

УДК 004.5

В. В. МАРТЫНОВ, А. М. КУЗНЕЦОВ

УЛУЧШЕНИЕ ЭРГОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВЕБ-ПРИЛОЖЕНИЙ

Статья посвящена оценке качества пользовательского интерфейса веб-приложения и его итеративному улучшению с учетом экономической эффективности. Рассматриваются наиболее общие проблемы, с которыми сталкиваются веб-дизайнеры и специалисты по веб-юзабилити, дается подход к их решению. *веб-приложение; эргономика; качество интерфейса*

Большинство программ, в том числе веб-сайтов, слишком сложны для использования. Этот вывод следует из результатов тестирования удобства использования программного обеспечения (ПО), практических наблюдений и личного опыта проектировщиков ПО. Такое направление информационных технологий, как эргономика пользовательских интерфейсов ПО, появилось в результате проведения исследований, основанных на предположениях о том, что делает существующие программы трудными для использования, а затем — предложением решений по преодолению возникших трудностей. Проектировщики программ, концепции эргономики ПО, должны полагаться на удовлетворение функций, возложенных на проектируемое ПО заказчиком, для окончательного успеха проекта.

Основные проблемы, осложняющие использование современных программных продуктов, таковы:

- **Пользователи не создают адекватную мысленную модель продукта.** Интерфейс большинства современных программных продуктов предполагает, что пользователи поймут концептуальную модель, тщательно выработанную разработчиками. Однако большинство пользователей вовсе не строят мысленной модели, которая была бы достаточно полной и точной для взаимодействия с программой. Пользователи слишком заняты и перегружены информацией. У них нет времени, энергии или желания интересоваться концептуальной моделью программы.

- **Даже опытные пользователи зачастую не в силах освоить часто используемые процедуры.** Дизайнеры знают, что новые пользователи испытывают трудности на этапе освоения программы, но ожидают, что эти пробле-

мы исчезнут по мере обучения. Данные исследований удобства использования показывают, что часто этого не происходит и что пользователи, сосредоточенные на своей задаче, не всегда уделяют внимание порядку выполняемых ими действий и не учатся на собственном опыте. В следующий раз, выполняя ту же операцию, они могут споткнуться на том же месте, что и в предыдущий раз.

- **Пользователи должны приложить немалые усилия, чтобы разобраться в каждой функции или экранной форме.** Большая часть программных продуктов спроектирована для небольшого количества пользователей, понимающих их концептуальную модель и разбирающихся в часто используемых процедурах. Для большинства клиентов каждая функция или процедура — раздражающая и нежелательная головоломка. Пользователи могут считать такие головоломки неизбежной платой за использование компьютеров, но, конечно, без этого им было бы куда лучше.

Методам устранения вышеперечисленных проблем посвящена данная статья.

ИССЛЕДОВАНИЕ ПРОЦЕССА ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ПРОГРАММОЙ

Прежде всего, следует проанализировать процесс взаимодействия пользователя с программным продуктом, а также понять, из чего строится пользовательский интерфейс ПО, и определить, каковы основные факторы, мешающие быстрому освоению и удобному использованию новых программных систем.

В общем случае процесс общения человека с машиной может быть схематично представлен четырьмя составляющими и направлением их взаимодействия [4] (рис. 1).

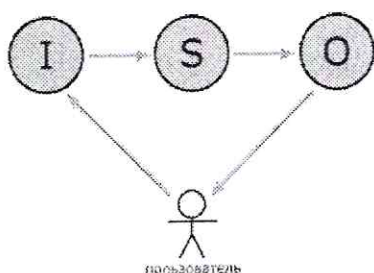


Рис. 1. Модель взаимодействия «человек–машина» с точки зрения пользователя

Первой составляющей является человек или пользователь. Он взаимодействует с системой путем ввода некоторых данных (подачи команд, осуществления манипуляций и т. п.). Объектом взаимодействия является входящая компонента пользовательского интерфейса (I) — набор управляющих элементов, с которыми пользователь непосредственно манипулирует для достижения цели [3, 4]. Ядро же системы (S), с точки зрения пользователя, выглядит как черный ящик. Оно представляет собой совокупность состояний, которые может принимать система с течением времени при взаимодействии с пользователем (или другими внешними объектами), либо без оных. Оценить текущее состояние системы и тем самым определить, решена ли текущая задача, пользователь способен с помощью исходящей компоненты пользовательского интерфейса (O). Она состоит из элементов-индикаторов, отражающих всю необходимую пользователю информацию о текущем состоянии системы.

