

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
“Сучасні інструментальні засоби розробки
користувачького інтерфейсу”

Тернопіль - 2012

Р.П. Шевчук // Опорний конспект лекцій з дисципліни „Сучасні інструментальні засоби розробки користувачького інтерфейсу”, для студентів за спеціальностями: 7.05010301 "Програмне забезпечення систем", 8.05010301 – «Програмне забезпечення систем» — Тернопіль, 2012. — 103 с.

Укладач: Шевчук Руслан Петрович, к.т.н., доцент кафедри комп'ютерних наук ТНЕУ

Відповідальний за випуск: Дивак Микола Петрович, д.т.н., професор., завідувач кафедри комп'ютерних наук ТНЕУ

Рецензенти:

Завідувач кафедри комп'ютерних
систем та мереж Тернопільського
національного технічного університету
імені Івана Пуллюя
доктор технічних наук, професор

С.А. Лупенко

Доцент кафедри комп'ютерних наук
Тернопільського національного
економічного університету
кандидат технічних наук, доцент

М.Я. Шпінталь

Затверджено на засіданні кафедри комп'ютерних наук ТНЕУ.
Протокол № 16 від «28» травня 2012 р.

ЗМІСТ

ВСТУП	5
1. Основи проектування користувацького інтерфейсу	6
1.1. Поняття користувацького інтерфейсу	6
1.2. Особливості проектування користувацького інтерфейсу	8
1.3. Стилі користувацького інтерфейсу.....	13
1.4. Моделі користувацького інтерфейсу.....	15
1.5. Колектив розробників користувацького інтерфейсу	17
2. Правила проектування користувацького інтерфейсу	19
2.1. Аналіз ергономічних та людських факторів при проектуванні користувацького інтерфейсу.....	19
2.2. Основне правило проектування користувацького інтерфейсу	22
2.3. Правило 2: зменшити навантаження на пам'ять користувача	27
2.4. Правило 3: зробити інтерфейс сумісним.....	30
3. Планування робіт по проектуванні та розробці користувацького інтерфейсу.....	33
3.1. Планування робіт щодо створення користувацького інтерфейсу.....	33
3.2. Вибір моделі розробки програмного забезпечення	34
3.3. Управлінський та технічний підходи	35
3.4. Календарне планування. Деталізація плану.....	36
4. Вимоги, стандарти та керівні принципи при проектуванні користувацького інтерфейсу	39
4.1. Вимоги та стандарти.....	39
4.2. Керівні принципи та керівництва за стилем	42
4.3. Проблеми проектування міжнародних інтерфейсів	45
5. Користувачі, їх робоче середовище та задачі.....	47
5.1. Користувачі продукту, їх робота та середовище	47
5.2. Методи збирання інформації про користувачів	48
5.3. Концептуальне проектування	49
6. Макети, моделі та прототипи інтерфейсів користувача	53
6.1. Макети, моделі і прототипи	53
6.2. Цілі візуалізації проекту.....	53
6.3. Методи матеріалізації проектних рішень.....	54
6.4. Відкидання прототипів.....	55
7. Оцінка практичності користувацького інтерфейсу.....	56
7.1. Особливості оцінювання практичності користувацького інтерфейсу.....	56
7.2. Поняття зручності застосування програмного продукту	57
7.3. Особливості тестування користувацьких інтерфейсів	60
8. Графічний користувацький інтерфейс.....	62
8.1. Користувацький інтерфейс та операційні системи	62
8.2. Інтерфейс командного рядка.....	62
8.3. Інтерфейси меню	64
8.4. Основні властивості графічного користувацького інтерфейсу.....	65

8.5. Архітектура графічного користувачького інтерфейсу	67
9. Етапи розробки користувачького інтерфейсу	69
9.1. Етапи розробки користувачького інтерфейсу. Ітераційна природа розробки ..	69
9.2. Перший етап: збір та аналіз інформації від користувачів	71
9.3. Другий етап: розробка КІ	72
9.4. Третій етап: побудова КІ	73
9.5. Четвертий етап: підтвердження якості КІ	73
10. Інструментарій розробника інтерфейсів.....	75
10.1. Інструментарій розробника. Передача інформації візуальним способом.	75
10.2. Використання звуку та анімації.....	77
10.3. Термінологія та міжнародне проектування. Ключові питання розробки.....	77
10.4 Додаткові рекомендації по розробці КІ.....	79
ТЕСТОВІ ПИТАННЯ	80

ВСТУП

Дисципліна «Сучасні інструментальні засоби розробки користувачького інтерфейсу» є дисципліною профільної та практичної підготовки і відноситься до циклу спеціальних дисциплін, забезпечує базову підготовку студентів спеціальностей 7.05010301 «Програмне забезпечення систем» та 8.05010301 «Програмне забезпечення систем» та характеризується широким міждисциплінарним підходом.

Основною метою при вивченні дисципліни є засвоєння студентами основних парадигм проектування високоякісних інтерфейсів користувача та знайомство з теоретичною і практичною базою, що використовується при вирішенні задач побудови користувачького інтерфейсу.

Опорний конспект лекцій складається з десяти лекційних тем та тестових завдань для самоконтролю.

У лекційних темах наведено теоретичні відомості щодо множини задач, розв'язуваних із застосуванням технологій побудови користувачького інтерфейсу, про стан розвитку сучасних принципів проектування інтерфейсів користувача, про проблеми й напрямки розвитку цього розділу програмування; про проблеми й напрямки розвитку КІ сучасних програмних засобів, про основні методи й засоби автоматизації проектування, використовуваних у програмних засобах.

1. Основи проектування користувацького інтерфейсу

1.1. Поняття користувацького інтерфейсу

Користувацький інтерфейс (КІ) – це сукупність засобів, за допомогою яких користувач спілкується з різними пристроями (з комп’ютером або побутовою технікою) або іншим складним інструментарієм (системою). Інтерфейс користувача комп’ютерного додатку включає:

- засоби відображення інформації, відображувану інформацію, формати і коди;
- командні режими, мову «користувач–інтерфейс»;
- пристрой та технології введення- виведення;
- діалоги, взаємодію та транзакції між користувачем та комп’ютером, зворотній зв’язок з користувачем;
- підтримку прийняття рішень в конкретній предметній області;
- порядок використання програми і документації на неї.

Користувацький інтерфейс часто розуміють лише як зовнішній вигляд програми. Однак насправді користувач сприймає через нього всю програму в цілому, тобто таке розуміння є надто вузьким.

В дійсності, користувацький інтерфейс (КІ) об’єднує в собі всі елементи і компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням (ПЗ). До цих елементів належать:

- набір задач користувача, які він розв’язує за допомогою системи;
- використовувана системою метафора (наприклад, робочий стіл Windows);
- елементи управління системою;
- навігація між блоками системи;
- візуальний (і не тільки) дизайн екранів програми;
- засоби відображення інформації, відображувана інформація і формати;
- пристрой та технології введення даних;
- діалоги, взаємодія і транзакції між користувачем і комп’ютером;
- зворотній зв’язок з користувачем;
- підтримка прийняття рішення в конкретній предметній області;
- порядок використання програми і документація на неї.

Думка користувача про те, що деякий КІ «хороший», тобто кращий за інші, залежить від невеликої кількості характеристик, які стають очевидними після півгодинного знайомства з будь- яким додатком незалежно від стилю КІ. Хороший GUI- , Web- або HUI- орієнтований додаток заснований на 5 головних принципах, які позначаються абревіатурою SAPCO (Simple – простий, Aesthetic – естетичний, Productive – продуктивний, Customisable – пристосовуваний, Other – інший).

Простий. Програмні об'єкти забезпечують підвищення продуктивності і нарощування можливостей реальної системи без необхідного внесення складності в КІ. Мінімалізм і розбиття реальних систем на множину слабозв'язаних рівнів широко використовується на початкових етапах розробки стилю представлених та взаємодії.

Ключові положення, які впливають на принцип простоти:

- додаток – це не ракетно-космічна техніка;
- об'єкти повинні бути мінімально складними та багаторівневими;
- помиляйтесь краще в бік простоти.

Ці положення залежать від складності прикладної області та вимог, наданих по відношенню до КІ та практичності.

Хороший прикладний КІ не потребує довідника або діалогової довідкової системи, щоб почати виконувати за його допомогою прості задачі кінцевих користувачів. В гіршому випадку хороший прикладний КІ повинен бути зрозумілим на інтуїтивному рівні (користувачу потрібне лише пояснення, як досягти результату).

Естетичний (справляє гарне враження). Програмні об'єкти повинні мати естетичну та ергономічну привабливість, при чому широко використовується графічний дизайн та візуалізація.

Ключові положення, які впливають на принцип естетичності:

- додаток повинен в меншій мірі походити на комп'ютерні артефакти, а в більшій мірі – на користувацькі об'єкти;
- додаток повинен бути привабливим;
- слід прагнути до максимальної візуалізації інформації.

Хороший прикладний КІ повинен спровалити гарне враження.

Продуктивний (сприймається як підходящий). Використання програмних продуктів вимагає мінімальних зусиль для виконання задач кінцевого користувача. КІ реалізується таким чином, щоб уникнути складної ієрархічності вікон та/або екранів, а також непотрібних дій з клавіатурою та мишкою.

Ключові положення, які впливають на принцип продуктивності:

- додаток повинен відповідати задачі;
- для оптимізації інтерфейса слід використовувати правило “80/20”;
- кількість кроків роботи необхідно звести до мінімума;
- слід забезпечити зручність додатку;
- додаток повинен бути поблажливим до користувача і не наказувати його за невеликі помилки.

Хороший прикладний КІ дозволяє виконувати роботу швидше і не гірше, ніж його реальні аналоги, а також уникає найбільш поширеніх пасток типу надто великої кількості екранів або надто великої кількості кроків.

Пристосуваний. Програмні об'єкти доступні в різних формах для задоволення індивідуальних потреб. Програмні об'єкти КІ повинні мати властивість пристосуваності.

Ключові положення, які впливають на принцип пристосуваності:

- при проектуванні не відступайте від моделі інструментального набору;
- включайте в початкові об'єкти можливості початкового рівня;
- забезпечуйте множинні представлення, шрифти і колір, щоб надати користувачу шанс вибору;
- всі можливості слід розкривати поступово, щоб користувач виявляв їх послідовно.

Хороший прикладний КІ дозволяє користувачу обрати метод взаємодії так як і методи макетування та доступу для оптимізації користувацьких потреб.

Інші принципи. Крім принципів простоти, естетичності, продуктивності та пристосуваності існує безліч інших принципів, більшість яких – різновиди принципів SAPC.

1.2. Особливості проектування користувацького інтерфейсу

На загальному рівні проектування КІ здається простою справою. Дійсно, спроектувати будь- який фрагмент КІ не складно, однак мало хто може впоратись з розробкою повного інтерфейсу з врахуванням всіх аспектів аж до етапу розгортки КІ.

Процес проектування КІ – це складний, нелінійний, недетермінований і неортогональний процес. Складність – звичайна властивість ПЗ, але в ще більшому ступені це стосується КІ – через численні фактори та невизначеності, які впливають на його розробку. Проектування КІ – нелінійний процес, оскільки існування фіксованого, впорядкованого і прямолінійного шляху від початку до кінця зовсім не обов'язкове. Процес проектування відрізняється невизначеністю, оскільки не існує рівняння, за яким можна було б одержати одинаковий результат при заданих одинакових початкових умовах, більше того, практично неможливо одержати ідентичний результат навіть під примусом. Користувацький інтерфейс неортогональний в тому сенсі, що будь- який аспект проектного розв'язку може впливати на інші аспекти, причому результат цього впливу не завжди є приємним сюрпризом.

Програмний КІ привертає все більшу увагу і набуває все більшого значення як складова конкурентної переваги. По мірі того, як перелік функцій програмних засобів стає все довшим та складнішим, користувачі, які відповідають за придбання продукту, дивляться на КІ як на вирішення проблеми складності. Якщо КІ продукту привертає увагу користувача, якщо він легкий у вивченні, простий у використанні, а також має прийнятну ціну та

можливості, продукт має конкурентну перевагу. Конкурентну перевагу можна одержати в тому випадку, якщо заявки про нижчі витрати на навчання та виграш в продуктивності відповідають дійсності.

Існує декілька умов, які дозволяють говорити про те, що проект ведеться в орієнтованому на користувача стилі:

- розуміння користувачів та їх задач, зачленення користувачів в усі аспекти життєвого циклу продукту;
- постановка цілей, які можна виміряти; встановлення критеріїв успіху з точки зору користувачів та підприємств;
- проект повинен передбачати нову компетентність користувача, яка по відношенню до продукту включає пакетування, маркетинг, навчання, відруковану інформацію, налагодження параметрів, інсталяцію, екрані, графіку, довідки, іншу експлуатаційну підтримку, оновлення та деінсталяцію;
- оцінювання та тестування за участю реальних користувачів для визначення, чи досягнуті цілі та які проблеми існують;
- ітеративний підхід – якщо цілі не досягнуті або існують проблеми, слід внести виправлення та провести повторну перевірку.

