванием;

– допускается использовать «горячие клавиши», соответствующие комбинации клавиш отображают в заголовках разделов меню;

– допускают использовать включение в меню пиктограмм;

– измененным цветом показывают недоступность некоторых разделов меню в ходе работы с приложением;

– допускают делать недоступные разделы невидимыми.

Недоступность некоторых разделов меню обусловливается следующим. Главное меню является статическим и присутствует на экране в течение всего времени работы с приложением. Таким образом, при работе с разными экранными формами (взаимодействии с разными функциональными блоками) не все

разделы меню имеют смысл. Такие разделы принято называть недоступными. Поэтому. в зависимости от контекста решаемых пользователем задач (иногда – от контекста самого пользователя), главное меню приложения выглядит всегда по-разному. О подобных различающихся внешних представлениях меню принято говорить, как о различных состояниях меню. В отличие от схемы навигационной системы, составленной ранее и необходимой, пользователь входит в непосредственное взаимодействие с разработчиком. Поэтому следует составить граф состояния меню. Вершинами этого графа являются различные состояния меню (внешние представления одного и того же меню с доступными и недоступными разделами). Каждая вершина имеет пояснения о соответствии данного состояния меню отдельным экранным формам. Дуги графа состояний соответствуют операциям (командам меню), переводящим его из одного состояние в другое.

Подобный граф используют при формировании тестовых заданий на последних стадиях проектирования интерфейса. В связи с этим важно при его формировании выполнить проверку соответствия пользовательских сценариев возможным переходам по графу.

Задание

Выполнить этапы предварительного и высокоуровневого проектирования при разработке пользовательского интерфейса приложения для предметной области, представленной информационной системой, отображающую деятельность мелкой фирмы, которая связана с изготовлением и/или поставкой ряда товаров.

Разработать главное меню в среде разработки приложения с анализом и обоснованием его различных состояний для рабочего места менеджера по направлению товара.

Требования к оформлению отчета

Отчет должен содержать:

– название и цели работы;

– профиль пользователя с указанием целей и задач;

– описание функциональности приложения, указание отдельных функций, функциональных блоков, соответствующих им операций и объектов;

– схему навигационной системы с указанием связей между функциональными блоками;

– описание структуры главного меню приложения;

– граф состояний меню;

– выводы относительно соответствия возможных переходов по графу и пользовательских сценариев;

– общие выводы, сделанные в процессе выполнения лабораторной работы.

Практическое занятие № 4. Построение электронного прототипа интерфейса

Цель работы

1.1. Закрепить теоретические знания по разработке пользовательского интерфейса.

1.2. Развить навыки создания вариантов прототипов интерфейса пользователя.

1.3. Приобретение умений по формированию электронного прототипа – демонстрационного ролика интерфейса.

Теоретические сведения

Тестирование интерфейса является исключительно важной задачей при проектировании интерфейса. Начальный этап тестирования связан с разработкой прототипа интерфейса. На этом этапе проектировщик использует имеющиеся результаты проектирования: общую схему приложения, планы отдельных экранных форм, глоссарий. Эти результаты сводятся воедино в общую схему, которую необходимо проверить по сформулированным ранее сценариям. Целью такой проверки является выявление несоответствия последовательности действий, описанного в сценарии, и структуры полной схемы. Обнаруженные несоответствия должны быть устранены за счет модификации экранных форм и/или корректировки общей схемы приложения.

Имея полную схему приложения, приступают к формированию электронного прототипа. Следует отметить, что прототип должен, в первую очередь, отображать функциональность интерфейса результирующей системы, поэтому его первые версии делают достаточно «примитивными». Последующие версии прототипа могут быть эстетически более совершенными.

Электронный прототип пользовательского интерфейса представляет собой демонстрационный ролик, выполненный в одной из презентационных программ – MS PowerPoint, MS Visio и др. Каждая экранная форма соответствует отдельному слайду, результат нажатия кнопок имитируется переходами между слайдами. Переходы реализуются с помощью организации гиперссылок. Электронная версия прототипа пользовательского интерфейса позволяет тестировать довольно сложные взаимодействия человека с приложением.

Разработка пользовательского интерфейса приложения представляет собой итеративный процесс. Каждая итерация связана с отдельным этапом проектирования, созданием прототипа по его результатам, тестированием прототипа и его модификацией. Разработчик должен прилагать особые усилия, чтобы уменьшить число итераций.

Задание

Собрать полную схему приложения.

Выполнить проверку соответствия структуры полной схемы и последовательностей действий, описанных в пользовательских сценариях (практическая работа № 3).

При выявлении несоответствий внести коррективы в содержание экранных форм и/или схему навигации по приложению.

Выделить различные состояния отдельных экранных форм, в которых могут находиться формы в процессе взаимодействия пользователя с приложением.

Сформировать слайды для создания демонстрационного ролика. Каждый слайд соответствует определенному состоянию отдельной экранной формы.

Согласно полной схеме приложения, собрать демонстрационной ролик. Для организации переходов между слайдами использовать гиперссылки.

Требования к оформлению отчета. Отчет должен содержать:

– название и цели работы;

– полную схему приложения, скорректированную при необходимости.

Защита отчета сопровождается предъявлением готового демонстрационного ролика.

Практические занятия №№ 5-6. Создание и тестирование пользовательского интерфейса

Цель работы:

1. Закрепить теоретические знания по разработке пользовательского интерфейса.

2. Развить навыки создания вариантов прототипов интерфейса пользователя.

3. Закрепить навыки по формированию электронного прототипа – демонстрационного ролика интерфейса.

4. Получить практические навыки по количественной оценке интерфейса на этапе низкоуровневого проектирования.

5. Закрепить принципы обоснования выбора прототипа интерфейса по его количественной оценке.

Теоретические сведения

Высокоуровневое проектирование интерфейса.

На практике высокоуровневое проектирование пользовательского интерфейса предваряет первоначальное проектирование, которое позволяет выявить требуемую функциональность создаваемого приложения, а также особенности его потенциальных пользователей. Указанные сведения можно получить, анализируя информацию, поступающую от пользователей. С этой целью производят опрос целевой аудитории и формируют профили пользователей. Профилями называют описания главных категорий пользователей. Одна из таких категорий может быть принята за основной профиль. Следует отметить, что набор характеристик, подробно описывающих пользователя, зависит от предметной

области и контекста решаемых им задач. Поэтому работа по определению целей и задач пользователей и работа по формированию их профилей ведется параллельно.