Таким образом, для пользователя смысл системы состоит в том, что, пользуясь входящей компонентой интерфейса, он может подавать системе различные команды в том или ином виде, воздействуя на компонент состояний системы и оценивая эти состояния и полученный результат при помощи исходящей компоненты, в итоге приближаться к достижению требуемой цели.

Изначально создавая приложение, разработчик должен обеспечить достижение всех необходимых пользователю состояний системы, т. е. обеспечить доступ ко всем функциям системы для тех или иных категорий пользователей.

Достигается это посредством создания адекватного интерфейса пользователя. В 1983 году компания Rational разработала методологию проектирования ПО под названием Rational Unified Process (RUP). В настоящее время она является самым полным и надежным подходом к разработке программных систем. Помимо прочего, в RUP имеется мето-

дика проектирования интерфейса пользователя, основанная на модели так называемых прецедентов. Прецедент (“use case” — вариант использования) — это основа проектирования системы в RUP. Под прецедентом понимается последовательность действий, которые система предпринимает для получения пользователем требуемого результата. Иначе говоря, прецедент — это функциональный атом системы.

Разработка интерфейса пользователя, как и другие этапы разработки ПО, выполняется на основе определенных вариантов использования.

В модели прецедентов, разработанной системным аналитиком, должно быть определено, какие пользователи имеются у системы и для чего пользователи будут использовать систему [5]. С помощью этого создаются прототипы пользовательских интерфейсов, которые позволяют пользователю эффективно выполнять все прецеденты системы.

ПОКАЗАТЕЛЬ КАЧЕСТВА ИНТЕРФЕЙСА

Согласно Дональду Норману [1], выполнение прецедентов, т. е. взаимодействие пользователя с любой системой, состоит из семи шагов.

Модель действий Нормана:

1. Определение цели.
2. Определение взаимодействия.
3. Определение последовательности действий.
4. Осуществление взаимодействия.
5. Восприятие состояния системы.
6. Интерпретация состояния.
7. Оценка состояния относительно выполнения цели.

Из этого списка видно, что процесс размышления занимает почти все время, в течение которого пользователь работает с системой, во всяком случае шесть из семи этапов полностью заняты умственной деятельностью. Соответственно повышение скорости этих размышлений приводит к существенному улучшению скорости работы.

К сожалению, существенно повысить скорость собственно мышления пользователей невозможно. Тем не менее уменьшить влияние факторов, усложняющих (и, соответственно, замедляющих) процесс мышления, вполне возможно в процессе улучшения прототипа интерфейса.

В результате анализа существующих методик описания требований к интерфейсам,

приходим к выводу: пользователь манипулирует системой, выполняя варианты ее использования. Каждый раз, в процессе выполнения того или иного варианта, происходит взаимодействие пользователя с системой, которое можно разделить на этапы взаимодействия и в дальнейшем проводить улучшение прототипа интерфейса по этим этапам.

С учетом вышесказанного, а также исходя из специфики веб-приложений, для последних представляется возможным упростить модель Нормана до четырех составляющих:

- **стадия планирования.** Сочетает в себе как этап определения цели, так и этап определения взаимодействия. На этом этапе пользователь пытается понять, ЧТО ему нужно сделать для достижения цели, т.е. какими элементами управления воспользоваться;

- **стадия трансляции.** Содержит этап определения последовательности действий — пользователь пытается реализовать спланированные действия, т.е. КАК воспользоваться выбранными элементами управления;

- **стадия физического действия.** Этап осуществления взаимодействия, на котором пользователь физически манипулирует выбранным элементом управления и выбранным способом;

- **стадия оценки состояния.** Этапы восприятия состояния системы, ее интерпретации и оценки, когда пользователь, руководствуясь интерфейсом, пытается понять, достиг ли он своей цели, и если не достиг, возвращается к первой стадии, т.е. к стадии планирования очередного взаимодействия.

