УДК [331.101.1:004.5]:004.42

Т. В. КАЗАК, А. Н. ВАСИЛЬКОВА

T. KAZAK, A. VASILKOVA

ЭРГОНОМИЧЕСКИЕ КРИТЕРИИ ОЦЕНКИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ

ERGONOMIC EVALUATION CRITERIA USER INTERFACES OF SOFTWARE APPLICATIONS

Казак Тамара Владимировна *– доктор психологических наук, профессор, заведующий кафедрой инженерной психологии и эргономики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», Республика Беларусь*
kazak@bsuir.by

Tamara Kazak *– Doctor of Psychological Sciences, Professor, Head of the Department of Engineering Psychology and Ergonomics of the educational Institution "Belarusian State University of Informatics and Radioelectronics", Republic of Belarus*
kazak@bsuir.by

Василькова Анастасия Николаевна *– магистр, старший преподаватель кафедры инженерной психологии и эргономики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», Республика Беларусь*
a.vasilkova@bsuir.by

Anastasia Vasilkova *– Master's Degree Student, Senior Lecturer of the Department of Engineering Psychology and Ergonomics of the Educational Institution "Belarusian State University of Informatics and Radioelectronics", Republic of Belarus*
a.vasilkova@bsuir.by

Аннотация: *в данной статье обобщены исследования по оценке наборов измерений удобства использования программного обеспечения эргономических критериев. Авторы приводят итоги исследовательской работы по оценке комплекса программного обеспечения в целях измерения удобства использования. В статье содержится описание каждого такого критерия. Все вместе они призваны обеспечивать качественный пользовательский интерфейс.*
В результате исследований выявлены противоречия в вопросах эргономики и идей разработчиков. Успех решения этой проблемы – в нахождении консенсуса между функционалом программного обеспечения и эргономикой интерфейса, другими словами, юзабилити пользователя.
Ключевые слова: *юзабилити интерфейса, эргономика интерфейса, критерии оценки эргономических показателей интерфейсов.*

Abstract: *this article summarizes research evaluating sets of software usability measurements of ergonomic criteria. The authors present the results of research work on evaluating a software package in order to measure usability. The article contains a description of each such criterion. Together they are designed to provide a high-quality user experience.*
As a result of the research, contradictions were identified in issues of ergonomics and the ideas of developers. The success of solving this problem lies in finding a consensus between the functionality of the software and the ergonomics of the interface, in other words, user usability.
Keywords: *interface usability, interface ergonomics, criteria for assessing ergonomic indicators of interfaces.*

Введение

В век цифровой трансформации и автоматизации человеческой деятельности использование технических устройств стало повседневной обыденностью. Это позволяет расширить или дополнить такие естественные возможности, как сложные вычисления, передача и получение информации, улучшение точности, перемещение в пространстве, принятие решений и многое другое.

Целью данной работы является определение основного набора показателей простоты использования, обычно известные как «эргономические критерии», основанные на обзоре уже написанных исследовательских работ. Исследование касается процесса проектирования эффективного пользовательского интерфейса как результата взаимного понимания разработчика и пользователя задач и действий с программным обеспечением [1].

Обоснование набора эргономических критериев оценки

Каждый программный продукт представляет собой набор различных функций. Это предоставляет пользователям различные элементы управления. Поэтому не-

обходимо иметь стандарты оценки технологий и пользовательских интерфейсов (англ. — user interface) — UI. Очевидно, что UI напрямую зависит от задачи, которую решает программное приложение (ввод и вывод данных и пр.). Однако все эти компоненты могут быть представлены пользователю в различных формах и форматах. Успех решения задачи при разработке программных приложений зависит от того, насколько функционален, понятен и удобен интерфейс.

Если интерфейс заставляет пользователей совершать ошибки, то этот интерфейс плохо продуман, по крайней мере, хуже, чем интерфейс, который помогает избежать подобных ошибок. На процесс проектирования пользовательского интерфейса наибольшее влияние оказывают собственное восприятие дизайнером ясности, удобства, красоты. Поэтому очень важно оценить качество пользовательского интерфейса. Проведение таких оценок на ранних стадиях процесса проектирования позволяет избежать многочисленных ошибок, просчетов, отклонений программного приложения со стороны конечного пользователя.