Система та її ПЗ спілкуються з користувачем мовою представлення візуальної та слухової інформації, а також на рівні невербальної регламентації взаємодії, яка виражається у вигляді часу відзвіву, надійності, поведінки ті інших людських факторів. Користувач спілкується з системою та її ПЗ мовою дій за допомогою клавіатури, миші, мовної та тактильної інформації. Іноді система або її ПЗ накладають на користувача обмеження, пов'язані з введенням інформації в межах певного проміжку часу (час відзвіву), безпомилковим спілкуванням (надійність) та коректним підходом до використання (поведінка).

В традиційних процесах проектування використовуються різні підходи. Основні змінні, пов'язані з процесом розробки, характеризують ступінь, в якій цей процес може бути віднесений до одного з типів. Це, наприклад, такі типи процесу проектування, як проектування “ззовні- всередину” (outside- in) або “зсередини- назовні” (inside- out); однократне (без ітерацій) або багатократне (ітеративне) проектування; проектування за типом “великого вибуху” (big bang, “все або нічого”) або еволюційне.

Підхід “зсередини- назовні” спочатку розглядає внутрішні властивості системи, в той час як підхід “ззовні- всередину” спрямований на КІ та властивості продукту, видимі кінцевому користувачу. В залежності від складності проекту та застосованої технології доцільно працювати одночасно над внутрішніми та зовнішніми властивостями продукту, використовуючи інтегральний підхід.

Ітерація – це планований обсяг робіт по конструюванню продукта, особливо КІ. Підхід, в якому використовується ітеративне проектування і розробка, концентрується на побудові КІ та його основних факторах практичності – це найкращий метод розробки ПЗ. Вибір подібного підходу не забороняє і не утримує бригаду розробників від використання аналогічного типу проектування, щоб знизити ризик, зменшити не зв'язану з КІ роботу або застосувати інші технології до продукту.

Еволюційне проектування та розробка зосереджені на побудові продукту з використанням підходу на основі покрокового нарощування і уточнення можливостей продукту. Рівні можливостей проектуються і реалізуються в стилі послідовних етапів. Підхід типу “великий вибух” представляє собою спробу розробити “все або нічого”, “піти на прорив” – в цілому ПЗ розробляється і реалізується паралельно.

Найкращий підхід до розробки – еволюційний ітеративний підхід “ззовні всередину”.

Орієнтоване на користувача проектування продукту (UCD – User-Centered Design) – міждисциплінарний та ітеративний процес розробки ПЗ, спрямований на досягнення користувальських цілей у відношенні продукта, на практичність та інші вимірювані властивості продукту на протязі його життєвого циклу.

Типові етапи проектування КІ включають:

- планування;
- формування вимог;
- проектування;
- конструювання;
- розгортка.

Даний процес проектування не передбачає якогось певного стиля КІ.

Деякі ідеї, пов’язані з процесом проектування:

- 1) Процес нагадує каскадну або водоспадну модель розробки. Однак в дійсності розробка – це процес, який іде проти потоку, - з часом він не стає легшим. Каскадна модель подібна плаванню проти течії.
- 2) Обсяг і складність роботи з часом зростають.
- 3) Невелика сукупність вимог перетворюється в солідний набір екранів КІ, інструкцій та довідок.
- 4) Екрани КІ та довідки перетворюються в проект реалізації, яка повинна взаємодіяти з системною інфраструктурою, мережами та базами даних.
- 5) Проект реалізації легко перетворюється в тисячі – або сотні тисяч – рядків програмних команд (програмного коду), написаних з використанням складних та суворих мов програмування, баз даних та комунікацій, а також відповідних структур даних.

6) Програмний код піддається тестуванню, щоб продемонструвати надійність, продуктивність, якість, а також відповідність вимогам (функційним, до КІ та практичності).

7) Кількість людей, притягнутих до розробки, з часом збільшується.

На практиці для багатьох аспектів процесу відсутні явно виражені або дискретні точки завершення, наприклад, проектування продовжується під час конструювання по мірі виправлення помилок у вимогах, проекті або конструкції. Більшість проектних етапів перекриваються: наприклад, планування (управління ризиками) та управління вимогами (контроль) тривають протягом всього життєвого цикла проекту.

Ітерації не обов'язкові, але можуть знадобитися для успішного завершення якого-небудь етапу процеса, оскільки для подолання наступної сходинки каскаду може знадобитися декілька спроб. Ітерації не повинні виконуватись заради ітерацій, а зобов'язані служити досягненню цілей користувачів.

Ключова частина процесу – етап конструювання. Даний етап розробки, на якому виконується реалізація і тестування продукту, може займати до 50% проектного часу навіть у випадку вірного планування та управління.

Етапи проектування КІ, орієнтованого на користувача:

- план – план створення продукту сконцентрований на побудові КІ та забезпеченні практичності; план враховує календарні терміни для кожного з етапів процесу, визначає основні фактори ризику, об'єднує всі можливі методи, встановлює цілі та критерії по відношенню до КІ;
- вимоги – на етапі встановлення вимог виконуються наступні задачі: опис користувачів, постановка задач користувачів, оцінка поточного рівня практичності, аналіз можливостей КІ, аналіз тенденцій;
- концептуальне проектування – концептуальний проект представляє собою сукупність високорівневих описів, абстракцій та оглядової інформації, яка дає розробникам та кінцевим користувачам загальне уявлення про програмний продукт, його структуру та КІ;
- проектування – проект КІ представляє сукупність характеристик програми, які сприймаються користувачем (вхідні сигнали, взаємодія користувача, відзвів системи на вхідні сигнали та взаємодію користувача);
- прототипування – створення прототипів та моделювання – ефективні засоби ранньої оцінки проекту; прототип – це матеріалізація побудованого проекту з використанням його передбачуваної платформи реалізації, включаючи обладнання, ОС, мови і засоби реалізації;

- специфікація – матеріалізація проекту програмного продукту в документальній формі, яка описує і показує дії користувачів, а також вигляд та поведінку ПЗ в специфічних ситуаціях;
- конструювання – написання коду та автономне тестування;
- оцінка – всі методи оцінки пов’язані з залученням потенційних користувачів програмного продукту;
- ітеративний підхід – загальні критерії досягнення цілей створення КІ повинні бути чітко визначені, зрозумілі й прийняті керівництвом та розробниками; досягнення поставлених цілей може вимагати багаторазових ітерацій;
- розгортка – після того, як продукт задовільняє вимогам та потребам користувачів, він розгортається для використання за призначенням; з цього моменту починається ряд наступних дій – оцінка продукту за участю користувачів, які не залучались до розробки, пілотне тестування, виконання користувацьких задач, які не оцінювались або не були передбачені під час проектування та розробки.

Сучасне середовище розробки ПЗ містить в собі велику кількість проблем: необхідність зниження витрат, скорочення термінів, розробки більш передбачуваних планів, надання більш якісних розв’язків, забезпечення нескладного в засвоєнні та використанні ПЗ, постійне оволодіння новими технологіями та засобами, досягнення кращих результатів в порівнянні з конкурентами, а також інші аспекти.

Задоволеність користувача програмним продуктом або його практичністю може бути в значній мірі віднесена на рахунок КІ. Взагалі, задоволеність користувача – це функція невеликої кількості факторів:

***задоволеність користувача = функція від МОЖЛИВОСТЕЙ
КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСА, ЧАСУ ВІДЗИВУ, НАДІЙНОСТІ,
пристосованості до інсталяції, інформаційної підтримки, пристосованості до
супроводження та інших факторів.***

Фактори, виділені в рівнянні великими літерами, відіграють в рівнянні найбільш важливу роль. Головне, на що звертають увагу користувачі – швидке, легке та надійне виконання роботи за допомогою засобів, які автоматизують, доповнюють та полегшують виконання задачі. Простота інсталяції/деінсталяції має важливе значення, оскільки вона задає тон на початку та в кінці досвіду спілкування користувача з програмою. Легкість оновлення ПЗ має таке ж важливе значення, як і інші інтерактивні властивості продукту. Інформаційна підтримка (навчання, настанови, довідки та підтримка експлуатації) важлива під час початкового або наступного освоєння продукту, але не має безпосереднього відношення до розв’язуваної задачі.

Існує також багато інших факторів (узгодженість, інтеграція, вартість), які впливають на рівень задоволеності користувача в залежності від їх значення, середовища, розв'язуваної задачі та ситуації.

В ході планування розробки продукту, аналізу вимог, проектування, реалізації, тестування і розгортки слід враховувати всі фактори, які входять в рівняння задоволеності користувача та їх відносну важливість.

Користувач або представник компанії не завжди може чітко визначити свої потреби та вимоги на етапі планування та анлізу вимог, однак він може бути дуже категоричним відносно того, що бажано й небажано в програмному продукті, коли він вже стане доступним для апробації й використання.

Ділові й функціональні потреби – найбільш загальний вид вимог. Однак вимоги до характеристик КІ, практичності, інтеграції та узгодженості дуже часто пропускаються. В багатьох випадках характеристики КІ визначаються неявно (може навіть несвідомо) у вигляді деяких очікувань. Враховуючи, що порядка 50% програмного коду приходиться на КІ, для розуміння вимог до видимих властивостей продукту потрібно докласти додаткових зусиль. Вимоги до практичності, інтеграції та узгодженості повинні бути визначені явно і піддаватись вимірюванню. Лише тоді можна перевірити й виміряти, наскільки продукт відповідає планам. Після встановлення вимог, їх необхідно контролювати і управляти ними.

Щодо термінів розробки, то це питання кваліфікації та досвіду застосовано до сучасної технології КІ. З множиною стилів та функцій КІ пов'язаний великий обсяг інформації, який включає масу деталей, особливості очікуваної поведінки, а також надлишкові відомості.

1.3. Стилі користувацького інтерфейсу

Існує ряд стилів КІ, які завоювали популярність в індустрії програмних засобів:

Графічний користувацький інтерфейс (GUI – Graphical User Interface) – стиль взаємодії “користувач- комп’ютер”, в якому застосовуються 4 елементи: вікна, піктограми, меню та покажчики. Іноді ГКІ називають WIMP-інтерфейсом (Windows, Icons, Menus, Pointers). Вікно – це область пристрою відображення, використовувана для представлення взаємодії з об’єктами, інформації про об’єкти або для виконання дій над об’єктом. Піктограма – це область пристрою відображення, використовувана для наглядного представлення об’єкта. Меню відображає набір альтернатив, за допомогою яких користувач може здійснити свій вибір. Покажчики – це графічні символи, які візуально показують місце розташування входа в систему для координатно-вказівного пристрою (миші або шарового маніпулятора). Найважливішими властивостями ГКІ є можливість безпосереднього маніпулювання, підтримка миші або покажчика, використання графіки та наявність області для функцій і

даних додатку. Кожній основній системі властиві свої унікальні стилюві принципи побудови ГКІ, причому всі додатки даної системи повинні бути витримані в цьому стилі. ГКІ не гарантує більш високого рівня практичності, однак належним чином спроектований ГКІ- орієнтований додаток може перевершувати його аналог з неграфічним інтерфейсом за рахунок ефективності роботи користувача та ступеня його задоволеності.

Користувацький Web- інтерфейс (WUI – Web User Interface) – дуже схожий на меню ієрархічної структури. Необхідна навігація виконується в рамках одного або декількох додатків з використанням текстових або візуальних гіперпосилань. Основні особливості WUI- додатків: інформація відображається в єдиному вікні (броузері), броузер забезпечує меню для Web-додатку; вибір дій обмежений; Web- сторінка має невисокий ступінь внутрішнього контроля над клієнтською областю для відкриття спеціалізованих спливаючих меню; створення спеціалізованих меню потребує додаткової роботи по програмуванню; функціональні можливості додатку повинні відображатись в методи для виклику команд; клієнтська область не містить традиційні піктограми; велика кількість додатків використовують графіку та анімацію в естетичних або навігаційних цілях, що несе в собі загрозу зовнішнього візуального шума і збільшення часу відзвіву при завантаженні та відкритті файлів; броузер і додатки забезпечують можливість відключення графіки, щоб на екрані відображалась лише текстова версія; підтримка покажчика здійснюється для вибору за допомогою одного кліка миші або вибору за навігаційними посиланнями; технологія drag- and- drop не підтримується, обмежені дії правої кнопки миші. До найбільш поширеніх компонентів WUI- інтерфейса відносяться банери (заголовки), навігаційні панелі та візуальні або текстові гіперпосилання, впорядковані різними способами. Банер – це візуальний заголовок, відображуваний в верхній частині Web- сторінки. Навігаційна панель – це список варіантів вибору гіперпосилань, які забезпечують доступ до інформації. Гіперпосилання – це варіант вибору, який відображає наступну сторінку інформації або пересуває фокус відображення на іншу область тієї ж сторінки.