Наиболее общий шаблон профиля содержит следующие разделы:

– социальные характеристики;

– навыки и умения работы с компьютером;

– мотивационно-целевая среда;

– рабочая среда;

– особенности взаимодействия с компьютером (специфические требования пользователей, необходимые информационные технологии и др.).

Профили пользователей могут по необходимости расширяться за счет добавления других (значимых с точки зрения проектировщика) характеристик пользователей.

После выделения одного или нескольких основных профилей пользователей и после определения целей и задач, стоящих перед ними, переходят к следующему этапу проектирования. Этот этап связан с составлением пользовательских сценариев. Как правило, начинают с персонификации профилей (присваивания каждому профилю условного имени), затем формулируют сценарии. Сценарий – это описание действий, выполняемых пользователем в рамках решения конкретной задачи на пути достижения его цели. Очевидно, что достигнуть некоторой цели можно, решая ряд задач. Каждую их них пользователь может решать несколькими способами, следовательно, должно быть сформировано несколько сценариев. Чем больше их будет, тем ниже вероятность того, что некоторые ключевые объекты и операции будут упущены.

В то же время у разработчика имеется информация, необходимая для формализации функциональности приложения. А после формирования сценариев становится известным перечень отдельных функций. В приложении функция представлена функциональным блоком с соответствующей экранной формой (формами). Возможно, что несколько функций объединяются в один функциональный блок. Таким образом, на этом этапе устанавливается необходимое

число экранных форм. Важно определить навигационные взаимосвязи функциональных блоков. На практике установлено, что наиболее подходящим числом связей для одного блока является число, равное трем. Иногда, когда последовательность выполнения функций жестко определена, между соответствующими функциональными блоками можно установить процессуальную связь. В этом случае их экранные формы вызываются последовательно одна из другой. Такие случаи имеют место не всегда, поэтому навигационные связи формируются либо, исходя из логики обработки данных, с которыми работает приложение, либо основываясь на представлениях пользователей (карточная сортировка). Навигационные связи между отдельными функциональными блоками отображаются на схеме навигационной системы. Возможности навигации в приложении передаются через различные навигационные элементы.

Основным навигационным элементом приложения является главное меню. Роль главного меню велика еще и потому, что оно осуществляет диалоговое взаимодействие в системе «пользователь-приложение». Кроме того, меню косвенно выполняет функцию обучения пользователя работе с приложением.

Формирование меню начинается с анализа функций приложения. Для этого в рамках каждой из них выделяют отдельные элементы: операции, выполняемые пользователями, и объекты, над которыми осуществляются эти операции. Следовательно, известно, какие функциональные блоки должны позволять пользователю осуществлять какие операции над какими объектами. Выделение операций и объектов удобно проводить на основе пользовательских сценариев и функционала приложения. Выделенные элементы группируются в общие разделы главного меню. Группировка отдельных элементов происходит в соответствии с представлениями об их логической связи. Таким образом, главное меню может иметь каскадные меню, выпадающие при выборе какого-либо раздела. Каскадное меню ставит в соответствие первичному разделу список подразделов.

Одним из требований к меню является их стандартизация, целью которой выступает формирование устойчивой пользовательской модели работы с при-

ложением. Существуют требования, выдвигаемые с позиций стандартизации, которые касаются места размещения заголовков разделов, содержания разделов, часто используемых в разных приложениях, формы заголовков, организации каскадных меню и др. Наиболее общие рекомендации стандартизации – следующие:

– группы функционально связанных разделов отделяют разделителями (черта или пустое место);

– не используют в названиях разделов фраз (желательно – не больше 2-х слов);

– названия разделов начинают с заглавной буквы;

– названия разделов меню, связанных с вызовом диалоговых окон, заканчивают многоточием;

– названия разделов меню, к которым относятся каскадные меню, заканчивают стрелкой;

– используют клавиши быстрого доступа к отдельным разделам меню. Их выделяют подчеркиванием;

– допускается использование «горячих клавиш», соответствующие комбинации клавиш отображают в заголовках разделов меню;

– допускают использовать включение в меню пиктограмм;

– измененным цветом показывают недоступность некоторых разделов меню в ходе работы с приложением;

– допускают делать недоступные разделы невидимыми.

Недоступность некоторых разделов меню обусловливается следующим. Главное меню является статическим и присутствует на экране в течение всего времени работы с приложением. Таким образом, при работе с разными экранными формами (взаимодействии с разными функциональными блоками) не все разделы меню имеют смысл. Такие разделы принято называть недоступными. Поэтому в зависимости от контекста решаемых пользователем задач (иногда – от контекста самого пользователя) главное меню приложения выглядит поразному. О подобных различающихся внешних представлениях меню принято

говорить, как о различных состояниях меню. В отличие от схемы навигационной системы, составленной ранее и необходимой в основном разработчику, пользователь входит в непосредственное взаимодействие с меню. Поэтому следует составить граф состояния меню. Вершинами этого графа являются различные состояния меню (внешние представления одного и того же меню с доступными и недоступными разделами). Каждая вершина имеет пояснения о соответствии данного состояния меню отдельным экранным формам. Дуги графа состояний соответствуют операциям (командам меню), переводящим его из одного состояния в другое.

Подобный граф используют при формировании тестовых заданий на последних стадиях проектирования интерфейса. В связи с этим важно при его формировании выполнить проверку соответствия пользовательских сценариев возможным переходам по графу.

Тестирование интерфейса является исключительно важной задачей при проектировании интерфейса. Начальный этап тестирования связан с разработкой прототипа интерфейса. На этом этапе проектировщик использует имеющиеся результаты проектирования: общую схему приложения, планы отдельных экранных форм, глоссарий. Эти результаты сводятся воедино в общую схему, которую необходимо проверить по сформулированным ранее сценариям. Целью такой проверки является выявление несоответствия последовательности действий, описанного в сценарии, и структуры полной схемы. Обнаруженные несоответствия должны быть устранены за счет модификации экранных форм и/или корректировки общей схемы приложения.

Имея полную схему приложения, приступают к формированию электронного прототипа. Следует отметить, что прототип должен, в первую очередь, отображать функциональность интерфейса результирующей системы, поэтому его первые версии делают достаточно «примитивными». Последующие версии прототипа могут быть эстетически более совершенными.

Электронный прототип пользовательского интерфейса представляет собой демонстрационный ролик, выполненный в одной из презентационных про-

грамм – MS PowerPoint, MS Visio и др. Каждая экранная форма соответствует отдельному слайду, результат нажатия кнопок имитируется переходами между слайдами. Переходы реализуются с помощью организации гиперссылок. Электронная версия прототипа пользовательского интерфейса позволяет тестировать довольно сложные взаимодействия человека с приложением.

Разработка пользовательского интерфейса приложения представляет собой итеративный процесс. Каждая итерация связана с отдельным этапом проектирования, созданием прототипа по его результатам, тестированием прототипа и его модификацией. Разработчик должен прилагать особые усилия, чтобы уменьшить число итераций.

Анализ качества интерфейса

Для анализа качества интерфейсов используется множество количественных и эвристических методов. Одним из лучших подходов к количественному анализу моделей интерфейсов является классическая модель GOMS (goals, objects, methods and selection rules).

Задание

Выполнить высокоуровневое проектирование интерфейса сайта «фотоальбомов» для одного из типов пользователей.

Цели сайта – проектируемый сайт позволяет пользователю-владельцу создавать коллекции фотографий и размещать их для публичного просмотра в сети. Другие пользователи просматривают коллекции.

Функциональные возможности

I. Владелец, который имеет следующие возможности:

1) создавать коллекции (альбомы) фотографий;

2) изменять содержимое коллекций;

3) разрешать (запрещать) публикацию коллекции для общего просмотра;

4) удалять коллекции;

5) вводить и изменять идентификационные данные (имя и пароль);

6) вводить (изменять, удалять) описание для коллекции и фотографии;

II. Пользователь интернета, который имеет следующие возможности:

1) знакомиться со списком коллекций, разрешенных к просмотру;

2) открывать (просматривать) коллекции и отдельные фотографии;

3) искать коллекции (отдельные фотографии) по описанию.

Собрать полную схему сайта

Выполнить проверку соответствия структуры полной схемы и последовательностей действий, описанных в пользовательских сценариях (практическая работа № 3). При выявлении несоответствий внести коррективы в содержание экранных форм и/или схему навигации по приложению.

Выделить различные состояния отдельных экранных форм, в которых могут находиться формы в процессе взаимодействия пользователя с приложением.

Сформировать слайды для создания демонстрационного ролика. Каждый слайд соответствует определенному состоянию отдельной экранной формы.

Согласно полной схеме приложения собрать демонстрационной ролик. Для организации переходов между слайдами использовать гиперссылки. Разработать демонстрационный ролик, содержащий вариант компоновки и выполнить количественную оценку его эффективности на основе метода GOMS.

Для каждого из экранных сценариев выполнить количественную оценку модели интерфейса на основе метода GOMS.

Предложить элементы дизайна поверхности (цветовые схемы, метафоры и т.д.)

Требование к отчету

Отчет должен содержать:

– краткие сценарии использования;

– диаграмму переходов;

– варианты компоновки;

– прототипы экранных форм и количественную оценку.

Практическое занятие № 7. **Юзабилити-тестирование**

1. Цель работы:

1. Закрепить теоретические знания по разработке пользовательского интерфейса.

2. Получить практические навыки по проведению юзабилити-тестирования.

Теоретические сведения

Юзабилити-тестирование – это эксперимент, выполняемый с целью определения того, насколько хорошо люди могут использовать некий искусственный объект для его предполагаемого применения, то есть юзабилити-тестирование измеряет юзабилити объекта. Юзабилити-тестирование – это метод оценки удобства использования продукта, основанный на привлечении пользователей в качестве тестирующих.

В ходе эксперимента участники выполняют серию заранее разработанных заданий. За выполнением заданий следят наблюдатели, которые отмечают, с какими сложностями сталкивается каждый из участников.

Юзабилити-тестирование сайтов позволяет выявить, отвечает ли сайт поставленным целям, а также поможет дать ответы на следующие вопросы:

Удачно ли пользователи решают поставленные перед ними задания?

Учитывая удачно завершенные задания, насколько быстро было выполнено каждое из них?

Учитывая удачно завершенные задания, сколько страниц (кликов) было использовано пользователями для выполнения каждого из них?

Насколько удовлетворены пользователи сайтом?

Какие изменения нужно внести, чтобы сделать сайт более успешным для еще большего количества пользователей?

Исследование и оценка сайтов может проводится разными методами, разработанными экспертами по юзабилити. Общим для всех методов является постановка реальных задач перед пользователями, а также фиксирование результатов тестирования для дальнейшего анализа. Для участия в юзабилити-

тестировании отбираются пользователи, соответствующие целевой аудитории сайта, они также не должны быть слишком знакомы с разработкой сайта, так как быстрее других смогут разобраться с его устройством, а это может создать иллюзию, что сайт понятен и удобен для целевых посетителей. Для того чтобы выявить и оценить наибольшее количество присутствующих на сайте проблем, необходимо привлекать реальных пользователей.

Проблемы юзабилити Якоб Нильсен предлагает оценивать по следующим факторам:

– частота возникновения той или иной проблемы у разных пользователей;

– влияние проблемы на пользователей: сложна она или проста в решении;

– частота возникновения данной проблемы у одного пользователя;

– получается ли у пользователя решить проблему с первого раза или он сталкивается с ней каждый раз, когда она возникает.

Для того чтобы провести качественное юзабилити-тестирование, у Вас должен быть подготовлен список специфических вопросов.

Например, если Вы хотите протестировать, насколько удобна функция поиска информации на Вашем сайте, Вы должны сконцентрировать свое внимание на ответах на следующие вопросы:

Используют ли пользователи поиск или просто кликают по страницам?

Преимущественно какие слова и фразы они используют для поиска?

Расположена ли строка поиска в удачном месте и соответствуют ли размеры поля ввода поисковым запросам, которые пользователи используют чаще всего?

Насколько быстро пользователь получает ответы на свои вопросы, используя поиск?

Насколько релевантны результаты поиска поисковому запросу пользователя? Если «да», то упорядочиваются ли они с уменьшением релевантности?

Помогает ли поисковый робот справится с ошибками при вводе текста?

Что нужно помнить при проведении юзабилити-тестирования сайтов?

Вы тестируете возможности сайта, а не способности пользователей.

Обращать внимание на то, как пользователи выполняют задания, а не на то, как они оценивают сайт, чему отдают предпочтения.

Результаты юзабилити-тестирования будут являться основой для рекомендаций по улучшению сайта.

Необходимо искать наилучшее решение проблем, выделенных разными пользователями.

Вы тестируете возможности сайта, а не способности пользователей.

Для многих людей термин «тестирование» ассоциируется с проверкой их знаний, что очень часто негативно ими воспринимается. Нужно убедить пользователей в том, что Вы тестируете не их способности, а возможности сайта, объяснить им, что они помогают найти его преимущества и недостатки. Когда у пользователей возникают трудности с выполнением задания, необходимо фиксировать это как проблему на сайте, а не как ошибку пользователя.

Нужно обращать внимание на то, как пользователи выполняют те или иные задания, а не на то, как они его оценивают и чему отдают предпочтения.

Можно измерить и показатели производительности, и предпочтения пользователей. Производительность определяется успехом выполнения задания, ошибками, временем выполнения и т.д. Предпочтения же – отчетом пользователей об их удовлетворении и комфорте при использовании сайта. Многие дизайнеры уверены, что, проектируя сайт, основываясь только на предпочтения некоторых пользователей, они делают его совершенным для всех. Это не доказано. На самом деле предпочтения и производительность – совсем не равноценные понятия. В результате проведения одного исследования было обнаружено, что у около 70% пользователей показатели производительности и оценки согласованы между собой. То есть они успешно выполняют задания, и им нравится сайт, или же неудачно выполняют, и сайт им тоже не очень нравится. Тем не менее, остается достаточно большой процент людей (30%), для которых показатели производительности и оценки не согласовываются. Они успешно выполняют задания, но сайт им не нравится, или же выполняют поставленные перед ними задачи не очень удачно, но, тем не менее, сайт им нравится.

Есть много причин, почему люди оценивают сайт выше, чем ожидается, если судить по тому, как они выполняют поставленные перед ними задания. Часто люди считают, что в тех проблемах, которые возникают, больше виноваты они, а не сайт. Они могут также думать, что оскорбят чувства владельцев, если поставят сайту низкую оценку. Они иногда не замечают проблем и думают, что у них все получается, хотя на самом деле это не так.

При проведении юзабилити-тестирования нужно обращать внимание на то, как пользователи выполняют те или иные задания, а не на то, какую они дают оценку.

Необходимо использовать результаты юзабилити-тестирования для разработки рекомендаций по дальнейшему продвижению ресурса.

Юзабилити-тестирование – это не конечный этап в плане по проектированию сайта, оно не заканчивается тогда, когда расходятся его участники. Команда, которая занимается анализом, должна обсудить полученные данные, выделить преимущества и составить рекомендации по изменению прототипа или основы сайта в соответствии с тем, что было выявлено в процессе тестирования.

Попытайтесь найти лучшее решение.

При анализе любого проекта нужно учитывать разные способы работы пользователей с ним, разный опыт, цели и задачи. Большинству проектов приходится бороться с ограничениями по времени, бюджету и ресурсам. Баланс этих составляющих является одной из важных проблем. Только взвесив все ограничения и найдя компромиссы, можно разработать веб-проект или приложение, которое будет успешно соответствовать требованиям большинства пользователей. Исследование показывает, что стоимость технической поддержки пользователей после запуска проекта гораздо выше, чем стоимость его отладки до запуска. Рассмотрев поведение всех участников, сценарии и то, что было выделено в процессе юзабилити-тестирования, нужно найти идеальные решения проблем сайта в соответствии с требованиями и предпочтениями пользователей, которые принимали участие в тестировании.

Было доказано, что после долгосрочного использования неудобного в работе продукта пользователи никогда не достигают тех же успехов, которых достигли бы, изначально работая с лучшим интерфейсом.

Задание

Вариант 1 (для группы из 2-4-х человек – предпочтительный)

Каждый участник группы определяет товар, который он предложит купить (выполнить действия до помещения в корзину включительно) тестируемым участникам. С помощью поисковой системы найдите какой-нибудь интернет- магазин, имеющий в наличие все товары.

Каждый участник группы по очереди выступает в роли тестируемого и пытается купить выбранный товар, остальные играют роль тестера и фиксируют (независимо друг от друга):

– возникшие проблемы;

– частоту возникновения той или иной проблемы у разных пользователей;

– влияние проблемы на пользователей: сложна она или проста в решении;

– частоту возникновения данной проблемы у одного пользователя;

– получается ли у пользователя решить проблему с первого раза или он сталкивается с ней каждый раз, когда она возникает;

– оценку пользователя сайта.

Составить отчет (можно совместный). В отчет поместить результаты, полученные в п. 2 и сделать выводы по удобству интерфейса.

Вариант 2 (индивидуальный)

Выберите два интернет-магазина, предлагающих аналогичный (желательно одинаковый) товар. Попытайтесь сделать покупку (до пункта поместить в корзину включительно) в каждом магазине. Оцените количество операций, которое Вы выполните в каждом случае.

Создайте отчет. В него поместите сравнительную таблицу, содержащую следующую информацию:

Параметры сравнения	Сайт	
	Сайт 1	Сайт 2
1. Личные предпочтения (+|–)		
2. Количество экранов		
3.1. Количество действий на экране 1		
…	…	…
3.N Количество действий на экране N		
Общее количество действий		
4.1. Оценка времени по методике GOMS1 для экрана 1		
…	…	…
4.N Оценка времени по методике GOMS для экрана N		
Примерное количество ошибок		

Сравните результаты количественных характеристик со своими впечатлениями.

3. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

3.1. Задания для самостоятельной работы студентов

3.1.1. Вопросы для самоподготовки и создания конспекта по теме № 1

1. Что изучает эргономика?

2. Какие виды совместимости среды «человек-машина» вы знаете?

3. Что понимается под человеко-компьютерным взаимодействием?

4. Что является основной задачей человеко-компьютерного взаимодействия?

5. Какие аспекты включает в себя узел взаимодействия?

6. Что такое интерфейс? Приведите примеры.

7. Что понимается под термином «юзабилити»?

8. Назовите пять составляющих юзабилити.

9. В чем разница понятий «эргономичность» и «юзабилити»?

10. Каковы особенности человеко-ориентированного интерфейса?

11. Что такое интерфейс web-сайта?

12. Что включает в себя понятие «опыт взаимодействия»?

13. Что понимается под информационной архитектурой?

3.1.2. Вопросы для самоподготовки и создания конспекта по теме № 2

1. Опишите процесс взаимодействия человека с компьютерной системой.

2. Перечислите способы удержания внимания на текущем объекте.

3. Объясните значение привычки.

4. Каким должен быть интерфейс, чтобы он способствовал формированию привычки?

5. Что такое фокус внимания?

6. Назовите способы удержания фокуса внимания.

7. Какие функции должен обеспечивать пользовательский интерфейс?

8. Что требуется от справочного механизма системы?

9. Из каких частей состоит интерфейс пользователя?

10. Какие стили графического пользовательского интерфейса Вы знаете?

11. Перечислите основные элементы управления.

12. Каковы особенности графического пользовательского web-интерфейса?

13. Назовите компоненты графического пользовательского web-интерфейса.

14. Что такое объектно-ориентированный пользовательский интерфейс?

3.1.3. Вопросы для самоподготовки и создания конспекта по теме № 3

1. Какие существуют подходы к проектированию пользовательских интерфейсов?

2. Какие этапы проектирования интерфейса можно выделить?

3. Что понимается под проектированием взаимодействия?

4. Какие уровни разработки взаимодействия можно выделить?

5. В чем смысл разработки опыта взаимодействия?

3.1.4. Вопросы для самоподготовки и создания конспекта по теме № 4

1. Для чего нужны сценарии использования?

2. Какие виды сценариев использования можно выделить?

3. Что требуется от сценария использования?

4. Перечислите уровни детализации сценариев использования.

5. Какие основные элементы должны присутствовать в сценарии использования?

6. Перечислите основные правила написания сценариев использования.

7. Что такое альтернативный сценарий использования?

8. Что понимают под активатором?

9. Кто такой актер в сценариях использования?

10. Какие стили графического пользовательского интерфейса Вы знаете?

11. Какие способы записи сценариев использования Вы знаете?

3.1.5. Вопросы для самоподготовки и создания конспекта по теме № 5

1. Какие два вопроса надо задать перед созданием сайта?

2. Что понимают под целями сайта?

3. Что такое «идентичность бренда»?

4. Что понимают под «метриками успешности»?

5. Зачем нужны функциональные спецификации?

6. Чем отличаются функциональные спецификации от требования к контенту?

7. Зачем нужно ранжирование требований?

8. Какие подходы к проектированию информационной архитектуры Вы знаете?

9. Какие информационные структуры вам известны?

10. Как информационный дизайн соотносится с дизайном навигации и дизайном интерфейса?

11. Чем соглашение отличается от метафоры?

12. Какие задачи решает дизайн навигации?

13. Укажите виды навигации.

14. Что такое прототипы страниц?

15. Какие методы оценки визуального дизайна сайта Вы можете указать?

16. Укажите основной способ привлечения внимания в визуальном дизайне.

3.1.6. Вопросы для самоподготовки и создания конспекта по теме № 6

1. Перечислите четыре основных критерия качества любого интерфейса. Какие два вопроса надо задать перед созданием сайта?

2. Из каких составляющих складывается длительность восприятия исходной информации?

3. В чем суть понятия «непосредственное манипулирование»?

4. Как создать кнопку «бесконечного» размера?

5. Как уменьшить дистанцию до кнопки?

6. Опишите методику для выполнения количественного анализа интерфейсов.

7. Почему человек совершает ошибки при работе на компьютере?

8. Какие виды человеческих ошибок наиболее опасны?

9. Какие способы борьбы с ошибками вам известны?

10. Как увеличить скорость обучения пользователя, работающего с вашим сайтом?

11. Какие три составляющие обеспечивают «понятность»?

12. Какие виды обучающих материалов вы знаете?

13. Перечислите принципы, соблюдение которых способствует увеличению удовлетворенности пользователя.

3.1.7. Вопросы для самоподготовки и создания конспекта по теме № 7

1. Какие задачи выполняет юзабилити-тестирование?

2. Как оценить проблемы юзабилити?

3. На какие моменты надо обращать внимание при тестировании?

4. На каких этапах разработки рекомендуется проводить тестирование?

5. Какие категории пользователей нужно выбирать для проведения тестирования?

6. Сколько пользователей необходимо для тестирования?

3.2. Перечень вопросов к экзамену

1. Человеко-компьютерное взаимодействие. Понятие интерфейса. Понятие «юзабилити».

2. Usability в web-дизайне. Опыт взаимодействия. Понятие об информационной архитектуре.

3. Структура взаимодействия пользователя с компьютерной системой. Понятие человеко-ориентированного интерфейса.

4. Основные требования к пользовательскому интерфейсу. Фокус внимания. Способы удержания внимания пользователя.

5. Назначение и общие функции пользовательского интерфейса. Синтаксическая структура интерфейса.

6. Справочный механизм информационной системы. Структура и стили пользовательского интерфейса.

7. Особенности и компоненты web-интерфейса.

8. Объектно-ориентированные пользовательские интерфейсы. Основные элементы управления.

9. Подходы к проектированию пользовательских интерфейсов. Этапы проектирование интерфейсов.

10. Опыт взаимодействия. Ранжирование требований.

11. Уровни и опыт взаимодействия. Элементы опыта взаимодействия. Уровни: стратегии, набора возможностей, структуры, компоновки, поверхности. Применение элементов.

12. Понятие о сценарии использования. Цели сценария использования.

13. Виды сценариев использования. Шаблоны сценариев.

14. Элементы сценариев – цели, заинтересованные лица, актеры (акторы), активаторы, события, предварительные условия, порядок событий, альтернативные пути, бизнес правила. Нотация сценариев использования.

15. Уровень стратегии. Определение стратегии. Цели сайта.

16. Уровень набора возможностей. Определение набора возможностей. Функциональность и содержание.

17. Сбор требований. Функциональные спецификации. Требования к содержанию. Ранжирование требований.

18. Уровень структуры. Проектирование взаимодействия. Концептуальные модели. Обработка ошибок.

19. Информационная архитектура. Архитектурные решения. Организационные принципы.

20. Язык и метаданные. Роли в команде и процесс разработки информационной архитектуры.

21. Дизайн навигации. Глобальная, дополнительная и контекстная навигации.

22. Информационный дизайн. Прототипы страниц.

23. Определение компоновки. Соглашение и метафоры. Макетная сетка. Визуальное представление компоновки.

24. Оценка визуального дизайна. Контраст и единообразие. Внутренняя и внешняя согласованность визуального дизайна.

25. Цветовые палитры и типографика. Макеты и руководства по стилю.

26. Определение смысловых связей. Логическая связь, связь по представлению пользователей и процессуальная связь.

27. Предсказание скорости. Метод GOMS. Критерии оценки качества.

28. Скорость работы пользователей, количество человеческих ошибок, скорость обучения, субъективное удовлетворение пользователей.

29. Понятие о Usability-тестировании. Задачи и методы тестирования.

30. Наблюдение за пользователями. Организация тестирования и оценка действий пользователя.

4. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

4.1. Учебная программа

ЧАСТНОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ИНСТИТУТ СОВРЕМЕННЫХ ЗНАНИЙ ИМЕНИ А. М. ШИРОКОВА»

УТВЕРЖДАЮ

Ректор Института современных знаний имени А.М. Широкова

_____ А.Л. Капилов

01.07.2016
(дата утверждения)

Регистрационный № УД-02-184/уч.

ИНФОРМАЦИОННАЯ АРХИТЕКТУРА И USABILITY

Учебная программа учреждения высшего образования по учебной дисциплине для специальности 1-19 01 01 «Дизайн (по направлениям)», направление специальности 1-19 01 01-06 «Дизайн (виртуальной среды)»

2016 г.

Учебная программа составлена на основе образовательного стандарта ОСВО 1-19 01 01-2013 и учебного плана Института современных знаний имени А. М. Широкова по направлению специальности 1-19 01 01-06 «Дизайн (виртуальной среды)»

СОСТАВИТЕЛЬ:

Ю. Д. Васильева, кандидат технических наук, доцент кафедры высшей математики и информатики Института современных знаний имени А. М. Широкова

РЕЦЕНЗЕНТЫ:

Ю. В. Виланский, кандидат технических наук, доцент, ведущий инженер-программист Научно-производственного частного унитарного предприятия «Тетраэдр»;

Кафедра дизайна Института современных знаний имени А. М. Широкова.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой высшей математики и информатики Института современных знаний имени А. М. Широкова (протокол № 9 от 11 мая 2016 г.);

Научно-методическим советом Института современных знаний имени А. М. Широкова (протокол № 4 от 30 июня 2016 г.).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Изучение данной дисциплины позволит студентам приобрести базовые знания по разработке и оценке пользовательских интерфейсов.

Цель дисциплины – изучение теоретических основ построения пользовательских интерфейсов для компьютерных программ и приобретение соответствующих базовых практических навыков их создания и оценки.

Для достижения этой цели необходимо решение следующих учебных задач:

– ознакомление с основными принципами человеко-компьютерного взаимодействия;

– изучение основных функций пользовательских интерфейсов;

– усвоение знаний о правилах построения пользовательских интерфейсов;

– изучение архитектурных решений;

– оценка визуального и информационного дизайна

– ознакомление с основными принципами Usability-тестирования.

Дисциплина «Информационная архитектура и Usability» относится к циклу специальных дисциплин и опирается на знания и умения студентов, полученные при изучении таких дисциплин, как «Основы информационных технологий» и «Информационные технологии в дизайне».

Знания, полученные при изучении данной дисциплины, могут быть использованы при изучении дисциплин «Основы web-дизайна», «Эргономика информационной среды», «Дизайн-проектирование», а также при подготовке курсовых и дипломных проектов.

Освоение образовательной программы по учебной дисциплине «Информационная архитектура и Usability» должно обеспечить формирование следующих академических компетенций:

АК-1. Владеть базовыми научно-теоретическими знаниями в области художественных, научно-технических, общественных и практических задач профессиональной деятельности.

АК-2. Владеть методикой системного и сравнительного анализа, междисциплинарным подходом к решению проблем, находить решения на стыке разных дисциплин, связанных с теорией и практикой дизайна.

АК-3. Владеть исследовательскими навыками.

АК-4. Уметь работать самостоятельно.

АК-5. Быть способным к творческой, креативной работе.

АК-6. Владеть междисциплинарным подходом при решении проблем.

АК-7. Иметь навыки использования современных технических средств обработки информации.

АК-9. Уметь учиться, быть расположенным к постоянному повышению профессиональной квалификации.

Также студент должен приобрести следующие социально-личностные компетенции:

СЛК-2. Совершенствовать и развивать свой интеллектуальный и общекультурный уровень, повышать проектно-художественное мастерство.

СЛК-6. Быть способным к критике и самокритике.

После изучения учебной дисциплины студент должен владеть следующими профессиональными компетенциями и быть способным:

ПК-2. Осуществлять дизайн-проектирование с учетом соотношения смыслообразующих и формообразующих факторов (художественно-формальных, эргономических, инженерно-психологических, технологических, конструктивных, экологических, социально-культурных, экономических) в условиях как аналогового, так и безаналогового проектирования.

ПК-3. Формировать выразительное образное решение объекта проектирования на основе конкретного содержания.

ПК-4. Осуществлять прогностическое дизайн-проектирование с использованием инновационных технологий.

ПК-5. Осуществлять экспертную оценку уровня дизайнерского решения по основным смыслообразующим и формообразующим факторам.

ПК-6. Адаптироваться к изменению объекта профессиональной деятельности, как в пределах специализации, так и направление специальности.

ПК-7. Осуществлять развитие научно-теоретической и практической базы обеспечения дизайн-деятельности.

ПК-9. Собирать, анализировать и систематизировать профессиональный опыт в области дизайн-деятельности.

ПК-10. Выявлять общие закономерности функционирования и развития дизайн-деятельности на основе собранного фактологического материала.

ПК-11. Анализировать композиционные, конструктивные, технологические, эргономические и колористические решения продуктов дизайн-деятельности.

ПК-12. Анализировать результаты собственных дизайн-решений.

ПК-18. Уметь проектировать, организовывать, анализировать процесс педагогического взаимодействия при освоении профессиональных компетенций по направлению специальности.

В соответствии с учебными планами направления специальности «Дизайн (виртуальной среды)» данная дисциплина изучается в четвертом семестре, начиная с набора 2015 г., а для набора 2014 г. – в пятом семестре. Общее количество часов – 106, аудиторных часов – 50 (22 часа лекций и 28 часов практических занятий). На самостоятельную работу отводится 56 часов.

Форма получения высшего образования – дневная.

Текущая аттестация по дисциплине проводится в форме экзамена.

В результате изучения дисциплины студенты должны:

знать:

– основные элементы управления, используемые при создании пользовательских интерфейсов;

– основные принципы построения удобных пользовательских интерфейсов;

– принципы проектирования интерфейсов;

– методики оценки качества интерфейсов;

уметь:

– оценивать удобство пользовательского интерфейса;

– организовывать тестирование качества пользовательских интерфейсов.

При проведении занятий в компьютерном классе предполагаются следующие формы работы:

– демонстрационная, когда студенты слушают объяснения и наблюдают за экраном компьютера;

– фронтальная, когда студенты синхронно работают под руководством преподавателя;

– самостоятельная работа студентов над лабораторными и индивидуальными заданиями.

Студенты должны самостоятельно выполнять некоторый объем работы на компьютере и представить его преподавателю в течение семестра.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Особенности взаимодействия человека с компьютерными системами

Человеко-компьютерное взаимодействие. Понятие интерфейса. Понятие юзабилити. Usability в web-дизайне. Опыт взаимодействия. Понятие об информационной архитектуре.

Структура взаимодействия пользователя с компьютерной системой. Понятие человеко-ориентированного интерфейса. Основные требования к пользовательскому интерфейсу. Фокус внимания. Способы удержания внимания пользователя.

Тема 2. Интерфейс пользователя

Назначение и общие функции пользовательского интерфейса. Синтаксическая структура интерфейса. Справочный механизм информационной системы. Структура пользовательского интерфейса. Стили пользовательского интерфейса. Особенности web-интерфейса. Компоненты web-интерфейса. Объектно-ориентированные пользовательские интерфейсы. Основные элементы управления.

Тема 3. Информационная архитектура. Опыт взаимодействия

Подходы к проектированию пользовательских интерфейсов. Этапы проектирование интерфейсов. Опыт взаимодействия. Ранжирование требований. Уровни взаимодействия. Опыт взаимодействия. Элементы опыта взаимодействия. Уровень стратегии. Уровень набора возможностей. Уровень структуры. Уровень компоновки. Уровень поверхности. Применение элементов.

Тема 4. Сценарии использования

Понятие о сценарии использования. Цели сценария использования. Виды сценариев использования. Шаблоны сценариев. Элементы сценариев – цели, заинтересованные лица, актеры (акторы), активаторы, события, предварительные условия, порядок событий, альтернативные пути, бизнес- правила. Нотация сценариев использования.

Тема 5. Проектирование взаимодействия и информационная архитектура

Уровень стратегии. Определение стратегии. Цели сайта. Уровень набора возможностей. Определение набора возможностей. Функциональность и содержание. Сбор требований. Функциональные спецификации. Требования к содержанию. Ранжирование требований. Уровень структуры. Проектирование взаимодействия. Концептуальные модели. Обработка ошибок.

Информационная архитектура. Архитектурные решения. Организационные принципы. Язык и метаданные. Роли в команде и процесс разработки. Дизайн навигации. Глобальная, дополнительная и контекстная навигации. Информационный дизайн. Прототипы страниц. Определение компоновки. Соглаше-

ние и метафоры. Макетная сетка. Визуальное представление компоновки. Оценка визуального дизайна. Контраст и единообразие. Внутренняя и внешняя согласованность визуального дизайна. Цветовые палитры и типографика. Макеты и руководства по стилю.

Тема 6. Критерии оценки качества пользовательских интерфейсов

Определение смысловых связей. Логическая связь, связь по представлению пользователей и процессуальная связь. Предсказание скорости. Метод GOMS. Критерии оценки качества. Скорость работы пользователей, количество человеческих ошибок, скорость обучения, субъективное удовлетворение пользователей.

Тема 7. Тестирование пользовательских интерфейсов

Понятие о Usability-тестировании. Задачи тестирования. Методы тестирования. Наблюдение за пользователями. Организация тестирования. Оптимальное количество членов в тестируемой группе. Оценка действий пользователя.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Самостоятельная работа студентов (СРС)	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Особенности взаимодействия человека с компьютерными системами	4	4				8	Результаты выполнения заданий по теме практического занятия
2	Интерфейс пользователя	2	4				8	
3	Информационная архитектура. Опыт взаимодействия	4	4				8	
4	Сценарии использования	2	4				8	
5	Проектирование взаимодействия и информационная архитектура	4	4				8	
6	Критерии оценки качества пользовательских интерфейсов	2	4				8	
7	Тестирование пользовательских интерфейсов	4	4				8	
	Итого	**22**	**28**				**56**	

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

СПИСОК ЛИТЕРАТУРЫ

Основная

1. Виланский, Ю. В. Информационная архитектура и Usability : курс лекций для студентов направления специальности 1-19 01 01-06 «Дизайн (виртуальной среды)» [Электронный ресурс] / Ю. В. Виланский. – Минск : Институт современных знаний им. А. М. Широкова, 2013.

2. Дакетт, Дж. HTML и CSS. Разработка и дизайн веб-сайтов / Дж. Дакетт. – М. : Эксмо, 2013. – 480 с.

3. Папанек, Виктор. Дизайн для реального мира / Виктор Папанек. – М. : Издательство «Издатель Дмитрий Аронов», 2015. – 416 с.

Дополнительная

1. Баканов, А. С. Эргономика пользовательского интерфейса : от проектирования к моделированию человеко-компьютерного взаимодействия / А. С. Баканов, А. А. Обознов. – М. : Институт психологии РАН, 2011. – 176 с.

2. Головач, В. В., Дизайн пользовательского интерфейса. Искусство мыть слона [Электронный ресурс] / В. В. Головач. – 2010. – 97 с. – Режим доступа: http://uibook2.usethics.ru/uibookII.pdf – Дата доступа: 23.04.2016.

3. Калиновский, А. И. Юзабилити : Как сделать сайт удобным / А. И. Калиновский. – Минск : Новое знание, 2005. – 220 с.

4. Круг, С. Как сделать сайт удобным. Юзабилити по методу Стива Круга / С. Круг. – СПб. : Питер, 2010. – 208 с.

5. Купер, Алан. Алан Купер об интерфейсе. Основы проектирования взаимодействия / Алан Купер, Роберт Рейман, Дэвид Крони. – СПб. : Символ-Плюс, 2009. – 688 с.

6. Морвиль, Питер. Информационная архитектура в Интернете / Питер Морвиль, Луис Розенфельд. – СПб. : Символ-Плюс, 2010. – 608 с.

7. Скотт, Билл. Проектирование веб-интерфейсов / Билл Скотт, Тереза Нейл. – СПб. : Символ-Плюс, 2010. – 352 с.

Интернет-ресурсы

1. Сообщество специалистов по юзабилити, проектированию пользовательских интерфейсов и user experience Беларуси [Электронный ресурс]. – Режим доступа: http://usability.by – Дата доступа: 05.05.2016.

2. Уроки юзабилити : блог [Электронный ресурс]. – Режим доступа: http://usability-by.blogspot.com.by/search/label/ Уроки юзабилити – Дата доступа: 05.05.2016.

ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

1. Знакомство с уровнями опыта взаимодействия.

2. Составление сценариев использования.

3. Разработка пользовательского интерфейса. Предварительное и высокоуровневое проектирования.

4. Построение электронного прототипа интерфейса.

5. Создание пользовательского интерфейса.

6. Тестирование пользовательского интерфейса.

7. Usability-тестирование.

ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

№ п/п	Название раздела, темы	Кол-во часов на СРС	Задание	Форма выполнения	Цель или задача СРС
1	**2**	**3**	**4**	**5**	**6**
1	Особенности взаимодействия человека с компьютерными системами	8	Изучить предложенные вопросы по темам, используя литературные источники, ресурсы интернета и лекционные материалы. Выполнить самостоятельные задания по тематике практических занятий.	Краткий конспект терминов и понятий, отчеты в виде документов Word	Расширение знаний, полученных во время лекционных занятий; закрепление навыков, полученных на практических занятиях
2	Интерфейс пользователя	8			
3	Информационная архитектура. Опыт взаимодействия	8			
4	Сценарии использования	8			
5	Проектирование взаимодействия и информационная архитектура	8			
6	Критерии оценки качества пользовательских интерфейсов	8			
7	Тестирование пользовательских интерфейсов	8			

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы Института по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УВО

на _____ /_____ учебный год

№ пп	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры высшей математики и информатики (протокол № ____от _____201__г.)

Заведующий кафедрой

_____ _____ _____

(ученая степень, ученое звание) (подпись) (И.О.Фамилия)

УТВЕРЖДАЮ
Декан факультета

_____ _____ _____

(ученая степень, ученое звание) (подпись) (И.О.Фамилия)

4.2. Литература

Основная

1. Виланский, Ю. В. Информационная архитектура и Usability : курс лекций для студентов направления специальности 1-19 01 01-06 «Дизайн (виртуальной среды)» [Электронный ресурс] / Ю. В. Виланский. – Минск : Институт современных знаний им. А. М. Широкова, 2013.

2. Дакетт, Дж. HTML и CSS. Разработка и дизайн веб-сайтов / Дж. Дакетт. – М. : Эксмо, 2013. – 480 с.

3. Папанек, Виктор. Дизайн для реального мира / Виктор Папанек. – М. : Изд.-во «Издатель Дмитрий Аронов», 2015. – 416 с.

4. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Руководство по обеспечению пригодности использования : ГОСТ Р ИСО 9241-11-2010. – Ч. : 11. – М. : Стандартинформ, 2011.

Дополнительная

1. Баканов, А. С. Эргономика пользовательского интерфейса : от проектирования к моделированию человеко-компьютерного взаимодействия / А. С. Баканов, А. А. Обознов. – М. : Институт психологии РАН, 2011. – 176 с.

2. Головач, В. В., Дизайн пользовательского интерфейса. Искусство мыть слона [Электронный ресурс] / В. В. Головач. – 2010. – 97 с. – Режим доступа: http://uibook2.usethics.ru/uibookII.pdf – Дата доступа: 23.04.2016.

3. Калиновский, А. И. Юзабилити : Как сделать сайт удобным / А. И. Калиновский. – Минск : Новое знание, 2005. – 220 с.

4. Круг, С. Как сделать сайт удобным. Юзабилити по методу Стива Круга / С. Круг. – СПб. : Питер, 2010. – 208 с.

5. Купер, Алан. Алан Купер об интерфейсе. Основы проектирования взаимодействия / Алан Купер, Роберт Рейман, Дэвид Крони. – СПб. : Символ-Плюс, 2009. – 688 с.

6. Морвиль, Питер. Информационная архитектура в Интернете / Питер Морвиль, Луис Розенфельд. – СПб. : Символ-Плюс, 2010. – 608 с.

7. Скотт, Билл. Проектирование веб-интерфейсов / Билл Скотт, Тереза Нейл. – СПб. : Символ-Плюс, 2010. – 352 с.

8. Коберн, А. Современные методы описания функциональных требований к системе / А. Коберн. – М. : Лори, 2002. – 288 с.

9. Гарретт, Дж. Веб-дизайн : книга Джесса Гарретта. Элементы опыта взаимодействия. / Дж. Гаррет ; пер. с англ. – СПб. : Символ-Плюс, 2008. – 192 с.

10. Шнейдерман, Б. Психология программирования : Человеческие факторы в вычисл. и информ. системах / Б. Шнейдерман. – М. : Радио и связь, 1984. – 304 с.

11. Нильсен, Я., Веб-дизайн : анализ удобства использования веб-сайтов по движению глаз / Я. Нильсен, К. Перниче. – М. : Вильямс, 2010. – 480 с.

12. Нильсен, Я. Web-дизайн: удобство использования Web-сайтов / Якоб Нильсен, Хоа Лоранжер. – М. : Вильямс, 2007. – 368 с.

13. Нильсен, Я. Веб-дизайн / Якоб Нильсен. – СПб. : Символ-Плюс, 2003. – 512 с.

Интернет-ресурсы

1. Сообщество специалистов по юзабилити, проектированию пользовательских интерфейсов и user experience Беларуси [Электронный ресурс]. – Режим доступа: http://usability.by – Дата доступа: 05.05.2016.

2. Уроки юзабилити : блог [Электронный ресурс]. – Режим доступа: http://usability-by.blogspot.com.by/search/label/ Уроки юзабилити – Дата доступа: 05.05.2016.

СОДЕРЖАНИЕ

Учебное электронное издание

Автор-составитель
Слепцов Владимир Федорович

ИНФОРМАЦИОННАЯ АРХИТЕКТУРА И USABILITY

*Электронный учебно-методический комплекс
для студентов специальности 1-19 01 01 Дизайн (по направлениям),
направление специальности 1-19 01 01-06 Дизайн (виртуальной среды)*

[Электронный ресурс]

Редактор *И. Б. Михнюк*
Технический редактор *Ю. В. Хадьков*

Подписано в печать 29.12.2017.
Гарнитура Times Roman. Объем 0,8 Мб

Частное учреждение образования
«Институт современных знаний имени А. М. Широкова»
Свидетельство о регистрации издателя №1/29 от 19.08.2013
220114, г. Минск, ул. Филимонова, 69.

ISBN 978-985-547-194-4