Различные свойства интерфейса проявляются на различных стадиях взаимодействия. Отсюда, эмпирически можем классифицировать свойства интерфейса веб-приложения.

Качество интерфейса можно оценить по трем основным критериям, к которым можно применить количественное измерение [6]:

- минимальное время достижения поставленных целей T ;
- минимальное число действий, совершенных пользователем, для достижения цели Q_a ;
- разница между ожидаемым и полученным в результате взаимодействия результатом ΔR .

Остальные критерии качества, например эмоциональное удовлетворение пользователя от работы с системой, можно либо считать производными от приведенных выше, либо

измерить их в каких-либо единицах невозможно, ввиду их субъективности.

Качество интерфейса, как и большинства других систем, напрямую зависит от качества его отдельных элементов. Выразим математически зависимость уровня юзабилити¹ интерфейса от показателей качества его отдельных управляющих элементов.

Качество элементов будем рассматривать по следующим параметрам:

- **Vis (visual)** — эффективность визуальной составляющей (под эффективностью здесь понимается степень выполнения поставленных перед элементами управления интерфейсом (ЭУ) задач с точки зрения данной составляющей). Сюда же включается понятие аффорданса, т.е. свойства объекта своим внешним видом пояснять, как им нужно пользоваться. Например, обычная «выпуклая» кнопка: по ее виду понятно, что на нее нужно «нажать». Зачастую значение аффорданса некоторых ЭУ может быть повышено за счет использования пиктограмм. Отсутствие раздражающих факторов во внешнем виде и поведении элемента также относится к эффективности визуальной составляющей;

- **Sem (semantic)** — эффективность семантической (смысловой) составляющей — это, например, степень того, насколько осмыслена надпись на кнопке, насколько она помогает понять ее назначение;

- **Arr (arrange)** — правильность взаимного расположения. Например, сходные по значению ЭУ должны быть расположены рядом, а более важные и часто используемые ЭУ должны быть расположены выше;

- **Qty (quantity)** — количество ЭУ данного типа;

- **Com (compatibility)** — уместность (совместимость) элемента в контексте реализации интерфейса в целом.

Необходимо разделить все множество ЭУ на несколько подгрупп. Обычно выделяют следующие подгруппы [8, 9]:

- **H (header)** заголовочный блок;
- **M (material)** материальное меню;
- **I (instrumental)** инструментальное меню;
- **S (service)** сервисные (функциональные) элементы;
- **C (contents)** информационное наполнение;
- **D (design)** элементы оформления;
- **A (auxiliary)** вспомогательные элементы.

¹Юзабилити — наличие комплекса свойств интерфейса, обеспечивающих комфортную работу пользователя с веб-приложением [11].

Качество всех интерфейсных элементов зависит от **субъективной оценки пользователем использованного дизайн-решения элемента** — $Vis(X)$, где X — элемент интерфейса. Под дизайн-решением мы будем понимать весь набор выразительных средств, который использует дизайнер для представления отдельных ЭУ:

- цветовое решение (сочетание цветов фона ЭУ и самого ЭУ);

- текстурное решение (в случае, когда ЭУ представляется не сплошным цветом, а для заполнения пространства элемента используется некая текстура: градиент, растровое изображение и т. п.);

- шрифтовое решение для выражения вербальной информации на ЭУ или рядом с ним (тип шрифта, его размер, начертание, расстояние между символами, расстояние между строками, растянутость символов, обводка вокруг символов);

- форма ЭУ.

Также качество всех групп элементов, кроме вспомогательных (A), напрямую зависит от степени сочетаемости данного элемента с реализацией всего интерфейса пользователя ($Com(X)$), где X — элемент интерфейса.

Каждому ЭУ в общей формуле расчета качества интерфейса ставится в соответствие вес этого элемента W_i , который выражает степень влияния последнего на общую картину качества.

Поскольку свойства эффективности визуальной составляющей и сочетаемости будут фигурировать в функции качества всех групп, для простоты определим их в виде нового свойства **общей визуальной эффективности CVE** (Common Visual Efficiency) и будем использовать в дальнейшем в формулах именно это обозначение:

$$CVE(X) = \frac{\sum_{i=1}^{Qty(X)} (W_i Vis(X_i)) + Com(X)}{Qty(X)}. \quad (1)$$

Выразим показатель качества Q каждой из групп:

Качество заголовка зависит от внешнего вида логотипа и смысла слогана [12], каждый из них умножается на соответствующий вес, определяющий важность данного элемента. Вес — нормированная величина, и вычисляется она на основе данных, полученных от экспертов.

$$Q_H = \frac{W_{logo} Vis(H_{logo}) + W_{slogan} Sem(H_{slogan})}{Qty(H)} + CVE(H). \quad (2)$$

Качество навигационных панелей материального и инструментального меню базируется на так называемом законе «семерки» [10], который гласит, что в соответствии с особенностями человеческого восприятия реальности число подобных элементов в простых системах должно быть примерно равно семи, для удобства освоения и использования этих простых систем. Для того, чтобы избежать деления на 0, в числитель и знаменатель добавляется по единице в качестве слагаемого. Также учитывается правильность расположения опций меню относительно друг друга.

$$Q_M = \frac{2Qty(M_{exist}) + Arr(M_{exist}) + 1}{|Qty(M_{exist}) - 7| + \sum_{i=1}^{Qty(M_{exist})} |Qty_i(M_{drop}) - 7| + 1} + CVE(M). \quad (3)$$

Здесь $Qty(M_{exist})$ — число элементов постоянно присутствующего на экране меню, $Qty(M_{drop})$ — число элементов выпадающего (англ. dropdown) меню.

Формула для инструментального меню строится аналогично, с той лишь особенностью, что в ней не присутствуют выпадающие меню:

$$Q_I = \frac{|Qty(I) - 7| + Arr(I)}{Qty(I)} + CVE(I). \quad (4)$$

Качество содержимого зависит в первую очередь от качества изложения текста:

$$Q_C = \frac{Sem(C)}{Qty(C)} + CVE(C). \quad (5)$$

Качество функциональных элементов вычисляется исходя как из правила семерки, так и правильности значений, выставленных по умолчанию и правильности подобранных типов ЭУ (кнопка, переключатель и т. п.) [2]:

$$Q_S = \frac{Sem(S_{defaults}) + Sem(S_{type}) + |Qty(S) - 7|}{Qty(S)} + CVE(S), \quad (6)$$

где $Sem(S_{defaults})$ — правильность выставленных значений по умолчанию, $Sem(S_{type})$ — правильность выбранного типа ЭУ.

Качество элементов оформления зависит как от визуальной эффективности, так и их функциональной насыщенности (как такие ЭУ помогают осваивать и использовать систему):

$$Q_D = \frac{\text{Sem}(D)}{\text{Qty}(S)} + \text{CVE}(D). \quad (7)$$

Качество вспомогательных элементов в общем случае обратно пропорционально их количеству в интерфейсе (перегруженность рекламными баннерами снижает эффективность интерфейса):

$$Q_A = \frac{1}{\text{Qty}(A)} + \text{CVE}(A). \quad (8)$$

Теперь, суммируя полученные показатели качества групп, можем определить общий показатель качества интерфейса:

$$Q = Q_H + Q_M + Q_I + \\ + Q_C + Q_S + Q_D + Q_A. \quad (9)$$

Таким образом, можно сделать вывод о том, что существует способ формализации такого понятия, как качество пользовательского интерфейса. Определив параметры качества и разложив ЭУ интерфейса по группам, мы получили выражения количественного показателя качества для каждой из групп. Сумма показателей всех групп в итоге дает показатель качества всего интерфейса в целом. Имея функцию качества, можно произвести ее улучшение для повышения показателя качества интерфейса.

ИТЕРАТИВНОЕ УЛУЧШЕНИЕ ИНТЕРФЕЙСА

Как было показано ранее, свойства интерфейса имеют иерархическую структуру. Основная цель их улучшения — отсортировать свойства интерфейса по приоритетам, т.е. определить наиболее критичные свойства тестируемого интерфейса, исправление которых наиболее важно для общего процесса улучшения юзабилити.

Для этого проведем декомпозицию проблемы на более простые составляющие части при помощи построения иерархии критериев оценки. В результате обработки суждений лиц, принимающих участие в тесте, определяется относительная значимость исследуемых альтернатив для всех критериев, находящихся в иерархии. Относительная значимость выражается численно в виде векторов

приоритетов. Полученные таким образом значения векторов являются оценками в шкале отношений и соответствуют так называемым жестким оценкам.

Воспользуемся методом сравнения относительно стандартов [7] для нахождения векторов приоритетов ЭУ.

На основе найденных векторов приоритетов появляется возможность отсортировать ЭУ интерфейса по значимости. При тестировании интерфейсов производится агрегирование мнений множества экспертов с учетом их разделения по уровню подготовки и опыту использования тестируемого продукта.

Юзабилити-тестирование представляет собой постановку экспериментов с целью выявления специфичной информации, касающейся удобства использования исследуемого продукта.

В разрабатываемом процессе улучшения юзабилити, где функция качества интерфейса представляет собой математическую модель процесса, юзабилити-тестирование позволяет реализовать эту модель. Подставив все необходимые параметры, найденные в результате тестирования, возможно получить значение функции качества интерфейса, организовав тем самым управление последним.

Тестовые задания строятся на базе тех задач, которые пользователям необходимо будет выполнять с применением данного веб-сайта. Задачи сайта определяются на основе спецификации к приложению, являющейся обязательным компонентом проектной документации и состоящей из прецедентов, преобразованных к требованиям вида «система должна ...». В качестве заданий для пользователей формулируются те самые задачи, выполнение которых должно обеспечивать веб-приложение. При выполнении каждой задачи действия пользователя журналируются. Устанавливаются следующие показатели, соответствующие каждой задаче:

- время выполнения задания T ;
- количество совершенных действий («клики» мышью, нажатия клавиш) Q_A для каждого активного (предусматривающего нажатие мышью) ЭУ;
- степень соответствия полученного результата ожидаемому результату R .

Теперь необходимо обеспечить вычисление веса W для каждого ЭУ, основываясь на алгоритме анализа иерархий [7]. Для этого строится иерархия (рис. 2), на первом уровне которой располагаются стадии взаимодействия человека с интерфейсом и их элементы,

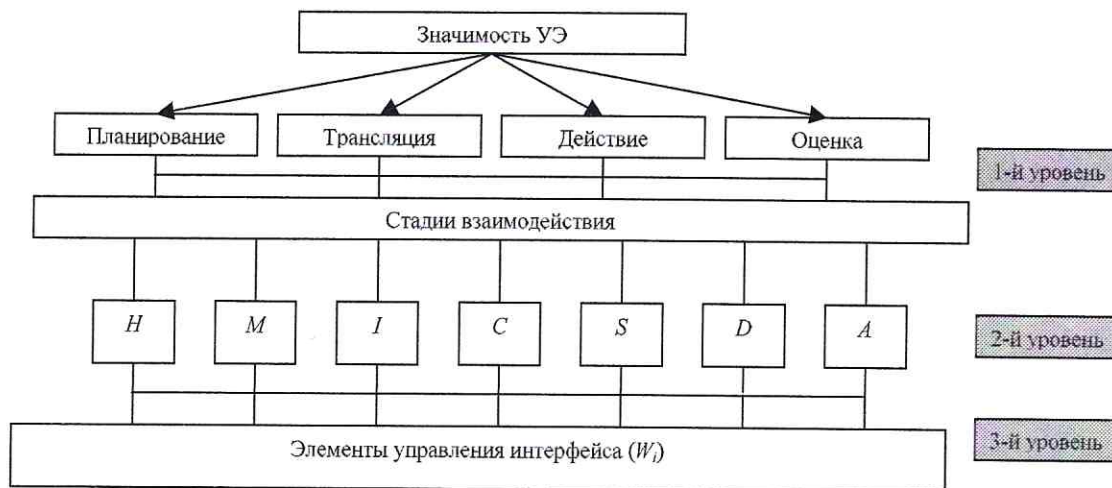


Рис. 2. Иерархия для определения весов УЭ

на втором — группы УЭ интерфейса, и на третьем — сами УЭ. Производится улучшение элементов пользовательского интерфейса по нескольким критериям. Каждому критерию соответствует свой уровень в данной иерархии. В конкретном случае критериями улучшения выступают стадии взаимодействия и группы пользовательских элементов. К достоинствам полученной схемы стоит отнести ее расширяемость, т. е. добавление новых критериев улучшения в виде уровней иерархии. Например, могут быть добавлены уровни категорий подготовки (опытности) пользователей, уровни типов используемых при тестировании браузеров и т. п.

Благодаря использованию метода стандартов, экспертам не приходится проводить попарные сравнения элементов иерархии, вместо этого они проставляют оценки по 3-балльной шкале, классифицируя каждый элемент по уровню соответствия предъявляемым к нему требованиям. В результате анализа иерархии получаем набор весов $W_1 \dots W_i$ для всех элементов интерфейса.

Вторая группа тестовых заданий направлена на вычисление параметров, используемых в функции качества интерфейса: Иерархия для определения весов $Vis(X)$, $Sem(X)$, $Arg(X)$, $Qty(X)$, $Com(X)$. Для каждого тестового задания выявляются те УЭ, которые могут быть использованы при его прохождении. По окончании прохождения каждого задания оцениваются по трехбалльной шкале все параметры каждого из используемых в нем УЭ. Вопросы строятся следующим образом: «Насколько корректно, на Ваш взгляд, сформулирована надпись на кнопке передачи данных?» ($Sem(S_n)$), «Насколько удобно расположена кнопка передачи данных?» ($Arg(S_n)$).

ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ПРОЦЕССА

Экономическая эффективность представленного метода обеспечивается путем нахождения оптимального количества пользователей для участия в тестировании качества и определения оптимального числа итераций тестирования.

Согласно распространенному мнению, эффективность юзабилити тестирования тем выше, чем больше пользователей привлечено к испытаниям. Но при исследовании работ Тома Ландауэра (Tom Landauer) и Якоба Нильсена (Jakob Nielsen) [6] сделан вывод, что количество эргономических ошибок, найденных при проведении тестов с n пользователями, равно

$$P = 1 - (1 - \lambda)^n, \quad (10)$$

где P — общее количество обнаруженных проблем с юзабилити, λ — процент эргономических ошибок, обнаруженных при проведении тестов с одним единственным пользователем.

На основе экспериментов было выявлено, что λ в большинстве случаев составляет 31%. Если построить график этой формулы при $\lambda = 31\%$, то получится зависимость, приведенная на рис. 3.

Отсюда следует: чем больше пользователей добавляется к тесту, тем меньше новой информации удастся извлечь от каждого нового пользователя, увеличивается количество одних и тех же действий и ошибок. После подключения пятого пользователя время и прочие ресурсы, необходимые для тестирования, начинают расходоваться уже не эф-

фактивно, так как повторно будут обнаруживаться одни и те же ошибки, а шансы обнаружить новые проблемы интерфейса будут уменьшаться. Далее, на основе данной зависимости представляется возможным то, что целесообразнее провести большее число итераций тестирования с меньшим числом пользователей (≤ 5), чем проведение одной или двух итераций с большим числом пользователей-тестеров.

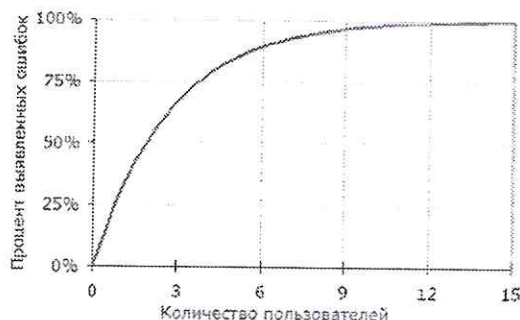


Рис. 3. Зависимость эффективности юзабилити-теста от количества привлеченных пользователей

Методика нахождения объема инвестиций в исследование эргономики веб-сайта основывается на том, что затраты на проведение юзабилити-тестов в основном состоят из выплат участникам тестирования, расходов на протоколирование и обработку результатов специалисту по юзабилити.

При инвертировании формулы (10) получено выражение, позволяющее вычислить точное количество требуемых пользователей:

$$n = \ln(1 - P) / \ln(1 - \lambda). \quad (11)$$

Для вычисления затрат получено следующее выражение:

$$S = n \cdot S_{\text{user}} + S_{\text{expert}}, \quad (12)$$

где S — искомые затраты, n — число пользователей, S_{user} — затраты на одного пользователя при прохождении одного теста, S_{expert} — затраты на эксперта по юзабилити при анализе одного теста.

Таким образом, на выявление P процентов проблем необходимо потратить

$$S = \frac{\ln(1 - P) \cdot S_{\text{user}}}{\ln(1 - \lambda)} + S_{\text{expert}}, \quad (13)$$

где λ — процент эргономических проблем, обнаруженных при проведении тестов с одним единственным пользователем.

Отсюда следует вывод о том, что выявление всех проблем нельзя назвать экономически оправданным решением, следовательно, необходимо определить рациональный процент проблем, которые необходимо выявить. Это реализуется посредством определения экономического эффекта, который достигается при выявлении данного количества юзабилити-ошибок. Повышение рентабельности интерфейса при нахождении и исправлении проблем моделируется благодаря определению времени, затраченного пользователем впусую во время одного сеанса работы с сайтом, содержащим эргономические ошибки. Для этого используется следующая зависимость, полученная Пильсеном и моделирующая предполагаемое время, сэкономленное пользователем при исправлении P процентов ошибок в интерфейсе (рис. 4).

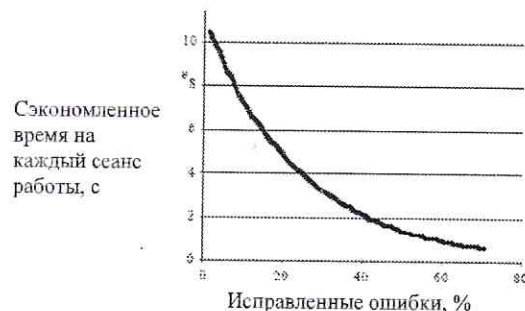


Рис. 4. Время, сэкономленное при исправлении ошибок юзабилити

На основе численных методов определяется функция, аппроксимирующая данную зависимость:

$$f(x) = e^{-0,0397x+2,36}. \quad (14)$$

Таким образом, производится оценка затрат на юзабилити-тестирование и определяется эффект от исправления ошибок юзабилити, выраженный в сэкономленном времени пользователя сайта. Сопоставим эти две зависимости и, приняв, что каждая сэкономленная секунда одного пользователя за сеанс добавляет C единиц прибыли сайту, получим

$$CVe^{-0,0397P+2,36} = \frac{\ln(1 - P) \cdot S_{\text{user}}}{\ln(1 - \lambda)} + S_{\text{expert}}, \quad (15)$$

где V — предполагаемое число сеансов работы с приложением; P — процент ошибок, которые необходимо исправить.

Благодаря тому, что единственным неизвестным остается P , данное выражение позволяет по окончании каждой итерации

юзабилити-тестирования определять необходимость последующей итерации, проверяя, достигнуто ли уже требуемое значение функции качества, т. е. какой процент ошибок уже исправлен.

ЗАКЛЮЧЕНИЕ

Таким образом, на основе исследований характера взаимодействия пользователя с интерфейсом веб-приложения приходим к следующим выводам:

- улучшение экономических показателей целесообразно производить по стадиям взаимодействия пользователя с интерфейсом и по группам ЭУ интерфейса;
- для каждой из стадий взаимодействия можно определить требования, при выполнении которых меняются такие показатели качества, как время и число действий, требуемые для достижения цели, соответствие полученного результата ожидаемому;
- качество каждой из групп ЭУ интерфейса выражается функцией. Результатом интеграции функций всех групп является общий показатель качества интерфейса;
- для улучшения полученной функции необходимо произвести сортировку ЭУ интерфейса по степени их значимости;
- значительная экономическая эффективность данного процесса достигается за счет определения части ошибок в существующем интерфейсе, которые целесообразно исправить. Дальнейшие работы по выявлению и исправлению ошибок экономически нецелесообразны.

СПИСОК ЛИТЕРАТУРЫ

1. Norman, D. A. Design of Everyday Things / Donald A. Norman. New York: Currency-Doubleday, 1988.
2. Improving Web site usability and appeal // MSN Usability research. 1997. July.
3. Landauer, T. K. Research methods in human-computer interaction / T. K. Landauer // Handbook of Human-Computer Interaction / M. Helander (Ed.). Amsterdam: North-Holland, 1988. P. 543–568.
4. Molich, R. Improving a human-computer dialogue / R. Molich, J. Nielsen // Communications of the ACM. 1990. 33, 3 (March). P. 338–348.
5. Nielsen, J. Prototyping user interfaces using an object-oriented hypertext programming system

/ J. Nielsen // Proc. NordDATA'89 Joint Scandinavian Computer Conf. Copenhagen, 19–22 June, 1989. P. 485–490.

6. Nielsen, J. A mathematical model of the finding of usability problems / J. Nielsen, T. K. Landauer // Proc. ACM INTERCSP'93 Conf. Amsterdam, 24–29 April, 1993. P. 206–213.
7. Андрейчиков, А. В. Анализ, синтез, планирование решений в экономике / А. В. Андрейчиков, О. П. Андрейчикова. М.: Финансы и статистика, 1999. 362 с.
8. Кузнецов, А. Веб-сайт как инструмент электронной коммерции / А. М. Кузнецов // Матер. 5-й Междунар. конф. по проблемам информатики и информационных технологий CSIT'2003. Уфа, 2003. С. 3–6. (Статья на англ. яз.)
9. Кузнецов, А. Как перепроектировать сайт? / А. М. Кузнецов // PC Week. Русская редакция. 2004. № 14. С. 14.
10. Кузнецов, А. Дизайн и юзабилити образовательных Интернет-ресурсов / А. М. Кузнецов, В. В. Мартынов // E Learning World. 2004. № 3 (Май–июнь). С. 30–37.
11. Кузнецов, А. Требования к графическому дизайну и юзабилити образовательных порталов / А. М. Кузнецов, В. В. Мартынов // Интернет-порталы: содержание и технологии: сб. науч. ст. / Редкол.: А. Н. Тихонов (предс.) и др.; ГНИИ ИТТ «Информика». М.: Просвещение, 2003. Вып. 1. С. 365–420.
12. Материалы исследований «Нильсен Норманн Групп». <http://www.nngroup.com>.

ОБ АВТОРАХ



Мартынов Виталий Владимирович, проф., зав. каф. эконом. информатики, рук. БРЦНИТ. Дипл. инж.-мех. (МПИ, 1981). Д-р техн. наук по АСУ (УГАТУ, 2000). Иссл. в обл. информ. систем, иссл. операций, прикл. геометрии.



Кузнецов Антон Михайлович, инж. БРЦ НИТ УГАТУ. Дипл. инж.-прогр. (УГАТУ, 2000). Канд. техн. наук по мат. и прогр. обесп. выч. машин (УГАТУ, 2004). Иссл. в обл. эргономики интерфейсов веб-приложений.