Користувацький інтерфейс кишеневкових пристрій (HUI) – користувацький інтерфейс кишеневкових комп’ютерів, для введення даних в яких застосовують “жестикуляційний” стиль з пером та сенсорним маленьком екраном. HUI- інтерфейс забезпечує деякі можливості ГКІ, а саме піктограми меню і аналогічна поведінка покажчика. У вікні пристроя одночасно відображається лише один об’єкт. Загальний стиль КІ для HUI- інтерфейса можна назвати SIMP- стилем (Screen, Icons, Menus, Pointers). Основні проблеми проектування HUI- інтерфейсу: спрощення вимог до користувача по введенню даних і взаємодії; використання обмеженої області дисплею. Основна перевага КПК – заміна фізичних кишеневкових органайзерів. Введення великих обсягів інформації в КПК складний, але досить ефективний.

Прикладний рівень КІ програмного забезпечення – GUI, WUI, HUI-інтерфейси представляють собою загальні стилі КІ. До факторів, пов'язаних з прикладним КІ, які впливають на практичність, належить спосіб використання GUI-стиля з компонентами КІ прикладного рівня. Побудований на основі КІ додаток в тій чи іншій мірі використовує компоненти стиля КІ або компоненти КІ прикладного рівня. Сполучення компонентів GUI-інтерфейса та прикладного КІ призводить до загального підвищення рівня практичності. Заснований на застосуванні КІ додаток не лише здійснює виведення інформації та взаємодію з користувачем за допомогою екрану, а й надає користувачу піктограми та покажчики. Прикладний рівень додатку, заснованого на застосуванні КІ, містить всі функціональні можливості, які виходять за межі оформлення діалогових вікон. Прикладний рівень включає всі видимі користувачу можливості і взаємодії, реалізовані із застосуванням різних стилів КІ. Можливості прикладного рівня включають наступні аспекти: концептуальне проектування; семантика і властивості об'єктів і операцій; наочне представлення та стиль поведінки об'єктів в межах клієнтської області; синтаксис (структуру) об'єктів – формати даних, діапазони входних даних та послідовність зміни вікон; методи взаємодії, які виходять за межі простого введення даних, та безпосереднього маніпулювання; унікальні дії; використання фізичиних пристройів.

Об'єктно-орієнтовані КІ – проектування програмних об'єктів дає можливість надати користувачу додаток, який має об'єктно-орієнтований стиль та/або об'єктно-орієнтовану внутрішню структуру (реалізацію). Більшість об'єктно-орієнтованих властивостей знаходять відображення в зовнішньому вигляді, поведінці, вимогах до взаємодії та функціональних можливостях КІ. Об'єктно-орієнтований прикладний КІ повинен мати наступні властивості: забезпечувати безпосереднє маніпулювання (перетягувати будь-які об'єкти куди завгодно), забезпечувати безпосереднє введення даних (записувати будь-яку інформацію), забезпечувати контекстну залежність від об'єктів (спливаючі та контекстні меню, довідки, узгодженість).

1.4. Моделі користувальського інтерфейсу

Не може бути одного найкращого інструмента, найкращої програми, інтерфейса для користувача комп'ютера, оскільки цілі користувача постійно змінюються в залежності від поставлених в даний момент задач. Користувачі потребують не лише різних інтерфейсів для вирішення різних задач, а й зміни функційності програми навіть під час виконання однієї задачі. Немає такої програми чи інтерфейсу, які б відповідали всім потребам користувача в усі часи.

Проектування інтерфейса – це відносно новий рід людської діяльності, який використовує метафору та модельне представлення як спосіб сприйняття світу людиною.

Розглянемо 4 моделі КІ – ментальну, користувацьку, програміста, проектувальника.

Ментальна, або концептуальна модель - лише внутрішнє відображення того, як користувач розуміє і взаємодіє з системою. Це відображення фізичної системи або ПЗ, в якому закладено ймовірну послідовність дій при виконанні операцій введення та виведення. Ментальна модель необов'язково точно відображає ситуацію та її компоненти, вона допомагає передбачати, що відбудеться далі.

Ментальні моделі дозволяють користувачам:

- передбачати (або позначити невидимі) події;
- знайти причини помічених подій;
- визначити необхідні дії для здійснення потрібних змін;
- використовувати їх як мнемонічні пристрої для запам'ятовування подій та зв'язків (відношень);
- забезпечити розуміння аналогічних пристройів;
- застосовувати стратегії, які дозволяють подолати обмеження, закладені в алгоритмі обробки інформації.

Модель користувача. Поганий дизайн інтерфейса викликає у користувачів сумніви в правильності своїх дій. Єдиний спосіб визначення вигляду користувацької моделі – поговорити з користувачем та подивитись, як він працює. Рекомендується 5 способів збирання інформації про користувача:

- аналіз їх задач;
- інтерв'ю з справжніми та потенційними користувачами;
- відвідування місць їх роботи;
- відзиви клієнтів;
- тести по придатності.

Спілкуватись потрібно зі справжніми користувачами, а не з їх менеджерами та керівництвом.

Модель програміста – найлегша для відображення, оскільки вона може бути формально і недвозначно описана. Дано моделі є функційною специфікацією програмного продукту.

Модель проектувальника. Проектувальник дізнається про ідеї, побажання користувача, поєднує їх зі своїми навичками та матеріалами, необхідними для програміста, та проектує ПЗ, яке повинно задовольняти потреби користувача. Модель проектувальника – це дещо середнє між моделлю користувача та моделлю програміста. Розробники, як правило, не входять в контакт з користувачами, для яких створюють програми. З'єднувачем між користувацьким оточенням та програмістським світом є проектувальник КІ.

Модель проектувальника описує об'єкти, з якими працює користувач, і техніку маніпулювання ними.

1.5. Колектив розробників користувацького інтерфейсу

Навіть для відносно невеликих проектів задача проектування програмного КІ є доволі непростою, не говорячи вже про розробку складного пакету програмних додатків з застосуванням нових технологій в умовах потужних конкурентних факторів та висунутих жорстких проектних обмежень. Сьогодні для створення стилів КІ та додатків неодмінно потрібен широкий набір різноманітних спеціальних навичок в таких областях, як користувацький інтерфейс, розробка ПЗ, тестування і якість функціонування ПЗ, графіка і візуалізація, навчання, довідкові системи, підтримка експлуатації, людські фактори, бізнес-планування, управління потоками робіт, бізнес-рейнжиніринг, управління проектами та управління змінами.

Успішність побудови КІ та програмного продукту полягає в професіоналізмі розробників, які повинні володіти наступними здібностями:

- вміти працювати в команді;
- працювати з різними людьми як в організаціях, які займаються розробкою, так і в комерційних організаціях;
- розуміти існуючі запити користувачів, а також їх майбутні потреби й цілі;
- вміти уточнювати й розширяті до необхідного ступеня деталізації неясно сформульовані вимоги;
- використовувати засоби й методи, які необов'язково призначенні для розробки ПЗ;
- швидко й ефективно оцінювати й відновлювати проект та реалізацію.

Проблеми організаційного, групового та індивідуального характера, які виникають при проектуванні продукта, відрізняються набагато більшою складністю, ніж технічні проблеми. 80% проблем пов'язані з людьми і лише 20% носять технічний характер.

Головна мета проектного колективу полягає в ефективній поставці продукту, який задовольняє вимогам до практичності, КІ та узгодженості. Крім того, колектив вимагає чіткого управління проектом, підтримки керівництва, інструментальних засобів та вірного настрою, щоб впоратись з нелегкою роботою.

Значення різних професійних навичок та участь у проекті відповідної кількості спеціалістів, які володіють кожним типом навичок, потребує ретельного планування для досягнення успішності проекту. Правильне визначення ролі розробника в колективі розробників дозволяє усунути будь-які невизначеності. Одна людина може виконувати одну або декілька ролей.

Кожен спеціаліст повинен володіти достатніми професійними навичками і убити готовим, згодним та здатним виконувати декілька ролей.

Для успішного створення продукта найважливіше значення має висококваліфікований персонал, який володіє навичками в таких областях, як: технологія розробки КІ, стандарти КІ, інструментальні засоби реалізації та використання КІ в додатках; проектування та реалізація ПЗ КІ; проектування та реалізація візуальних та графічних конструкцій; проектування та реалізація засобів навчання, керівництв, систем допомоги та електронної експлуатаційної підтримки; психологія та ергономіка, людська поведінка, сприйняття, навчання та пізнання; складання реальних бізнес- процесів.

2. Правила проектування користувацького інтерфейсу

2.1. Аналіз ергономічних та людських факторів при проектуванні користувацького інтерфейсу

Ергономіка вивчає взаємозв'язки між людиною та його роботою. Цей термін використовується як синонім дослідження людського фактора, під яким розуміється вивчення людини та використання одержаної інформації до проектування засобів, задач і середовища з метою досягти продуктивної та комфортної роботи.

Продуктивна робота приносить істотну користь бізнесу. Для вимірювання продуктивності роботи використовуються метрики (наприклад, час, необхідний для виконання задачі без помилок). З іншого боку, зручність в роботі надає не настільки істотну користь бізнесу. Вимірювання комфортності можливе в термінах рівня задоволеності користувача та його думки з приводу легкості використання продукта.

Що стосується ергономіки ПЗ та людських факторів ПЗ, то вони служать для опису результатів вивчення інформації, яка відноситься до людини-користувача, та її застосування до проектування ПЗ. Прикладне ПЗ, використовуване в комп’ютерних системах, однозначно може розцінюватись як засіб, застосовуваний для успішного виконання деякої роботи.

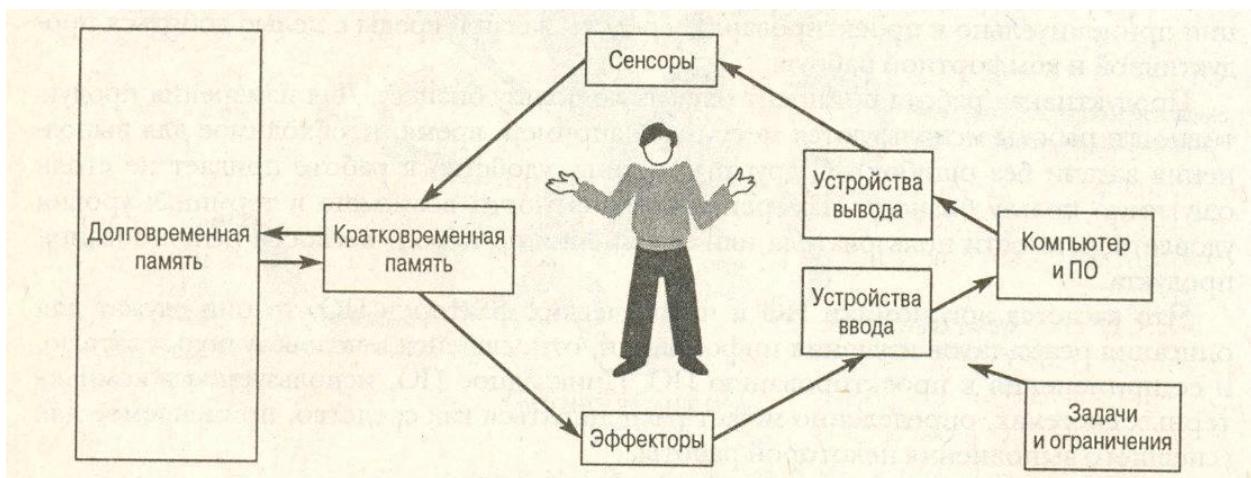
До класичних аспектів створення апаратного забезпечення належать антропометрія (вивчення фізичних розмірів тіла людини з врахуванням можливостей чоловічого та жіночого організма). Серед інших проектних факторів можна назвати положення тіла при роботі, навантаження на хребет, силу руки, а також умови фізичного середовища, в якому виконується робота. Серед всієї множини користувачів є люди з надто різними фізичними характеристиками, що повинно враховуватись в апаратних засобах, спроектованих належним чином.

Існує 2 основних принципи ергономіки:

- пристосування людини до роботи;
- пристосування роботи до людини.

Пристосування людини до роботи рівносильне навчанню і набуттю практики, в той час як пристосування роботи до людини означає проектування засобів і задач, які відповідають можливостям людей. Хоча ергономічні принципи використовуються в якості стандарту по відношенню до апаратного забезпечення, вони знаходять найширше застосування і в галузі ПЗ. Існують також деякі їх корисні розширення. Аналогічно випадку апаратного забезпечення спостерігається велике розмаїття в психологічних та організаційних характеристиках в поведінці людей. Крім того, існує значне розмаїття в задачах, необхідній освіті, навичках та стилях виконання цих задач.

Необхідне достовірне знання людей з точки зору задач використання комп'ютерного ПЗ для виконання роботи незалежно від того, чи є користувач розробником ПЗ або він працює з прикладним ПЗ. На рисунку показана спрощена модель обробки інформації людиною.



Загальні функції, які відображають важливі аспекти процесу людського пізнання, описуються досить простою моделлю. Найбільш виражені характеристики процесу людського пізнання, яким повинен приділяти увагу проектувальник ПЗ, включає наступні властивості:

- відчуття та сприйняття;
- навчання та запам'ятовування;
- увага та мова.

Користувач одержує інформацію від комп'ютера через зір та слух. Що стосується людських факторів, то основна фізична вимога до апаратного та програмного забезпечення полягає в тому, що обладнання і програми повинні сприйматись користувачем і відзвіватись на вхідні сигнали користувача. Зорова та слухова інформація, яка передається користувачу прикладним ПЗ, повинна знаходитись в діапазоні, що сприймається. З сенсорних буферів дані передаються в короткочасну пам'ять, де включаються процеси розпізнавання образів, спрямовані на усунення невизначеності, визначення контекста, ідентифікацію та інтерпретацію образів.

Недостатність уваги до людського фактору призводить до створення непродуктивного, важкого у вивченні та у використанні та неприязного ПЗ.

Сенсорні буфери в системі обробки інформації людиною тимчасово зберігають фіксовані образи, обсяг яких достатній для того, щоб користувач їх міг аналізувати. Хоча запам'ятовувана інформація відрізняється високим рівнем деталізації, вона зберігається протягом дуже короткого проміжку часу. Існує декілька типів сенсорної пам'яті – по одному для кожного з чуттів. Іконічна пам'ять використовується для зберігання зорової інформації, а звукоудавана пам'ять зберігає звукові дані. Образ представляє собою зоровий

запис, який зберігає просторові відношення, а звуковідбиття зберігає часові відношення. Інформація в кожному буфері швидко руйнується.

Частина пам'яті людини, яка зберігає обмежений обсяг інформації, яка належить до поточного часу, називається короткочасною пам'яттю. Інформація, яка зберігається в пам'яті цього типу, відрізняється наступними особливостями:

- запам'ятується автоматично;
- для видобування потрібні дуже незначні зусилля;
- обсяг порівняно невеликий (7 ± 2 елемента);
- утримується повторенням;
- видобувається відносно повільно (в порівнянні зі швидкістю вхідних сигналів);
- легко втрачається за рахунок відвертання або недостатності уваги.

Інформація, яка належить до колишнього досвіду, міститься в довгочасній пам'яті і відрізняється наступними особливостями:

- запам'ятування та видобування вимагають деяких зусиль;
- залежить від індивідуальної інтерпретації;
- відрізняється величезним обсягом (до мільярда елементів);
- організована ієрархічно за рахунок зусиль;
- відносно постійна в часі;
- усвідомлюється швидше, ніж згадується.

Частина пам'яті людини, яка зберігає історію підходів до вирішення поточних проблем, зберігає тимчасові результати та відображає зміни в планах на майбутнє, називається проміжковою пам'яттю.

Існують методи обробки інформації, спрямовані на навчання, увагу та прийняття рішень. Навчання – це складний процес, який приймає різні форми (запам'ятування, засвоєння понять та правил, моторні навички для автоматичних процедур, що повторюються). Люди навчаються, базуючись на вже існуючих знаннях. Увага передбачає свідому обізнаність та обробку інформації, відрізняється винахідливістю, може бути сконцентрованою або розсередженою. Пошук інформації, оцінка альтернатив і результатів, вибір спрямування дій складають процес прийняття рішень.

Люди, відповідальні за проектування програмних продуктів, повинні бути дуже уважні до фізичних вад користувачів. До 5% від загального числа всіх користувачів страждають на деяку форму фізичних вад. Застосовно до підтримки користувачів, яким необхідно забезпечити їх особливі фізичні потреби, використовується термін доступність. Фактор доступності слід вивчати як можливість більш якісної підтримки користувачів і уваги до їх потреб.

Висновки, які можна зробити з короткого аналізу ергономічних факторів, виходять за межі задоволення зору, слуху, відчуття, нюху, смаку та розуму

користувача. Визначаючи переваги та недоліки апаратного і програмного забезпечення комп'ютерів, проектувальники повинні спиратись на сильні сторони і обмеження, притаманні індивідуальним користувачам і колективам користувачів. Застосування ергономіки до ПЗ – це неперервний процес, що розвивається, він спрямований на оптимізацію ПЗ у відповідності до задач і процесів робочого середовища.

Керуючі принципи, в основу яких покладено принцип ергономіки про пристосування роботи до людини:

- використовуйте ергономічні стилі в апаратному забезпеченні та КІ;
- пам'ятай про користувачів;
- не вимагайте надто багато або надто мало від користувачів;
- використовуйте звичне коло знань;
- заохочуйте навчання;
- не перевантажуйте короткочасну пам'ять;
- використовуйте довгочасну пам'ять;
- проектуйте з розрахунку на помилки;
- сприяйте розробці автоматичних процедур;
- підтримуйте прийняття рішень та потік робіт;
- використовуйте установки, прийняті в системі за замовчуванням, і засоби налагодження;
- використовуйте системні можливості забезпечення доступності;
- постійно навчайтесь.

2.2. Основне правило проектування користувацького інтерфейсу

Золоте правило проектувальника: “Ніколи не робили іншим того, що вони зробили тобі”. Згадайте, що вам не подобається в ПЗ, яким ви користуєтесь, і не робіть того ж в програмі, над якою працюєте” (Трейсі Леонард).

Раніше ПЗ розроблялось без врахування вимог та побажань користувача, який повинен був підлаштовуватись до системи. Сьогодні ж програмна система повинна прилаштовуватись до користувача. Кожен позитивний досвід спілкування з програмою дозволяє користувачу розширити область знайомства з ПЗ та підвищувати свій рівень компетентності. Добре продуманий інтерфейс забезпечує плідну взаємодію користувача і комп'ютера.

Починаючи проектування, потрібно виділити найважливіший принцип, який буде визначальним при пошуку компромісів. Спроба дотриматись всіх принципів проектування може привести до того, що це не виправдає себе і негативно вплине на кінцевий результат. Принципи – не догма, а керівництво до дій.

Принципи розробки користувацького інтерфейсу:

- контроль користувачем інтерфейса;
- зменшення завантаження пам'яті користувача;
- послідовність користувацького інтерфейса.

Хансен представив найкоротший список принципів проектування:

- знати користувача;
- скоротити запам'ятування;
- оптимізувати операції;
- позбутися помилок.

Ці принципи застосовні до всього програмного та апаратного забезпечення, в усіх типах та стилях інтерфейсів. Трактування цих принципів буде, звісно, залежати від апаратного забезпечення, операційної системи, складових користувацького інтерфейса та його задач. Користувацькі моделі теж різні і впливають на те, як будуть застосовуватись принципи. На деяких важливих етапах розробки проекту може постати питання "Що відбудеться далі?". Відповідь повинна бути "Що захоче користувач".

Правило 1: дати контроль користувачу.

Основне правило проектування – дати користувачу контроль над системою. Проста аналогія – водій автомобіля та пасажир потягу. Рішення – їхати власним авто або потягом – повинен приймати саме пасажир (в нашому випадку користувач).

Розглянемо роботу банку. Комп'ютером тут користується велика кількість людей, від президента до клерків. Клерки, на відміну від президента, який має повний доступ до можливостей системи, обмежені в засобах вирішення своїх задач. Вони в певній мірі залежать від замовників. Клеркам також недоступні інформація і задачі, з якими працює президент та керівництво. Принцип розробки "дати користувачу контроль над системою" подібний до того, як президент банку дозволяє співробітникам наблизитись до інформації. Клерки повинні мати інтерфейс, який дозволяє в повному обсязі вирішувати поставлені перед ними задачі, який забезпечує контроль, гнучкість, комфортабельність і економію сил. Тобто розробник інтерфейса повинен виділити в якості основного такий принцип проектування, при яому продукт буде задовольняти всім вимогам.

Досвідчені проективальники дозволяють користувачам вирішувати деякі задачі за власним розсудом. Так, наприклад, архітектори по завершенні будівництва складного комплексу будівель повинні прокласти між ними доріжки для пішоходів. Доки вони не знають, в якому саме місці люди перетинатимуть майданчики, тому хороши архітектори ніколи не прокладають доріжки одночасно з побудовою будівель. На майданчиках між домами ставлять таблички: "Ходіть, будь-ласка, по траві". Через деякий час будівельники повертаються і лише тепер, згідно волевиявленню населення, асфальтують протоптані доріжки.

Це прекрасний приклад з реального життя, як проектувальники повинні надати можливість користувачу контролювати ситуацію і робити інтерфейс таким, щоб він відповідав потребам та побажанням користувача.

Принципи, за якими користувачу дається контроль над системою:

- безрежимність - використання режимів розважливо;
- гнучкість – надання користувачу можливості обирати (наприклад, працювати мишею, клавіатурою чи їх комбінацією);
- можливість переривання – дозвіл користувачу сфокусувати увагу;
- корисність – демонстрація повідомлень, які допомагатимуть йому в роботі;
- поблажливість – створення умов для негайних та зворотніх дій, а також зворотнього зв'язку;
- можливість орієнтування – забезпечення відповідних шляхів і виходів;
- доступність – пристосування системи до користувачів з різним рівнем підготовки;
- полегшення в користуванні – створення більш зрозумілого КІ;
- пристосуваність – надання можливості користувачу налагоджувати інтерфейс за своїм смаком;
- інтерактивність – дозвіл користувачу напряму маніпулювати об'єктами інтерфейса.

Безрежимність. Прикладом використання режимів роботи може служити палітра інструментів малювання в графічному редакторі або вікно друку з багатьма можливими варіантами друку (режимами). Режими – атрибути багатьох програмних інтерфейсів. Але застосовувати їх потрібно лише при необхідності. Гарний приклад простої роботи в режимі – текстовий процесор, де при наборі тексту, ви завжди знаходитесь в одному з режимів – вставки чи заміни.

Існує багато інтерфейсів, які надлишково часто перемикаються між режимами. В якийсь момент на екрані з'являється модальне діалогове вікно, і користувач стає обмеженим в діях не лише в програмі, але й на екрані.

Є 2 типа режимів інтерфейсу, які в багатьох випадках необхідні, однак надовго позбавляють користувача самостійності:

- режим додатку – дозволяє працювати лише в певному місці програми або встановлює конкретний режим роботи. Наприклад, при режимі перегляду документа чи бази даних немає можливості додавати, видаляти або модифікувати записи. В наш час проектувальники інтерфейсів все частіше обходяться без перемикання режимів. Можливо, оптимальним варіантом є показ даних в форматі, який відповідає рівню доступа користувача;
- системний режим – використовується він достатньо рідко. Доки користувач знаходиться в цьому режимі, йому не дозволено

працювати ніде, крім поточної програми. Наприклад, ситуація, коли в принтері закінчився папір і спливло повідомлення про це, то користувач в деяких програмах не може продовжити роботу, доки не вставить папір в принтер. При розробці таких повідомлень та інформації про допомогу слід пам'ятати про те, наскільки незручний системний режим для користувача.

При виборі режимів важливо слідувати принципу негайного візуального оберненого зв'язку. Користувач повинен бути постійно впевнений в тому, що він знаходиться в потрібному режимі. Режими інтерфейсу повинні бути настільки природні, щоб користувачу було комфортно працювати з ними.

Гнучкість. Можливість роботи з клавіатурою передбачає використання клавіатури замість миші. Це не значить, що користувачу буде легше працювати, просто він або не зможе нею користуватись, або її в нього немає. Панелі інструментів створені, щоб прискорити роботу при використанні миші. Однак при роботі з клавіатурою до них неможливо дістатись – для подібних випадків передбачені «спадні» меню. Більшість користувачів мають звичку працювати як з клавіатурою, так і з мишею.

Можливість переривання. Програмні інтерфейси повинні бути спроектовані так, щоб користувач міг в будь-яку хвилину перерватись або зберегти результати роботи в даній стадії. Забезпечення контролю користувач над програмою та його підтримка – ось головні принципи розробки. Новачки часто користуються довідковою системою, а більш досвідчені користувачі більше покладаються на власні знання.

Не потрібно змушувати користувачів закінчувати виконання початих послідовностей дій. Вони повинні мати вибір – анулювати або зберегти дані і повернутись туди, де вони перервались. Нехай у користувачі залишиться можливість контролювати процес роботи в програмі.

Корисність. Повідомлення типу «Пароль надто короткий. Він повинен бути як мінімум 26908 байт» можливо й інформативне, але воно не допомагає користувачу, тому що він не знає, скільки символів може вміститись в 26908 байт. Користувачі не повинні переводити число символів в байти. Таке повідомлення порушує принцип «прозорості».

В усьому інтерфейсі потрібно використовувати зрозумілі для користувача терміни. Даний принцип застосовний не лише для повідомлень, але й для всіх текстів на екрані, будь то запрошення, інструкція, заголовок чи напис на кнопці. Звісно, простір екрану має певну ціну, але важливіше використовувати мову, яку легше зрозуміти. Повідомлення – ключ до діалогу програми з користувачем. Всі текстові аспекти інтерфейсу повинні бути розроблені тими, хто має знання з орфографії та лексики. Вдалий напис – це мистецтво, в тому числі написання системної та програмної документації, а також повідомлень. Слід обрати вірний тон в повідомленнях та запрошеннях. Не менш важливо застрахуватись від проблем та помилок. Невдала термінологія та невдалий тон

призведуть до того, що користувачі будуть звинувачувати себе в помилках, що виникають.

Побажливість. Недостатність зворотнього зв'язку в більшості програмних продуктів змушує користувача витрачати багато сил на виконання поставленої задачі. Працюючи з інтерфейсом командного рядка в MS DOS, файли видаляються командою DEL, а потім її виконання потрібно перевіряти командою DIR, тобто зворотнього зв'язку з командою DEL немає.

Кожен програмний продукт повинен мати функції відхилення дії та повторення дії. Необхідно інформувати користувача про те, що дана дія не може бути відхиlena, і, по можливості, дозволити йому альтернативну дію. Постійно потрібно тримати користувача в курсі того, що відбувається в даний момент.

Можливість орієнтування. Потрібно надати користувачу можливість вільно орієнтуватись в інтерфейсі, визначити шляхи доступу в будь-яку його частину, дозволити пересуватись вперед і назад, по нисхідній та висхідній структурі інтерфейсу, створити зручні контекстні підказки там, де вони потрібні. Наприклад, панель задач Windows показує, які програми відкриті, і дозволяє користувачу доступ до всіх програм та даних.

Панелі інструментів, меню пуска, палтіри та інші елементи інтерфейсу існуючих операційних систем, набори програмних продуктів та утиліти розроблені, щоб допомогти користувачам орієнтуватись в операційній системі та в просторі жорсткого диску, шукати програми і дані. Системні та програмні “помічники” та “асистенти” теж пропонують допомогу при орієнтації в програмних функціях або задачах. Користувачі повинні одержувати задоволення при роботі з інтерфейсом будь-якого програмного продукту.

Доступність. Користувачі мають різний рівень навичок використання програм. Деякі програми пропонують спеціальні інтерфейси, які допомагають обрати рівень складності взаємодії. Наприклад, панель задач та спадні меню програм можуть бути стандартними або ускладненими в залежності від вибору користувача та типу виконуваної задачі. Не потрібно приносити в жертву досвідчених користувачів на користь звичайних. Потрібно передбачити для них швидкий доступ до функцій програми.

Полегшення в користуванні. КІ – «міфічна» частина програмного продукту. При хорошому проекті користувачі навіть не відчувають його присутності. Якщо він розроблений невдало, користувачам доведеться докласти чимало зусиль для ефективного використання програмного продукту. Задача інтерфейсу – допомогти людям вільно маніпулювати і працювати з об'єктами, це й називається “прозорим” інтерфейсом. «Прозорість» інтерфейсу забезпечується тим, що людині буде дана можливість користуватись об'єктами, відмінними від системних команд.

Пристосуваність. Потрібно надавати користувачу можливість налагоджувати представлення інформації так, як йому подобається (кольори,

шрифти, вигляд огляду, розташування елементів), поведінку інтерфейса (дії за замовчуванням, макроси, кнопки) та інтерфейсні функції (натискання кнопок чи клавіш, сполучення клавіш для швидкого вибору команд, мнемоніка, розташування кнопкою миші для виконання команд).

Візуально багаті сенсорні графічні та мультимедійні інтерфейси повинні містити функцію налагодження користувачем. Працюючи, користувач повинен почуватись комфортно, налагодивши інтерфейс за своїм смаком, користуючись своїми улюбленими кольорами, шрифтами, графікою та ін.

Інтерактивність. Всюди, де це можливо, потрібно дозволяти користувачу напряму взаємодіяти з об'єктами на екрані. Потрібно оптимізувати взаємодію користувача з інтерфейсом в більш природному, натуральному стилі. Користувачі повинні комфортно почуватись при виконанні даної операції та знати про передбачуваний результат. Крім того, необхідно, щоб інтерфейс було легко вивчити.

2.3. Правило 2: зменшити навантаження на пам'ять користувача

Принципи, які дозволяють знизити завантаження пам'яті користувача:

- запам'ятування – не потрібно завантажувати короткосучасну пам'ять;
- розпізнавання – потрібно опиратись на розпізнавання, а не на повторення;
- інформування – потрібно представляти візуальні заставки;
- терпимість – потрібно передбачити установки за замовчуванням, команди відхилення дії та повторення дій;
- швидкість – потрібно передбачити “швидкі” шляхи;
- інтуїтивність – потрібно активізувати синтаксис дій з об'єктами;
- перенос – потрібно використовувати метафори з реального світу;
- контекст – потрібно застосовувати розкриття та пояснення понять та дій;
- організація – потрібно збільшити візуальну ясність.

Запам'ятування. Короткосучасна пам'ять допомагає нам зберігати інформацію про невеликий проміжок часу. Користувачі звичайно працюють над декількома задачами одночасно, тому не варто КІ завантажувати короткосучасну пам'ять людини в моменти перемикань між ними. Цей принцип проектування часто порушується, що змушує застосовувати зовнішні “засоби” для зберігання інформації (папір, калькулятор).

Функції програм (відилення останньої дії та його повтор), дії з використанням буфера обміну дозволяють користувачам маніпулювати частинами інформації, необхідними в багатьох місцях, а також всередині звичайних задач. Оптимальний варіант, якщо програма в потрібному місці

може автоматично зберегти та передати дані, доки людина зайнята виконанням задачі.

Не змушуйте користувачів запам'ятовувати та повторювати те, що може й повинен робити комп'ютер.

Розпізнавання. Підтримка інтерфейсом довгочасної пам'яті передбачає розпізнавання інформації користувачем, перш ніж він згадає її. Легше обрати якийсь об'єкт зі списку, ніж згадувати його правильну назву для введення в порожній рядок. Онлайн допомога, наприклад, повідомлення, поради по користуванню інструментами, система контекстної допомоги, допомагає користувачу та обирати інформацію, а не згадувати її.

Потрібно передбачити списки і меню, які містять об'єкти чи документи, які можна обрати, не змушуючи користувачів вводити інформацію вручну без підтримки системи. Потрібно передбачити списки найбільш популярних об'єктів та документів, які можна просто обрати без заповнення командних рядків та ін.

Інформування. Необхідний аспект будь-якого ГКІ та ООКІ полягає в тому, що користувачі повинні знати, де вони знаходяться, що роблять і що можуть зробити в подальшому.

Візуальна інформація слугує нагадуванням для користувачів. Коли користувачі знаходяться в якомусь режимі або працюють мишею, це повинно відображатись на екрані. Індикація повинна повідомляти користувачу про режим, в якому він знаходиться. Форма курсора може змінюватись для вказання поточного режиму або дії, а індикатор – вмикатись і вимикатись. Тест на візуальну інформативність продукту: відійдіть від комп'ютера під час виконання задачі та пізніше поверніться до роботи. Зверніть увагу на візуальні підказки інтерфейсу, які повинні інформувати про те, з чим ви працювали, де знаходились і що робили.

Терпимість. Потрібно передбачити в інтерфейсі більше установок, які можуть бути змінені самим користувачем. Крім того, не можна забувати про забезпечення повернення до установок за замовчуванням. Іншими словами, користувачі повинні мати можливість змінити колір, шрифт, властивості програми або системи, не задумуючись про їх початковий варіант.

Потрібно використовувати здатність комп'ютера зберігати та відшукувати інформацію про вибір користувача, а також про властивості системи. Потрібно передбачити багаторівневі системи відхилення та повторення команд, які забезпечують впевнену та спокійну роботу з програмою.

Швидкість. Крім комбінованого використання миші та клавіатури потрібно відшукати інші можливості пришвидшити роботу з програмою. “Гарячі” клавіші зменшують завантаження пам'яті користувача та доводять виконання операції до автоматизму.

Є 2 основних способи для установки ярликів: прискорюючі та мнемонічні. Мнемонічні (або доступні) – це одиночні буквенно- цифрові символи, які пересувають курсор на потрібний об'єкт та дозволяють зробити вибір. Вони використовують різні меню (панель, спадні, які з'являються) та списки. Мнемонічні символи повинні бути унікальними для кожного роду дій. Наприклад, типове меню вікна використовує стандартні клавіші – F для файлу, E для редагування, V для перегляду, H для виклику довідкової системи. Наступний рівень меню (спадні) використовують свої установки мнемонічних клавіш. Наприклад, N – новий документ, O – відкрити, C – закрити, S – зберегти. Мнемонічні символи пришвидшують рух і вибір потрібного меню або списку.

Прискорювачі (або клавіші швидкого доступу) – це клавіша або комбінація клавіш, які користувачі повинні натиснути для здійснення якої-небудь дії. Наприклад, Ctrl+P – прискорювач для друку.

Як тільки користувачі добре засвоють програмний продукт, вони починають відчувати потребу в прискорювачах. Не слід ігнорувати цю необхідність, однак при проектуванні потрібно слідувати стандартам.

Інтуїтивність. Не потрібно розробляти повністю ОО інтерфейс, щоб одержати переваги від використання ОО синтаксису взаємодії. Йому слідують навіть орієнтовані на додатки програми, подібні текстовому процесору. Переваги ОО синтаксису полягають в тому, що користувачам не потрібно запам'ятовувати, яка дія правильна в певний момент часу для даного об'єкту. Перш за все потрібно обрати об'єкт. Потім дозволяються дії, які можуть бути виконані над об'єктом, лише тоді, коли він доступний. Недоступні дії виділяються сірим кольором.

ОО синтаксис дозволяє людині зрозуміти взаємозв'язок між об'єктами та діями в програмному продукті.

Перенос. Метафори реального світу дозволяють користувачу переносити свої знання з реального світу в світ комп'ютерів. Наприклад, не потрібно для програми Калькулятор розробляти новий інтерфейс, він повинен збігатись з інтерфейсом звичайного калькулятора, який добре засвоїли користувачі. Але потрібно бути обережними при виборі та використанні метафор для інтерфейсу. Обираючи метафору, зафіксуйте її та слідуйте ній весь час. Якщо ви виявите, що метафора не відповідає свому призначенню в усьому інтерфейсі, оберіть нову метафору. Продовжуйте метафору, не перериваючи її.

Контекст. Користувачі не повинні помилитись з приводу можливостей програмного продукту. Не потрібно показувати абсолютно всі його функції, а лише ті, в яких є потреба. Деякі програмні продукти надають різні меню для користувачів. Спочатку можна обрати просте меню для щоденних, звичайних потреб. Пізніше, по мірі засвоєння продукту об при виникненні потреби в більш складних властивостях програми, користувачі зможуть обрати більш складен меню. Це приклад того, як користувач одержує контроль над задачею

та програмою. Нові властивості інтерфейсів (Майстри та Порадники) розкривають та пояснюють поняття та дії при вирішенні звичайних задач. Майстри допомагають користувачу виконувати задачу крок за кроком, кожен з яких ясний й зрозумілий.

Ніколи не слід забувати про легкий доступ до часто використовуваних функцій та дій. Можна приховати популярні властивості та функції та дозволити користувачу викликати їх по мірі необхідності. Не потрібно намагтись відобразити всю інформацію в головному вікні. Можна використовувати вторинні вікна.

Організація. Потрібно застосовувати принципи візуального проектування, що людині було легше сприймати інформацію: групування об'єктів в меню або списки; нумерація об'єктів; використання заголовків та запрошень.

Деякі програми одночасно надають надто багато інформації на екрані. Це викликає відчуття хаосу. Інформація повинна показуватись в порядку, зрозумілому для користувачів. Тобто “форма повинна слідувати призначенню”.

Комп’ютерні графіки та оформлення книг добре засвоїли мистецтво подання інформації. Цю навичку повинні мати й члени команди по розробці КІ.

2.4. Правило 3: зробити інтерфейс сумісним

Сумісність – ключовий аспект для використання інтерфейса. Однак не слід будь що прагнути до неї. Однією з основних переваг послідовності є те, що користувачі можуть перенести свої знання та навички зі старої програми, якою вони користувались раніше, в нову.

При створенні інформаційного повідомлення бажано присвоїти йому ідентифікаційний номер. Далі разом з повідомленням використовуйте його номер. Це дозволить користувачу зазначати, що схожі повідомлення з’являються кожен раз в певній ситуації, незалежно від того, в якому місці інтерфейсу він знаходиться.

Принципи створення сумісного інтерфейсу:

- послідовність – проектування послідовного інтерфейсу;
- досвід – спільна сумісність всіх програм;
- прогнозування – збереження результатів взаємодії;
- відношення – естетична привабливість та цілісність;
- передбачуваність – заохочення вивчення.

Послідовність. Користувачі повинні мати опорні точки при пересуванні інтерфейсом. Це заголовки вікон, навігаційні карти та деревовидні структури. Інша візуальна допомога надає негайний динамічний огляд місця розташування. Користувач також повинен мати можливість завершити поставлену задачу без зміни середовища роботи або перемикання між стилями

введення інформації. Людині необхідна візуальна допомога, яка прогнозує результати роботи. Контекстна допомога (поради і т.ін., меню об'єктів і кнопок) також гарна можливість для користувача. Він не повинен виходити з програми, не одержавши інформації, яка підкаже, що відбудеться при завершенні задачі.

Досвід. Один з головних аспектів в розробці інтерфейсу – можливість навачиння користувача голвним концепціям системи та програмного продукту, які можуть застосовуватись на нових ситуаціях та інших програмах. Сумісність проявляється на 3 рівнях: подання інформації, поведінка програми та техніка взаємодії. Сумісність – одна з головних властивостей КІ.

Сумісність в поданні інформації передбачає, що користувачі можуть сприймати інформацію та об'єкти в схожому логічному, візуальному та фізичному вигляді в усьому програмному продукті. Не змінюйте стилю представлення інформації без видимих причин.

Сумісність в поведінці програми передбачає, що об'єкт одинаковий всюди. Поведінка контрольних інструментів, таких як кнопки, списки і меню, не змінюється всередині або від програми до програми. Користувачі повинні бути впевнені в послідовній поведінці об'єктів інтерфейсу.

Сумісність техніки взаємодії теж дуже важлива. Однакові сполучення “швидких” клавіш повинні працювати в схожих за призначенням програмах. Способи роботи з мишею не повинні відрізнятись від програми до програми. Мнемонічні схеми клавіатури теж повинні бути незмінні. Користувачі очікують аналогічних результатів при взаємодії однаковими методами з різними об'єктами.

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

Прогнозування. Якщо користувач, виконуючи одні й ті ж дії, одержує різні результати, він більш скильний звинувачувати себе, ніж продукт. Це призводить до того, що людина втрачає впевненість в результатах своєї роботи.

Сумісність кроків і дій повинна витримуватись в усьому програмному продукті. Процеси реєстрації, навігації повинні бути послідовними. Стандартні елементи інтерфейсу повинні поводитись однаково. Якщо результати можуть відрізнятись від того, що очікує користувач, інформуйте його перед виконанням дії. Дайте йому опції виконання дії, скасвання дії або здійснення іншої дії.

Відношення. Деякі з сучасних програмних продуктів виглядають так, ніби вони розроблені різними людьми, які навіть жодного разу не спілкувались між собою. Сумніви користувачів про цілісність програмного продукту виникають,

якщо в усіх його частинах виявляється непослідовність кольорів, шрифтів, іконок та інших складових інтерфейсу.

Приємний для погляду інтерфейс не повинен приховувати недолік функційності програмного продукту. Користувачі повинні одержати гарний інтерфейс, який допоможе їм в роботі.

Передбачуваність. Задача проектувальників КІ – створити дружній інтерфейс, який заохочуватиме користувача досліджувати його складові і властивості без страху зробити щось неправильно.

Зараз настає час перетворення дружніх інтерфейсів в приємні при використанні та привабливі інтерфейси навіть в програмах, призначених для бізнесу.

3. Планування робіт по проектуванні та розробці користувацького інтерфейсу

3.1. Планування робіт щодо створення користувацького інтерфейсу

Кінцеві КІ програмних продуктів привертають все більшу увагу і набувають значення як характерна особливість продукту, яка забезпечує конкретну перевагу. По мірі того, як перелік функцій продуктів стає все довше, користувачі, які відповідають за придбання продуктів, все частіше звертаються до інтерфейсу. Якщо КІ продукту справляє враження простого для вивчення та використання, продукт має всі шанси одержати конкурентну перевагу, в особливості якщо він претендує на зниження витрат при засвоєнні, а з точки зору продуктивності надає реальні вигоди.

Але для появи продукту з реальною перевагою потрібна узгоджена, систематична та упорна робота з боку керівництва та технічного персоналу. Процес проектування та розробки програмних продуктів включає: планування, збір вимог, концептуальне проектування, проектування, реалізацію (системне проектування, проектування програм, напис програм та компонентне тестування), тестування (системне та комплексне), виконання ітерацій (доки вимоги не будуть задоволені) та завершуючу дію по розгортці.

Існують різні способи представлення найбільш важливих елементів процесу розробки – кожен крок виконується явно і свідомо або кожен крок виконується неявно і несвідомо. Деяка критична “маса” роботи повинна бути виконана ефективно та продуктивно. Найкращий спосіб приступити до роботи над складним, непідвладним помилкам та таким, що відрізняється високою ймовірністю ризиків, проектом полягає в складанні плану, який включає діяльність по управлінню ризиками.

Перший крок на шляху до орієнтованого на користувача продукту – це планування. Звичайно орієнований на користувача процес зображується як лінійний, однак він завідомо нелінійний. Проходження послідовних етапів розробки носить нелінійний та неортогональний характер. Центральним моментом кожного з етапів процесу розробки є по меншій мірі одна оцінка відповідності продукту вимогам, яка проводиться сумісно з кінцевими користувачами. В ідеальному випадку користувачі безпосередньо приймають участь в процесі розробки продукту.

Від 30 до 50% всього обсягу програмного коду пов’язано з КІ.

Проектування та реалізація прикладного ПЗ з гарним КІ вимагає врахування багатьох деталей, глибоких знань та упорної праці в прикладній області, а також в області ПЗ та реалізації. Задача ще більш ускладнюється, якщо до продукту висуваються суворі вимоги у відношенні практичності, узгодженості, інтеграції, а також до кросплатформенних аспектів та конкурентоздатності. Однак найбільш важкий бік роботи над КІ звичайно

пов'язаний з людськими факторами – навиками і досвідом людей, груповою динамікою та організаційною поведінкою.

В якості першого кроку для вирішення важкої задачі потрібно розглянути обсяг робіт, витрати та ризик, пов'язані з розробкою продукту (ризикованого з точки зору високого рівня змістовності КІ або високого рівня практичності), та скласти докладний план.

План створення продукту, спрямований на деталізацію та вимірювання елементів процесу розробки КІ, допомагає деякою мірою виключити з процесу такі фактори, як натхнення, диво та сліпу віру в удачу, а також до деякої міри обмежити ризик.

Щоб досягнути успіху в створенні продукту, план повинен задовольняти наступні вимоги:

- в плані створення продукту повинні бути конкретно відображені найважливіші ключові події календарного графіку, які відносяться до кожного з етапів процесу, а також компоненти КІ та показники практичності;
- виконання плану повинно базуватись на обліку та відстежуванні відповідності якості компонентів поставки та показників практичності плану;
- відстежування ходу виконання плану та звітність по поточному стану робіт повинні служити меті управління елементами ризику, пов'язаними з багатьма нестійкими задачами;
- на кожному етапі процесу здійснюється явна оцінка та формування звітності, які стосуються відповідності продукту вимогам, які до нього висуваються.

Ніколи не можна починати проект без плану.

3.2. Вибір моделі розробки програмного забезпечення

Чимало критики виказується на адресу каскадної моделі розробки ПЗ, яка звичайно відображається у вигляді низхадного потоку робіт. З точки зору проектування та розробки ПЗ та КІ краще всього на практиці зарекомендував себе вибір деякої форми ітеративної або еволюційної моделі. Ще один ефективний практичний метод – ітеративна модель розробки за участю користувачів. Існує багато різновидів ітеративних процесів та процесів сумісної розробки – потрібно обрати ожин з них, який спрацьовує застосовано до вашої організації та проекту, та притримуватись його.

Ітеративний підхід з залученням користувачів краще всього зарекомендував себе на практиці, оскільки ні в кого немає готових відповідей на масу питань та проблем, які виникають при проектуванні програмного КІ, який спрямований на задоволення потреб організацій та користувачів. Експерти рекомендують ітеративний підхід також з наступних причин:

- КІ програмних продуктів складні. Може знадобитись чимало ітерацій, щоб пропрацювати деталі конкретного підходу для задоволення потреб з точки зору ергономіки, користувачів та бізнес- процесів;
- Слід вивчати користувачів, способи їх взаємодії з комп’ютерами та програмними додатками. Ітерації за участю користувачів допомагають пристосувати певний підхід до конкретної групи користувачів та задачі.

3.3. Управлінський та технічний підходи

Проектування та розробка КІ, а також забезпечення практичності ПЗ з використанням відомих стилів КІ вимагають твердих навичок та вміння працювати разом. Жоден спеціаліст окрім не володіє всіма знаннями, необхідними для виконання різних задач. Колективу розробників потрібні чіткі цілі, підтримка з боку керівництва, інструментарій, приміщення та відповідний настрій, щоб впоратись з складною роботою.

Не слід недооцінювати важливість наявності в колективі відповідної кількості спеціалістів, які володіють певним типом навичок.

Навички, необхідні для проекту, відображаються на специфічні ролі орієнтованого на користувачів колективу по створенню продукту.

Кожен член колективу повинен володіти достатніми навичками для виконання декількох ролей. Якщо деякі спеціалісти не підходять, їх можна задіяти в іншому проекті. План- графік розробки повинен враховувати існуючий рівень навичок та особливості формування проектного колективу. Успіх продукту – заслуга всіх учасників колективу. Якщо ж продукт зазнає невдачі, повинно бути ясно, хто й за що несе відповідальність.

Управління проектування та розробкою КІ повинно носити активний характер. Наряду з іншими задачами управління вкрай важливо сформувати належну бригаду, яка володіє необхідними навичками. Слід чітко розподілити відповідальність, призначити технічних лідерів та відповідальних керівників, скласти та відстежувати необхідні плани, виконання яких піддається вимірюванню, стежити за виконанням вимог, встановити обов’язки та винагороди.

Технічний підхід до проекту визначається технічним персоналом. Технічні спеціалісти приймають рішення про кількість користувачів, які залучаються до участі в проекті, а також форму їх участі, технології КІ, стилі, стандарти, розглядають такі питання, як технологія і засіб реалізації, взаємозв’язок з роботами, які не стосуються КІ, їх черговість, підхід до створення прототипів, підхід до оцінки та ітерації.

3.4. Календарне планування. Деталізація плану.

План створення продукту не може вважатись завершеним, доки його виконання не піддається вимірюванню, або доки в ньому документально не зафіковані календарні терміни основних етапів та поставки компонент. Основними етапами називають події, передбачені план-графіком, результат яких можна вимірюти.

Роботи, в результаті яких не створюються потрібні компоненти продукту, повинні бути критично оцінені до того, як найбільш цінні проектні ресурси будуть покладені на жорсткий план-графік. Якщо вони не сприяють вдосконаленню продукту, їх не потрібно виконувати.

Для кожного етапу розробки оцінюється, наскільки компоненти, що поставляються, відповідають критеріям якості, а також складаються звіти про досягнуту продуктивність в порівнянні з планом. Відхилення в термінах не обов'язково призводять до негативного результату, оскільки корекцією деяких відхилень можна управляти. Для досягнення цілей іноді потрібно вносити корективи та вказівки в проектний план.

Кожен крок план-графіка для ітеративного підходу представляється як проміжний етап створення КІ з вимірюваними результатами для ключових елементів поставки кожного кроку. До елементів поставки належать такі оформлені результати, як цілі створення КІ, специфікація, дослідження ринку, бізнес-претенденти, розробка, звіти про тестування.

План-графік створення КІ слід розглядати в 3 різних часових перспективах:

- загальної тривалості проекту (узагальнені етапи);
- планування на 90 днів вперед;
- деталізація план-графіка в межах двох тижнів (особливо під час проектування, програмування та модульного тестування).

Погляд з точки зору загальної тривалості проекту дає повну картину по відношенню до ЖЦ проекту та важливих етапів. Оскільки проект з більшою тривалістю відрізняється значним ризиком та його хід важче прогнозувати, доцільно оформити 90-денні відрізки робіт та більш деталізовані проміжні етапи. Незалежно від тривалості проекту двотижневі плани легко обмірковувати, прогнозувати, деталізувати, виконувати та відстежувати.

Вміст створюваних в ході проектування функцій слід розділяти на “порції” ПЗ, які включають діалогові вікна, сторінки та екрани, а також компоненти та методи, які їх підтримують.

Один з підходів до розбиття плану полягає в поетапній поставці вмісту проекту з використанням планованих ітерацій, як показано на рис.1. Ітеративний орієнтований на користувачів план визначає принаймні 4 основних ітерацій:

- I1=концептуальні проектні рішення. Проектні рішення для головних, найбільш часто використовуваних елементів з підвищеним ризиком; дослідження альтернативних рішень, компромісів та обмежень; оцінка інструментальних засобів. Ці рішення базуються на наближенному підході до прототипування та оцінки.
- I2=20%-ий обсяг проектування. Виправлення та/або покращення рішень, вироблених в ході ітерації I1; додавання 20% функційних можливостей, використання яких займає 80% часу; аналіз інших областей, які стосуються проектування. Ці рішення базуються на точному підході до прототипування та оцінки.
- I3=наступні 20%. Виправлення та/або покращення рішень, вироблених в ході ітерації I2; додавання наступних 20% функційних можливостей та вирішення інших питань, які стосуються проектування (проблемні задачі користувачів).
- I4+=завершення проектування. Виправлення та/або покращення рішень, вироблених в ході ітерації I3; додавання останніх функційних можливостей, які мають менше значення та які можна реалізувати протягом останнього короткого періоду проектування. Це, наприклад, повідомлення, мало використовувані функції. Для цих елементів розробляються специфікації, вони підлягають перегляду та реалізуються.

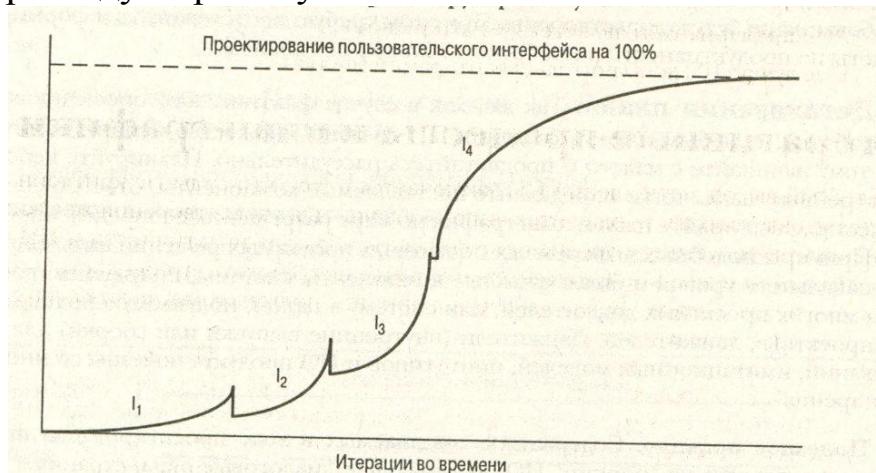


Рис.3.1. Концептуальні проектні рішення нарощуються під час ітерацій на етапі проектування

Ітерації мають важливе значення лише тоді, коли мова йде про задоволення критерій.

Головний висновок по відношенню до КІ полягає в тому, що сильні критерії рідко вдається задовольнити за одну ітерацію – звичайно для задоволення жорстких критеріїв та при допустимому рівні помилок, що залишилися, для практичності та КІ потрібно декілька ітерацій. Ітерації, взагалі, це метод спроб та помилок.

Ітерації не обов'язково напряму впливають на збільшення календарного плану. Єдина частина проекту, яка вимагає змін, це частина, яка призводить до відхилення від заданих критеріїв або до чисельних проблем практичності. В цьому випадку в ефективному проекті спостерігається відставання, яке слід наздоганяти.

При складанні план-графіків великих проектів та складного ПЗ можна використовувати наведені нижче оцінки календарної тривалості робіт:

- концептуальне проектування – 8 тижнів;
- початкова ітерація з використанням прототипу – 6- 8 тижнів;
- проектування, прототипування та підготовка до тестування – 6 тижнів;
- тестування, аналіз та планування наступної ітерації – 2 тижні;
- високорівневе проектування/прототипування/оцінка/специфікація – 1- 2 місяці;
- низькорівневе проектування/прототипування/оцінка/специфікація – 2- 3 місяці;
- кожен користувацький елемент управління – 2- 3 місяці.

4. Вимоги, стандарти та керівні принципи при проектуванні користувацького інтерфейсу

4.1. Вимоги та стандарти

В якості причин невдалої розробки програмних продуктів частіше всього називають дві, які мають пряме відношення до вимог: недолік ясних, визначених та вимірюваних вимог, а також обмежений контроль за вимогами під час розробки. Можливо, існує й третя причина - нестача ясних вимог по відношенню до практичності, КІ, узгодженості, комплектації, користувацьких та ділових потреб та очікувань.

Найбільш важку частину збирання вимог складає робота з людьми для визначення їх реальних потреб та запитів. Вимоги необхідно збирати при розробці будь- якого продукту. Основна проблема, пов'язана з розроблюваним продуктом (незалежно від етапу розробки), полягає в збиранні та управлінні введенням нових вимог в процесі розробки. При збиранні вимог потрібно приймати до уваги міркування як технічного, так і соціального порядку.

Орієнтований на користувачів колектив розробників формує наступні **ВИМОГИ**:

- аналіз задач та прецедентів;
- можливості (функції, користувацький інтерфейс і т.д.);
- критерії (практичність, продуктивність, якість і т.д.);
- конкуренти та кращі зразки;
- користувачі, технології, обмеження;
- методи залучення користувачів;
- візуалізація проектних рішень.

Вимоги до КІ додатків звичайно представлені на дуже високому рівні і в них не вистачає специфіки. Вимоги до КІ залишають великий простір для інтерпретації та уявлення, однак існуючі документи можуть служити непоганою відправною точкою для розуміння потреб продукту.

Задачі кінцевих користувачів вимагають хорошого опису разом із виясненням розподілу частоти їх виконання. Крім виявлення задач, які стоять перед користувачем, корисно ознайомитись з процедурами, яким слідують користувачі в процесі виконання роботи.

Наряду з описом функційних можливостей необхідна деталізована інформація про кінцевих користувачів, тому що кожен користувач - це особистість з унікальними навичками, поглядами та знаннями. Опис користувачів повинен бути досить повним, щоб проектувальники та розробники КІ могли зрозуміти потенційних користувачів, уяснити розбіжності їх навичок та відносин. Вимоги повинні включати також

аналогічну інформацію, яка стосується користувачів країни, для якої розробляється продукт.

Використання методів реверсивної інженерії дозволяє сформулювати вимоги, базуючись на існуючій системі (реінжініринг ПЗ - реконструкція існуючих систем з наступною переробкою та реалізацією нових проектних рішень). Перелік функцій, необхідні дані, час відзвіву, потік задач, поточна інформація та методи побудови КІ виводяться (реконструються) на основі аналізу існуючої системи. Потреби організації та користувачів додаються до вимог та піддаються перегляду з точки зору їх повноти та пріоритетів.

Наряду з існуючими використовуваними продуктами до інших джерел даних по можливостям, КІ, інформаційні підтримці та практичності належать реальні конкуруючі та аналогічні продукти. Конкуруючі та аналогічні продукти розглядають з використанням методів, застосовуваних для оцінки існуючих продуктів.

Є чимало джерел інформації у вигляді галузевих публікацій.

Приклади можливостей КІ, необхідних для будь-яких типів інтерфейсів: підтримка графіки (статичної та анімаційної), меню, засоби навігації, клавіші швидкого вибору команд, засоби маніпулювання, спеціальні операції, мінімалізм, налагодження під користувача, альтернативні пристрої введення-виведення, звукова інформація, довідники та робоча підтримка, інтерактивні керівництва, допомога та "майстер"- програми.

Типовий сценарій збирання вимог:

- кінцевий користувач звертається до керівника;
- керівник звертається до представника ІТ- підрозділу;
- представник ІТ- підрозділу звертається до менеджера ІТ- підрозділу;
- менеджер ІТ- підрозділа звертається до спеціаліста з планування продуктів;
- спеціаліст з планування розвитку продуктів звертається до колективу розробників.

Вимоги краще всього сприймаються у вигляді окремих команд та екранів прототипу КІ, тоді користувачі зможуть оцінити та схвалити властивості та практичність цього прототипу. При збиранні, структуризації та визначенні пріоритетів потреб надзвичайно корисні таблиці вимог, оскільки до них можна додавати стовпці для визначення можливостей, які вже представлені в КІ, а також відображення результатів тестування вимог. При відстежуванні змін у вимогах необхідна більша дисциплінованість, ніж при початковій фіксації вимог.

Якщо функція або характеристика КІ передбачає вимірювання, звітність, відстежування та служить основою для дій, то її слід визначати як частину вимог до продукту.

Очевидні питання, які повинні увійти у вимоги:

- стиль КІ;
- платформа та інші стандарти КІ для додатку;
- сумісність з провідним ПЗ, працюючим на даній платформі;
- вміст екрану;
- поведінка екрану;
- характеристики зовнішнього вигляду екрану;
- методи взаємодії користувачів із системою;
- можливості роботи з клавіатурою;
- зворотній зв'язок користувача у відповідь на стан системи та час відгуку;
- користувацький контроль над різними функціями;
- запам'ятування результатів операцій розташування та зміни розмірів вікна, а також даних, стану та контекста;
- можливості навігації для додатку;
- збереження даних користувача при навігації;
- запам'ятування проміжних даних користувача при навігації;
- інтерактивне навчання, підтримка продуктивності та довідкова система;
- попередження помилок та відновлення системи після помилок;
- методи прямого введення для усунення діалогу;
- перевірка правильності значень полів, а також ідентифікація потрібних полів;
- стандартне використання кольору, індикаторів, графіки і т.д.;
- засоби забезпечення доступу для користувачів з фізичними вадами.

Вимоги по практичності легко встановлювати в невизначеному та важкому для вимірювання вигляді.

Стандартні метрики, які можуть бути корисними при фіксації практичності ПЗ: встановлення або оновлення; вивчення; ефективність; продуктивність; узгодженість; безпомилкове виконання задач; виконання задач без підказок; довідка; візуальні та мультимедійні можливості; дефекти; задоволеність системою; переваги.

Перелік найбільш важливих характеристик, пов'язаних з виконанням користувачем задач та потоками робіт:

- кількість підтримуваних переривань задач;
- функціональні можливості, які підтримуються під час переривань та навігації;
- підтримка КІ під час переривань.

Стандарти роблять наше життя простішим, розкриваючи характеристики об'єктів і систем, які нас оточують. Стандарти є всюди - це основа індустріалізації. Стандарти комп'ютерного проектування розробляються державними та суспільними організаціями, іншими локальними та

міжнародними формаціями. Найвідоміші організації по розробці стандартів - Американський національний інститут стандартів ANSI, Німецький інженерний стандарт DIN та Міжнародна організація по стандартизації ISO. Стандарти існують для дисплеїв, клавіатур, системних деталей та ін. Стандарти на ПЗ звичайно застосовні для основних характеристик КІ. Стандарти повинні постійно оновлюватись та вдосконалюватись, інакше вони починають гальмувати розвиток технологій та перешкоджати впровадженню новацій. Деякі з сьогоднішніх стандартів не повністю відповідають нинішньому комп'ютерному програмному та апаратному забезпечення, а також всім потребам користувача.

4.2. Керівні принципи та керівництва за стилем

Керівні принципи побудови інтерфейса розраховані на сьогоднішні системи введення та виведення інформації. Вони починають користовувати такі технології, які використання пера, написання від руки та голосове введення. Одна з проблем розробки інструкцій, які відповідають новим технологіям - це розшифрування способів взаємодії користувача з системою, оскільки ступінь цієї взаємодії ще точно не визначена. Інструкції повинні базуватись на тому, як користувачі реагують на новації та створюватись по проходженні деякого часу, необхідного, щоб користувачі засвоїли інтерфейс та склали певну думку про нього. Ніколи не встановлюйте стандартів і не пишіть інструкцій, не знаючи, як користувач працює з новітніми технологіями.

В керівництвах по елементах інтерфейсу та його органах управління сказано, коли їх потрібно використовувати, як "подати" і якою повинна бути техніка роботи з ними (клавіатурна чи за допомогою миші). Повний набір керівництв розкриває сутність кожного об'єкту та елементу інтерфейсу в термінах та способах представлення на екрані, їх поведінку, механізм взаємодії з ними користувачів.

Більшість програмних продуктів створені для роботи на різних платформах. З тих пір, як ці платформи мають різні операційні системи, інструменти та стилі інтерфейсу, дуже складно розробляти інтерфейс, який задовольняє всі платформи або навіть просто працюючий на кожній з платформ. Недавнє додавнення - підбір індустріальних керівництв по проектуванню - було розроблене Беллкором. Воно містить опис та керівні принципи для основних компонентів проектування КІ для таких компаній та операційних систем, як IBM CUA, OSF Motif, Microsoft Windows та ін.

Задачі керівних принципів по проектуванню однозначні: надати користувачам можливість доступу до інформації з будь-якого місця системи, в будь-якій формі, створити такий інтерфейс, який допомагав би людям працювати та подобався їм. Добре розроблений інтерфейс дозволяє

користувачам сфокусуватись на виконанні задач, а не на особливостях ПЗ та АЗ.

Найпростіші принципи по розробці ПЗ, сформульовані Берстом:

- для свого програмного середовища оберіть відповідні промислові керівні принципи;
- створіть корпоративне керівництво за стилем оформлення вашого програмного інтерфейса;
- орієнтуйтесь на кінцевий додаток, використовуючи дані керівних принципів;
- при розробці власного додатку та купівлі програмних продуктів слідуйте керівним принципам.

Стандарти, промислові керівництва по розробці та стилю оформлення в порядку спадання пріоритету:

- міжнародні стандарти;
- галузеві керівні принципи по платформах;
- корпоративне керівництво по оформленню стилю інтерфейсу;
- керівництво по оформленню стилю групи продуктів;
- керівництво по оформленню стилю продукту.

Цілі і керівні принципи розробки інтерфейсу повинні бути реалістичними і доступними для користувачів. Специфіка того чи іншого бізнеса накладе обмеження на ваше середовище. Ваші керівні принципи по розробці КІ також повинні проходити тестування. Щоб продукт відповідав керівним принципам, необхідно мати підтримку з боку розробників.

Сумісність системи та інтерфейсу в деяких випадках може протирічити очкуванням та побажанням користувачів. Не варто слідувати керівним принципам в області сумісності лише заради самої сумісності - спочатку потрібно правильно розробити будь- що, а вже потім послідовно застосовувати це до всього інтерфейсу. Розробка інтерфейса - більше мистецтво, ніж наука.

Розробка КІ на макрорівні представляє собою шаблон КІ - продукт збирається весь цілком та його концепція стає ясна користувачам по мірі взаємодії з ним.

Як тільки з друку виходять нові керівні принципи по розробці інтерфейсів, продавці інструментарію намагаються якомога швидше включити елементи і технології нових інтерфейсів в запропоновані ним інструменти. Нові інтерфейсні технології завжди є рушійною силою для розробки і проектування все більш досконалих інструментів.

Стандарти та керівні принципи є будівельними блоками, на які повинні спиратись всі зусилля по розробці та проектуванню. Однак це не значить, що кожен блок, побудований у відповідності до будівельних стандартів або програм містобудівництва, буде зручним для житла. Аналогічно продукти компонентного та програмного забезпечення не будуть зручними та

приємними у використанні тільки тому, що виконувались всі стандарти та керівні принципи. Стандарти та керівні принципи - частина загальної концепції проектування, яка включає принципи розробки, методологію розробки, тестування на зручність застосування, а також розуміння потреб користувачів.

Принципи, стандарти, інструкції та керівництва, які стосуються практичності та КІ, звичайно не розглядаються в документації по процесам створення ПЗ і не враховуються при плануванні розробки більшості продуктів.

Існує ряд керівництв за стилем КІ, створених поставщиками платформ ОС та/або броузерів. Ці керівництва за стилем для платформ носять загальний характер в тому сенсі, що не містять вказівок по досягненню специфічних та узгоджених результатів високого рівня для будь-якого додатку або пакету додатків, або для різноманітної сукупності додатків, функціонуючих на даній платформі.

Компанії витрачають значні зусилля, розробляючи внутрішні керівництва за стилем КІ, щоб компенсувати недоліки загальних керівництв. Більшість з цих зусиль не приносять бажаних результатів по відношенню до практичності, інтеграції та узгодженості застосовано до продуктів, розроблюваних незалежними колективами. Наслідком даної проблеми є зростання витрат на навчання, зниження продуктивності та почуття глибокого розчарування у користувачів, проектувальників, розробників та менеджерів.

Мета керівництва за стилем полягає якраз в тому, щоб допомогти розробникам бути послідовними.

Принципи - це загальні вказівки по проектуванню, які носять якісний характер та належать до категорії тверджень, яких корисно дотримуватись. Приклад часто виголошуваного принципу проектування - прагнення до простоти.

Потрібно обрати не більше 10 важливих принципів, а потім виробляти інструкції, які містять вимірювані оцінки для цих принципів, щоб вони стали діючими та практичними.

Інструкція (або керівна вказівка) - це правило проектування, корисне при реалізації та легко піддається вимірюванню в термінах відповідності. Виконання інструкцій легко піддається безпосередньому вимірюванню, вони об'єктивні і конкретні. В багатьох керівництвах за стилем інструкції приводяться як правила, а останні вказівки як рекомендації.

Стандарти - це керівні вказівки, які відбиваються безпосередньо в КІ операційної системи або яких дотримується велика кількість провідних організацій галузі при розробці додатків. Завжди потрібно слідувати галузевим стандартам, якщо немає твердої, заснованої на думці користувачів, впевненості в доцільності відхилень.

Керівництво за стилем КІ - це концептуальна та вельми високорівнева специфікація, яка описує загальний зовнішній вигляд, поведінку та

користувачьку взаємодію, а унікальні подrobiці, які стосуються КІ, опускаються. Керівництво за стилем слід розробляти, виходячи з потреб проекту, але при цьому прагнути уникнути незадоволення спонсорів, користувачів проекту та користувачів продукту.

Керівництво за стилем можна представити собі як високорівневий огляд або керівництво користувача, яке описує спосіб використання продукту або пакету додатків. Керівництво за стилем КІ містить принципи, інструкції та стандарти, застосовані при проектуванні продукту.

Наявність принципів, стандартів, інструкцій та керівництв за стилем необхідна, але недостатня умова поставки високоякісного ПЗ КІ. В загальному випадку, кращий підхід полягає в тому, щоб підтвердити обґрунтованість використання яких-небудь керівництв стосовно до конкретного продукту, спираючись на думку проектного колективу та кінцевих користувачів.

Традиційні керівництва за стилем КІ носять описовий характер, тобто дається загальна вказівка без конкретизації підсказаної директиви. Інструкції описового типу рекомендують розробнику КІ використовувати вікна з різними цілями. Однак керівництво описового типу слабо конкретизує інструкції або директиви по відношенню до того, як саме відобразжати об'єкт або дію всередині вікна або бажану поведінку при взаємодії. Розробник повинен виділити конкретні деталі, повністю прочитавши керівництво за стилем, сформувати бажаний стиль, а потім визначити конкретні деталі додатків, які виходять за рамки базисних властивостей КІ

4.3. Проблеми проектування міжнародних інтерфейсів

Появу програмного проектування та розробку інтерфейсів можна вважати подією міжнародного масштабу. Будь-який програмний продукт, розроблюваний сьогодні, має потенційно широку аудиторію користувачів з різних країн, вододіючих різними мовами та різними культурними звичаями. Цей факт одержав назву інтернаціоналізації, або NLS (National Language Support - підтримка національної мови), який став головною умовою міжнародного успіху програмних продуктів.

Сучасне програмування та інструментарій КІ дозволяють відокремити велику частину того, що користувач бачить і чим оперує на екрані (програмно закодовані елементи), від останньої частини програми (жорстко закодованої інформації). Це дає можливість без особливих зусиль представляти програми різними мовами.

Проблеми, які необхідно розв'язати при проектуванні міжнародних інтерфейсів:

- абревіатури;
- акроніми;
- описовий текст;

- виділення мнемонік;
- граматика персоналізації;
- двонапрямлені мови;
- довжина тексту;
- прописні букви;
- заголовки колонок;
- іконки;
- використання символів;
- використання кольору;
- клавіші швидкого доступу;
- комбінації клавіш;
- призначення функціональних клавіш;
- описувачі полів вводу;
- перші літери для навігації;
- символи з кодуванням в два байти (DBCS, Unicode);
- сортування інформації;
- формат грошових одиниць;
- формат телефонних номерів;
- формат розміру паперу;
- формат розмірів;
- формат адреси;
- формат дати;
- формат чисел;
- формат часу;
- гумор.

5. Користувачі, їх робоче середовище та задачі

5.1. Користувачі продукту, їх робота та середовище

Одна з головних причин створення невдалих продуктів полягає в недостатньому залученні користувачів до проекту. Не менше значення мають і наслідки недостатньо активної участі користувачів в проекті, зокрема відсутність глибоких знань про реальних користувачів продукту, середовище використання розробки та рівня застосування тих чи інших задач. Деякі продукти відрізняються досить конкретним колом користувачів та досить певним середовищем використання і задачами. Для інших проектів сукупність користувачів відрізняється великим розмаїттям поряд із значими відмінностями в середовищах і задачах. Детальна інформація про користувачів, середовища і задачі допомагає встановити рамки, в яких повинно здійснюватись проектування КІ продукта та забезпечуватись його практичність. Поряд з вимогами до КІ та практичності критична інформація про користувачів, середовища і задачі допомагає колективу розробників виділити ті особливості продукту, які бажають бачити реальні користувачі або яких вони потребують. Інформація про користувачів та їх потреби важлива для відповідного вибору методів створення КІ та підходів до проектування спільногого стилю додатків.

Приклади найбільш помітних характеристик користувача в зв'язку з його роботою на комп'ютері і з його відношенням до ПЗ:

- попередні знання та досвід, використовувані в області апаратного та програмного забезпечення;
- попередні знання та досвід, які стосуються підтримки задач прикладної області;
- фізичні та індивідуальні характеристики;
- фізичні та соціальні особливості робочого середовища.

Всім користувачам додогоди неможливо. Програмний продукт та його КІ не можуть повністю підходити всім користувачам. Не потрібно висувати надто багато припущення по відношенню до будь-якої групи користувачів, а також фізичних та соціальних умов для роботи.

Приклади необхідних проектних рішень:

- інтерфейс для новачків, випадкових користувачів та спеціалістів;
- поступове розкриття всіх можливостей;
- декілька методів підтримки продуктивності;
- можливості вибору користувацьких переваг;
- множинні представлення.

Критична інформація про користувачів включає статистичні розподіли для наступних характеристик:

- освіта, знання та досвід роботи з комп'ютерними системами та ПЗ, а також в прикладній області, для підтримки якої розробляється ПЗ;
- ставлення до роботи, бізнес-процедур та виконуваних задач;
- ставлення до підтримки апаратного та програмного забезпечення комп'ютерів;
- фізичні та особистісні характеристики;
- характеристики соціального та фізичного робочого середовища.

Інші фактори, які слід брати до уваги при вивченні користувачів з точки зору розроблюваного ПЗ:

- освіта;
- основна мова;
- навички введення даних та використання вказівника миші;
- зорові, слухові, мовні та інші види фізичних обмежень;
- вік і стать;
- переваги та стиль навчання;
- рівень навичок та досвіда користувачів ПЗ;
- мотивація.

5.2. Методи збирання інформації про користувачів

Основна мета аналізу полягає в тому, щоб подивитись, що відбувається в робочому середовищі, і визначити, в чому може полягати реальна допомога з боку програмної системи.

Методи залучення користувачів в проект:

- метод спостереження - спостереження за реальними користувачами, які виконують реальну роботу;
- метод опитування - опитуються всі учасники проекту, які виконують комплекс задач або входять в робоче середовище;
- метод індивідуальних інтерв'ю - неформальні інтерв'ю з реальними користувачами;
- метод групового інтерв'ю - формальні або неформальні інтерв'ю з колективом або групою реальних користувачів;
- інші джерела інформації.

Питання мають бути побудовані у вигляді тестів з певними варіантами відповідей.

Питання про користувачів: Як би ви оцінили ваш досвід роботи з комп'ютером? Як давно ви користуєтесь комп'ютером? Як би ви оцінили досвід роботи з існуючою системою?

Питання про існуючі системи: Які комп'ютери ви використовуєте і наскільки часто? Яку операційну систему ви використовуєте? