

ЕФЕКТИВНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА

Найважливішим аспектом будь-якого програмного забезпечення є зручність і простота використання для цільової аудиторії. Саме інтерфейс слугує проміжною ланкою між користувачем і системою. Таким чином, важливо, щоб інтерфейс користувача був надійним, коротким і ясным. В даний час не існує певних правил або інструкцій для розробки інтерфейсу користувача. Дана стаття спрямована на надання пропозицій щодо дизайну інтерфейсу користувача, що дозволить користувачеві більш легко користуватися функціями системи, а також надасть можливість розробникам направляти користувачів в сторону більш ефективного використання системи.

Ключові слова: інтерфейс, веб-дизайн, юзабіліті, кольорова гама.

T.K. SKRYPNYK, E.A. MANZIUK, S.S. MOSKALIUK
Khmelnytsky National University

EFFICIENT USER INTERFACE

This article aims to provide proposals to design a user interface that allows the user to more easily use the system functions and will allow developers to direct users towards more efficient use of the system. The first thing a person sees, starting work with the information system is design. With usability, ease of use and other important nuances he will face later. One of the most important components is its adaptability. Locating the screen relative to each other should not be initiated in various sizes of the screen. The term "usability" describes the degree of comprehensibility website user, which implies its excellent ability to navigate. External interface should be highly visible. It should not distract the user from working with the information system. Although considered guidelines and rules for creating the user interface should be remembered that the quality of the user interface depends on the user and may not be ideal for all users. The goal of the developer should be to develop the required interface for most users, which should enable them to more effectively use the features of the system.

Keywords: interface, web design, usability, range of colours.

Постановка проблеми. Еволюція призначеного для користувача інтерфейсу почалась з появи простих програм, які використовували інтерфейс командного рядка. Цей інтерфейс вимагав від користувача знань команд, які будуть виконувати цільове призначення програми, що вимагає від користувача наявності попередніх знань роботи системи. Так як в той час використання комп'ютерів було не дуже широко поширене, в першу чергу обмежувалось аматорськими програмістами і науково-дослідницькими лабораторіями, зрозумілий інтерфейс користувача не був пріоритетним завданням розробників. Розповсюдження використання обчислювальних пристроїв збільшило аудиторію, яка мала взаємодіяти з програмним забезпеченням, тому вони мали отримати більш простий та передбачуваний інтерфейс [1]. Ця вимога, з урахуванням уявлень аудиторії, призводить до неоднозначних уявлень про дизайн користувацького інтерфейсу.

Аналіз останніх досліджень і публікацій. На сьогодні існує ряд стилів інтерфейсу користувача, які одержали поширення в індустрії програмних засобів: графічний інтерфейс користувача (gui – graphical user interface), web-інтерфейс користувача (wui – web user interface), інтерфейс користувача кишенькових персональних комп'ютерів (hui – handheld user interface), прикладний рівень інтерфейсу користувача програмного забезпечення, об'єктно-орієнтований інтерфейс користувача [2].

У графічному інтерфейсі застосовуються 4 елементи: вікна, піктограми, меню та покажчики. Найважливішими властивостями є можливість безпосереднього маніпулювання об'єктами, підтримка миші та інших вказівних пристроїв, використання графіки та наявність області для функцій [2].

Web-інтерфейс схожий на меню ієрархічної структури. Навігація виконується в рамках одного або декількох додатків з використанням текстових або візуальних гіперпосилань. Основні особливості WUI-додатків: інформація відображається в єдиному вікні (браузері), браузер забезпечує меню для Web-додатку; вибір дій обмежений; Web-сторінка має невисокий ступінь внутрішнього контролю над клієнтською областю для відкриття спеціалізованих контекстних меню; створення спеціалізованих меню потребує додаткового програмування; функціональні можливості додатку повинні відображатись в методи для виклику команд; клієнтська область не містить традиційних піктограм; деякі додатки використовують графіку та анімацію в естетичних або навігаційних цілях, що несе в собі загрозу зовнішнього візуального шуму і збільшення часу відгуку при завантаженні та відкритті файлів; браузер і додатки забезпечують можливість відключення графіки [3].

Інтерфейс користувача кишенькових персональних комп'ютерів застосовує "жестікуляційний" стиль з пером та сенсорним екраном для введення даних. У вікні пристрою одночасно відображається лише один об'єкт. Основні проблеми проектування: спрощення вимог до користувача по введенню даних і взаємодії; використання обмеженої області дисплею [4].

Прикладний рівень інтерфейсу користувача програмного забезпечення здійснює виведення інформації та забезпечує взаємодію з користувачем за допомогою екрану та надає користувачу піктограми та курсор. Прикладний рівень додатку, містить всі функціональні можливості, які виходять за межі

оформлення діалогових вікон та реалізовані із застосуванням різних стилів інтерфейсів користувача. Можливості прикладного рівня включають наступні аспекти: концептуальне проектування; семантику об'єктів і операцій; наочне представлення та стиль поведінки об'єктів в межах клієнтської області; синтаксис об'єктів – формати даних, діапазони вхідних даних та послідовність зміни вікон; методи взаємодії та безпосереднього маніпулювання; унікальні дії; використання фізичних пристроїв [2].

Постановка цілей. Інтерфейс має важливе значення для будь-якої програмної системи і є невід'ємною її складовою, орієнтованою, перш за все, на кінцевого користувача. Саме через інтерфейс користувач судить про прикладну програму в цілому, більше того, часто рішення про використання прикладної програми користувач приймає по тому, наскільки йому зручний і зрозумілий інтерфейс. Разом з тим, трудомісткість проектування і розробки інтерфейсу досить велика. За оцінками фахівців в середньому вона складає більше половини часу реалізації проекту. Актуальним є зниження витрат на розробку і супровід програмних систем або розробка ефективного програмного інструментарію.

Для вирішення поставленої проблеми в межах роботи поставлено такі завдання — визначити основні правила, які впливають на якість інтерфейсу користувача.

Виклад основних матеріалів дослідження. Інтерфейс користувача відіграє дуже важливу роль у кожній інформаційній системі. Звісно ідеальних інтерфейсів не існує, але є деяке уявлення про ідеальний для користувача інтерфейс, який має певні характеристики [5]:

- адаптивність;
- юзабіліті;
- чіткість;
- лаконічність.

Для підтримки цих характеристик дизайнери інтерфейсів користувача повинні дотримуватись наступних правил.

Перше, що бачить людина, починаючи роботу з інформаційною системою це дизайн. З юзабіліті, зручністю користування і іншими важливими нюансами він зіткнеться пізніше. Однією з найважливіших його складових є адаптивність. Ця властивість з'явилась після того, як з'явилися дисплеї з різними форматами. Більшість сучасних дисплеїв мають формат 1920*1080, який дозволяє розробникам відображати велику кількість контенту, але потрібно також враховувати інші формати, тому рекомендується розробляти інтерфейс адаптивно через діапазон розширення, без перерозподілення. Розміщення елементів на екрані відносно один одного не повинно бути порушено при різних розмірах екрану. Інтерфейс повинен не часто змінюватись і реагувати на зміну розміру екрану. Створення візуально привабливого змісту має важливе значення для привернення уваги користувача. Ієрархічний і логічний порядок прив'язки контенту є невід'ємною частиною гарної ідеї. Користувач повинен мати можливість легко відрізнити елементи більш високого значення або з менш важливим. Ієрархічне впорядкування також створює місце для користувача, щоб індивідуально аналізувати зміст, а не пропустити його. Також потрібно об'єднувати контент, який схожий за призначенням, комбінувати стилі оформлення, рамки та інше. Наявність єдиної колонки макету замість декількох є кращим, оскільки вона забезпечує кращу навігацію читачу в більш доступному для огляду шляху – від початку до кінця. Більша кількість стовпців має ризик відволікти користувача від основної мети сторінки [6].

Поняття «юзабіліті» характеризує ступінь зрозумілості веб-сайту для користувача, з чого випливає його здатність відмінно орієнтуватися. Сюди входять такі поняття, як зручність, швидкість, простота, інтуїтивність, широкі можливості інтерфейсу. Розташування елементів повинно бути звичним для людей. Не змушуйте людину зайвий раз думати, вона повинна знайти все на звичних для себе місцях. Зручні і диференційовані виклики дій допомагають користувачеві визначити відповідну відповідь. Виклик дії може бути кнопкою, посиланням або будь-яким іншим елементом, який може викликати певну дію. Для того, щоб зробити їх більш помітними, розробник може використовувати більш темний відтінок кольору або навпаки. Він може навіть регулювати глибину елемента, надаючи йому тінь і перехід кольору, щоб зробити його ближче, тоді як інші візуально віддаляються [7].

Реалізація розбиття на сторінки при обробці великих обсягів даних допомагає користувачам організовувати думки і полегшує розуміння контенту. Наприклад, реалізація прокрутки дозволяє відображати більше контенту на одній сторінці ніж фактичний розмір екрану, але створення занадто довгих стрічок прокрутки часто створює складності в навігації для користувачів. Просте рішення цієї проблеми полягає в реалізації розбиття на сторінки, або категоризації і розділивши весь вміст на різні сторінки і потім застосовуючи стрічку прокрутки індивідуальному порядку [2].

Форма перевірки повинна бути зроблена для кожного рядка окрема, а не в кінці процесу коли кнопка “Зберегти/Відправити” натиснута. У цьому випадку метод перевірки викликається в кінці і відображається повідомлення про помилку. Користувачеві буде необхідно повернутися назад і виправити невірні дані, які він раніше зробив. Це збільшує час роботи з інформаційною системою, а також втомлює користувача. При застосуванні перевірки в кожному рядку повідомлення буде показано користувачеві відразу після завершення вводу інформації, відповідно зменшить час корекції цієї інформації за рахунок відсутності втрат на повернення до рядку, а також сфокусованості саме на цій інформації.

Зовнішній вигляд інтерфейсу повинен бути не сильно помітним. Він не повинен відволікати увагу користувача від роботи з інформаційною системою. Але при цьому інтерфейс повинен викликати у

користувача правильні емоції, які підготують його до продуктивного проведення часу у системі. При створенні якісного, чистого і елегантного інтерфейсу, не потрібно використовувати більше 2-3 кольорів. Це істина, якою не варто нехтувати. Йдеться лише про колірну схему самої інформаційної системи. Інші кольори з'являються пізніше в фотографіях, схемах, малюнках, це нормально. Не потрібно використовувати кричущі, ідкі кольори, вони викличуть лише негативні емоції, людина не зможе довго працювати у такій системі. Краще використовувати м'які і спокійні кольори і колірні схеми, вони будуть викликати умиротворення і довіру. В цілому, в дизайні інтерфейсу користувача не повинно бути нічого зайвого, все повинно виглядати гранично акуратно, ніщо не повинно відволікати увагу користувача від контенту та уникати його перевантаження інформацією. Не потрібно створювати зайвих елементів, якщо вони не несуть ніякого сенсу і від них можна відмовитися. Кращий дизайн – легкий і просторий, що не обмежений і не переобтяжений. Також потрібно проаналізувати аудиторію користувачів інформаційної системи, так як в кожній групі різні вимоги та можливості використання [8].

Іншим важливим аспектом при розробці інтерфейсу є сумісність. Однак не слід прагнути до неї прагнути будь-якою ціною. Одною з основних її переваг є те, що користувачі можуть перенести свої знання і навички з старої програми, якою вони користувалися раніше, в нову. Принципи створення сумісності інтерфейсу:

1) Проектування послідовного інтерфейсу.

Користувачі повинні мати опорні точки при переміщенні в інтерфейсі. Це заголовки вікон, навігаційні карти і деревовидні структури. Інша візуальна допомога надає негайний, динамічний огляд місця розташування. Користувач також повинен мати можливість завершити поставлене завдання без зміни середовища роботи або перемикання між стилями введення інформації. Якщо спочатку він використовував клавіатуру, то повинна бути забезпечена можливість завершити роботу теж з нею як з головним інструментом для взаємодії.

2) Загальна сумісність всіх програм.

Вивчення однієї програми не повинно кардинально відрізнятись від вивчення подібної програми. Коли схожі об'єкти не працюють однаково в різних ситуаціях, у користувачів відбувається зміна розуміння функцій програми. Це гальмує вивчення програми і призводить до того, що користувач втрачає впевненість в своїх знаннях.

3) Збереження результатів взаємодії.

Якщо результати можуть бути відмінні від того, що очікує користувач, то потрібно інформувати його перед виконанням дії. Дати йому опції виконання дії, можливість скасувати дію або провести інше.

4) Естетична привабливість і цілісність.

Привабливий для погляду інтерфейс не повинен приховувати недолік функціональності програмного продукту [5].

Іншою проблемою інтерфейсу користувача є перевантаженість елементами управління. Перевантаженість вікон програми елементами управління зустрічається часто. Взяти інтернет-браузер. Крім того, що він сам містить кілька панелей інструментів, більше десятка кнопок, меню, скролери і т.д., вміст більшості сайтів включає безліч елементів управління – списки, кнопки. За всім цим користувачеві дуже складно знайти дійсно потрібну йому інформацію. Часто розробники програм не знають, як зробити по-іншому, і мають у своєму розпорядженні всю необхідну, на їхню думку, інформацію в одному вікні, яке при цьому перетворюється в нагромодження написів і елементів управління. Навіть досвідчений користувач буде відчувати складності при роботі з такою програмою, не кажучи вже про новачка.

Програми призначені для того, щоб допомагати користувачеві робити свою роботу. Користувач не повинен витрачати свій час на маніпуляції з програмою. Перевантаженість елементами управління призводить до того, що людина відволікається від основного завдання і приймається розглядати вікно програми, намагаючись зрозуміти, що тут, де, і як з цим працювати. Велика кількість сторонніх елементів затуляє необхідну інформацію. В результаті продуктивність роботи з програмою знижується. Крім того, користувачеві в цьому випадку практично неможливо створити у себе в голові модель поведінки цього вікна, і пізніше її використовувати. Кожен раз при роботі з цим вікном процес пізнання починається спочатку. Необхідно завжди прагнути до мінімізації числа елементів управління на екрані. Чим менше їх буде, тим краще. Може скластися враження, що це абсурд, адже виходить, що ідеальний інтерфейс повинен бути зовсім непомітним. Так воно і є. Існує поняття програм-агентів, які працюють у фоновому режимі, збираючи і накопичуючи інформацію, на основі якої виконують певні дії. Проблемою є пункт головного меню, що спрацьовує відразу. Користувачі звикли, що пункт головного меню в програмах містить підменю [2].

Якщо в головному меню присутній пункт, який спрацьовує відразу при натисканні на нього, робить програму непередбачуваною. Ще гірше, якщо це пункт «Вихід», який закриває програму без жодного попередження. Уявіть, користувач досліджує меню програми і несподівано програма зникає. Найчастіше таким пунктом дійсно є пункт «Вихід». Наприклад, пункт «Вихід» з програми краще розташовувати самим нижнім пунктом самого лівого підменю, незалежно від того, чи підходить він туди за змістом, чи ні. Користувачі починають шукати його в першу чергу саме тут. Іншою неприємністю є невідповідні малюнки на кнопках. Практично жодна програма не обходиться без панелей інструментів з кнопками. Кнопки мають малюнки. Натискання на кнопку викликає певну дію. Видається логічним, що малюнок на кнопці повинен

відповідати її дії, проте багато про це забувають. Деякі розробники просто не вміють малювати картинки, деякі просто не хочуть, і беруть готові (з інших програм). В результаті важко без підказки визначити, що означає та чи інша кнопка. Малюнки іноді виходять абсолютно не пов'язаними зі змістом. Чіткий і зрозумілий малюнок сприяє підвищенню ефективності роботи. Якщо малюнок відповідає дії, ця асоціація легко запам'ятовується користувачем, і в подальшому один тільки погляд на кнопку дозволяє миттєво згадати її призначення. Якщо ж ні, то призначення доводиться згадувати, або дізнаватися заново кожен раз.

Висновки. Незважаючи на розглянуті інструкції та правила для створення інтерфейсу користувача слід пам'ятати, що якість призначеного для користувача інтерфейсу залежить від конкретного користувача і не може бути ідеальною для усіх користувачів. Метою розробника повинна бути розробка бажаного інтерфейсу для більшості користувачів, який має дозволити їм найбільш ефективно використовувати особливості системи.

Тема дослідження буде залишатись актуальною завжди, так як технічний прогрес не стоїть на місці, а погляди людей на ідеальний інтерфейс змінюються з кожним роком.

Література

1. Константайн Л. Разработка программного обеспечения / Л. Константайн, Л. Локвуд. – Санкт-Петербург : Питер, 2004. – 592 с.
2. Тидвелл Д. Разработка пользовательских интерфейсов / Д. Тидвелл. – 2-е изд. – Санкт-Петербург : Питер, 2011. – 480 с.
3. Розенфельд Л. Информационная архитектура в Интернете / Л. Розенфельд, П. Морвиль. – 3-е изд. – Санкт-Петербург : Символ-Плюс, 2010. – 608 с.
4. Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем / Джеф Раскин. – Санкт-Петербург : Символ-Плюс, 2004. – 272 с.
5. Торрес Р. Практическое руководство по проектированию и разработке пользовательского интерфейса / Р. Торрес. – Москва : Вильямс, 2002. – 400 с.
6. Мандел Тео Дизайн интерфейсов / Тео Мандел. – Москва : ДМК Пресс, 2005. – 410 с.
7. Нильсен Я. Web-дизайн: удобство использования Web-сайтов / Якоб Нильсен, Хоа Лоранжер. – Москва : Вильямс, 2007. – 368 с.
8. Феличи Д. Типографика: шрифт, верстка, дизайн / Д. Феличи. – Санкт-Петербург : ВHV-СПб, 2004. – 360 с.

Рецензія/Peer review : 6.12.2016 р.

Надрукована/Printed :6.2.2017 р.

Рецензент: д. т. н., проф. Сорокатий Р.В.