Есть различные способы оценить качество пользовательского интерфейса.

Процесс оценки включает этапы определения требований к качеству. Другими словами, интерфейс программного продукта может оцениваться на основе различных стандартов, показателей качества и атрибутов интерфейса. Хотя оценка качества пользовательского интерфейса достаточно субъективна и ее сложно формализовать, можно с уверенностью сказать, что хороший интерфейс должен обеспечивать эффективность и продуктивность работы пользователя.

Рассмотрим определенные рекомендации, которым необходимо следовать и опираться на них для проектирования пользовательского интерфейса с учетом потребностей и ожиданий пользователей:

1. Полезность, удобство использования, желание работать — три аспекта, которые должны лежать в основе разработки любого программного обеспечения [5, 6].

2. Отображение реального мира в пользовательском интерфейсе. Разработчик должен попытаться отразить язык и концепции, которые пользователи используют в реальном мире, в зависимости от их принадлежности к целевой аудитории. Предоставление логически структурированной информации и удовлетворение ожиданий пользователей от их реального опыта значительно снизит когнитивную нагрузку и облегчит использование программного продукта.

3. Стандарты и последовательность. Разработчики пользовательского интерфейса должны обеспечить выполнение условных стандартов как для графических элементов, так и для терминологии. Например, значок, представляющий одну категорию или концепцию, не должен обозначать другую концепцию, если она используется несколько раз в программе или на сайте.

4. Эффективность и гибкость. По мере увеличения частоты использования возникает потребность в меньшем количестве взаимодействий, которые обеспечивают более быструю навигацию и простоту использования. Этого можно достичь с помощью функциональных клавиш, команд, закрепленных за комбинацией клавиш, макросов. Хорошим решением было бы позволить пользователям настраивать или адаптировать интерфейс к своим потребностям, чтобы частые действия могли выполняться с помощью более удобных для пользователя инструментов.

5. Понимание. Человеческая память и внимание ограничено, и мы можем хранить в нашей кратковременной памяти лишь несколько предметов (не более 7). Из-за этих ограничений дизайнерам необходимо проектировать интерфейс так, чтобы пользователи могли просто использовать распознавание фрагментов информации вместо запоминания. Распознать что-то всегда легче, чем запоминать, потому что узнавание предполагает восприятие сигналов, которые помогают нам проникнуть в нашу долговременную память и позволяют связать образы и понять их.

6. Документация и ссылки. В идеальном программном продукте пользователи ориентируются в системе, не прибегая к документации. Однако, независимо от типа решения, документация все равно необходима. Когда пользователям нужна помощь, важно легко ее найти. Документация должна быть оформлена таким образом, чтобы она направляла пользователей к выполнению необходимых шагов для решения проблемы, с которой они сталкиваются.

7. Предотвращение ошибок. При проектировании разработчики стремятся свести к минимуму возможные ошибки. Обычный пользователь не должен выявлять и устранять проблемы с приложением, которые могут выходить за рамки их возможностей. Это уровень тестировщиков. Устранение или пометка действия, которые могут привести к ошибкам, — это два возможных способа предотвращения ошибок.

8. Свобода действий пользователя. Хорошее решение — создать цифровое пространство, где пользователь может делать шаги назад, включая отмену и повтор предыдущих шагов.

9. Состояние системы. Важно информировать пользователей о текущем состоянии программного приложения. Есть различные варианты для решения проблемы, которая возникает при таких операциях как «загрузка», «поиск», «восстановление» или др. Легко понятное и четко видимое состояние должно отображаться на экране в течение разумного периода времени, т. е. периода времени, пока система находится в соответствующем состоянии.

10. Минимализм. Цель этого аспекта — свести к минимуму беспорядок в интерфейсе. Вся ненужная информация конкурирует за ограниченные ресурсы внимания пользователя, что может помешать поиску актуальной информации. Поэтому экранное пространство должно быть ограничено необходимыми компонентами для текущих задач, обеспечивая при этом четко видимые и однозначные средства перехода к другому контенту [2, 3].

Несмотря на достаточное количество работ по вопросам разработки эргономичных пользовательских интерфейсов, отсутствие однозначных стандартов и разнообразие рекомендаций приводят к тому, что существует проблема восприятия эргономики программного обеспечения как науки в неоднозначности и стадии развития. Однако, активные исследования в этой области уже позволили сделать много полезных выводов относительно факторов, влияющих на эффективность использования программного обеспечения. Как показывает опыт, многие интерфейсы неудовлетворительны [6—8]. Это связано с тем, что они слишком уз-

копрофильны и требуют фиксированной комбинации пользовательских навыков. Хороший интерфейс должен обслуживать разные группы пользователей. Это один из главных вопросов проектирования эргономичных интерфейсов.

Рассмотрим некоторые важные вопросы в процессе проектирования эффективных интерфейсов.

Первое, это построение модели активности пользователя. Эта структура может помочь в получении оценки безошибочности, показателей потребления ресурсов и вариантного анализа альтернатив методов обработки данных [7]. Второе, мы можем выделить человеко-ориентированный и компьютерно-ориентированный подходы к разработке пользовательских интерфейсов. Однако любой из этих подходов должен быть результатом анализа задач пользователя. Из принципов эргономики [5] можно понять, почему современное программное обеспечение не может использоваться достаточно продуктивно в контексте разнообразных и неструктурированных заданий. Рассматривая типичные современные программные продукты, мы выявили некоторые недостатки, такие как:

1. Слабая интеграция. Чтобы получить желаемые результаты, конечным пользователям часто приходится совмещать работу нескольких приложений. Каждое приложение имеет свои собственные требования к формату данных. Обычно это требует ручного вмешательства пользователя. Например, с помощью текстового редактора вывести данные из одной программы в определенный формат ввода для другой. Другая проблема заключается в том, что как только данные были импортированы, их структура часто теряется. Каждая программа и сама операционная система имеют свои собственные требования к форматам данных и режимам их импорта. Некоторые программы имеют несколько режимов, которые требуют от пользователя не только соответствующих знаний, но и понимания нужного режима текущего приложения. Это является высокой когнитивной нагрузкой, следовательно, источником ошибок.

2. Потеря контекста. В ходе выполнения задачи пользователь попадает в ситуацию, когда требуется получить данные, которые сразу недоступны. Их необходимо получить каким-то образом вне контекста текущей задачи. Многие системы переключаются только тогда, когда ожидаемая задача переходит в состояние ожидания. Возможно, вместо того, чтобы приостанавливать такую задачу, ее следует закрыть. Это означает, что контекст части незавершенной задачи теряется, и пользователь должен явно вернуть ее при восстановлении.

3. Непонятность. Точные формы или форматы данных, необходимые для команд или функций программы, часто приходится искать в огромных и сложных для понимания руководствах.

4. Синдром швейцарского армейского ножа. Многие функции программного продукта могут дублировать с различными названиями идентичные функции, такие как электронная почта, поиск, письмо (поиск), архивирование (хранение), составление письма (редактирование) и т. д. Такие функции на самом деле являются похожими версиями гораздо более общих функций, которые могут применяется к одному конкретному контексту. Это может привести к большому и необоснованному дублированию функциональности и повышенной противоречивости, вызывающей постоянную опасность путаницы в понимании логики пользовательского интерфейса.

Может показаться, что недостатки были намеренно созданы с целью недопущения пользователей к эффективному использованию технологий. Однако разработчики никогда намеренно не создают интерфейсы для повседневного использования с недостатками. Недостатки конструкции выявляются в ходе реальных испытаний, которые обнаруживают виды использования, неучтенные должным образом на этапе проектирования. Недостатки — это неучет чего-либо, а намеренные затруднения использования функционала. Исключением могут быть программные продукты, предназначенные для научных целей для изучения влияния этих недостатков на пользователей.

Наличие подобных недостатков накладывает на пользователя большую когнитивную нагрузку. Это отвлекает его от реальной задачи, которую необходимо выполнить. В результате появляются многочисленными ошибки пользователей. Особенно это касается рутинных задач, в которых компьютеры легко превосходят людей. Люди начинают совершать ошибки из-за падения фокуса внимания при выполнении повторяющихся задач. Но значительная часть усилий по использованию компьютеров в качестве интеллектуального инструмента — это задача без интеллектуального содержания. Например, форматы преобразования данных. Текущая ситуация вынуждает пользователей постоянно тратить большую часть внимания на низкоуровневых аспектах общения.

Ошибки в большинстве случаев — это просто «глупые» ошибки, вызывающие раздражение и потерю времени, но не причиняющие большого вреда. Однако время от времени хорошо понятная ошибка пользователя обходится дорого, вызывая неправильные результаты или потерю большого объема работы, или даже необратимую потерю информации, которая причиняет большие затруднения [3, 4]. Даже без таких проблем общее время работы пользователей может тратиться на угадывание или попытки выяснить что-то довольно тривиальное. Эти «неизбежные» ошибки неизбежны только потому, что простейшие принципы эргономики не применяется к дизайну пользовательского интерфейса. Важным условием хорошего интерфейса является следующее:

1. Система представляется пользователю интегрированной системой с единым интерфейсом, а не набором различных интерфейсов, по одному для каждой отдельной функции системы, даже если каждый отдельный интерфейс хорошо продуман. Это, в свою очередь, требует, чтобы в системе была единая концептуальная основа: вся система — это единый концепт и структура, в которую естественным образом могут быть интегрированы различные программы.

2. Хороший дизайн пользовательского интерфейса делает взаимодействие пользователя с приложением или сайтом простым, интуитивно понятным, эффективным. Плохой пользовательский интерфейс может раздражать или мешать пользователям продолжать работу. Например, в качестве раздражающего фактора часто встречается обязательная регистрация и авторизация на ресурсах, которыми пользователи могут пользоваться однократно. Например, доставка цветов, заказ сто-

лика в ресторане — опыт говорит о том, что люди часто предпочитают пробовать новое, и нет необходимости ради однократной услуги проходить процедуру регистрации и авторизации, а потом отписываться от спама.

Еще один распространенный паттерн неудачного интерфейса — кнопки или иконки переключения языков отображения. Использование иконок флагов — не очень удачное решение ввиду того, что возможно некорректное отображение на черно-белом экране. Наиболее удачной является идея использования интуитивно понятных обозначений — значок глобуса.

Немаловажным фактором при проектировании эффективных и эргономичных интерфейсов является выбор правильных инструментов для работы и отображения их компонентов.

Общеизвестно, что основой проектирования интерфейса является выбор правильного инструмента для работы. Обычно это должно означать следующее: внимание к общепринятым символам и использование знакомых элементов управления. Например, для выбора нескольких опций использовать переключатели в виде квадратиков или использовать кнопки «плюс» и «минус», которые позволяют пользователю настроить количество желаемых опций по своему выбору.

3. Обход двусмысленностей. Для начала стоит отметить, что пользователи привыкли сами ставить «галочки» вместо удаления уже выбранных. Однако в данном случае проблема интерфейса заключается в неоднозначности толкования понятий. Чтобы удалить ингредиент по порядку, нужно снять флажок или оставить его? Эта путаница вводит пользователя в заблуждение и вызывает ряд недоразумений.

4. Чрезмерный выбор вариантов. Раскрывающиеся списки для выбора из множества элементов не являются хорошим решением. Пользователю гораздо проще, например, ввести год своего рождения, чем листать длинный список по годам. Лучше использовать списки и предложить пользователю вводить данные с клавиатуры самостоятельно, и только потом, согласно введенной комбинации, предлагать слова или символы для примера вариантов заполнения.

5. Размывание контента. Чтобы узнать всю историю компании, пользователю необходимо зайти на пять разных веб-страниц, хотя гораздо эргономичнее было бы сформировать всю концепцию компании в одной странице. Такое разрежение контента в разных разделах заставляет пользователя «блуждать» в интерфейсе и, как следствие, тратить лишнее время. Разработчикам в этом случае следует расширить содержимое одной страницы «О программе» / «О компании» за счет содержимого вложенных дочерних страниц так, чтобы пользователь, посетив этот сайт, получил полную информацию о компании, не совершая ненужные переходы на страницах.

Заключение

Проблема проектирования пользовательских интерфейсов в настоящее время актуальна. Поэтому для облегчения работы с программным продуктом разработчикам необходимо продумать и создать эргономичный пользовательский интерфейс, который соответственно выполнил бы все возложенные на него задачи. В этой статье доказывается важность эргономичного интерфейса, преимущества хорошего пользовательского интерфейса, такие как снижение общей стоимости системы, поддержка или редизайн, значительное снижение количества человеческих ошибок при использовании программного продукта.

Учитывая потребности и ожидания пользователя от дизайна интерфейса, необходимо учитывать рекомендации, которые следует принять во внимание и включить в этапы процесса проектирования пользовательского интерфейса. К ним относятся понятная документация и советы по использованию, с помощью которых разработчики пользовательского интерфейса могут игнорировать свободу действий пользователя, основанную на цифровом пространстве. Также рекомендуется прибегнуть к минимализму, который включает в себя определение основных функций интерфейса и устранение загроможденных скоплений компонентов, ненужных для текущих задач. Эти рекомендации и советы являются основой для определенных критериев.

В статье содержится характеристика оценки качества программного обеспечения, а именно обобщаются критерии эргономичности пользовательского интерфейса на основе обзора уже написанных научных публикаций. Все на следующие общие группы: функциональность, надежность, юзабилити, эффективность, мобильность.

Эргономические критерии должны учитывать возможные ошибки пользователей. Поэтому, эргономичные приложения должны быть интерактивными, интуитивно понятными, прерываемыми, «снисходительными» к ошибкам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Казак, Т. В. Технологические и психологические аспекты разработки интерфейсов. Прикладные вопросы точных наук: материалы V Международной научно-практической конференции студентов, аспирантов, преподавателей (АМТИ, г. Армавир, Россия) / Т. В. Казак, Н. И. Потапенко, А. Н. Василькова. – Армавир : РИО АГПУ, 2022 – 374 с.
2. Penha, M. Ergonomic Evaluation of Usability with Users – Application of the Technique of Cooperative Evaluation [Электронный ресурс] / M.Penha [at al.]. – Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013. – Режим доступа: https://link.springer.com/content/pdf/10.1007/978-3-642-39229-0_41.pdf. – Дата доступа: 15.10.2023.
3. Semerikov, S. Sustainability in software development. education: the case of general professional competencies [Электронный ресурс] / S. Semerikov [at al.]. – Proceedings of the International Conference on Sustainable Future: Environmental, Technological, Social and Economic Issues, volume 166. – Режим доступа: https://www.e3s-conferences.org/articles/e3sconf/pdf/2020/26/e3sconf_icsf2020_10036.pdf. – Дата доступа: 15.10.2023. (https://doi.org/10.1051/e3sconf/202016610036.)
4. Fernandez, J. Heuristic-Based Usability Evaluation Support: A Systematic Literature Review and Comparative Study [Электронный ресурс] / J. Fernandez, J.Marias // Proceedings of the XXI International Conference on Human-Computer Interaction, September 202. – N. 23, P. 1–9. – Режим доступа: https://dl.acm.org/doi/10.1145/3471391.3471395. – Дата доступа: 15.10.2023. (https://doi.org/10.1145/ 3471391.3471395).
5. Human Interface Guidelines [Электронный ресурс]. – Режим доступа: https://developer.apple.com/design/human-interface-guidelines. – Дата доступа: 15.10.2023.
6. Khan, Md.A.M.Agent-Based Ergonomic User Interface Development Environment: Analysis Phase [Электронный ресурс] / Md.Abdul Muqsit Khan // International Journal of Innovative Technology and Exploring Engineering (IJITEE). – 2019. – Vol. 9. – P. 1406–1410. – Режим доступа: https://www.ijitee.org/wp-content/uploads/papers/v9i1/A3997119119.pdf. – Дата доступа: 15.10.2023. https://doi.org/10.35940/ijitee.A3997.119119).
7. Yamaoka, T. Evaluating User Interface Design Using Hierarchical Requirements Extraction Method (REM). In: Stephanidis, C., Antona, M. (eds) Universal Access in Human-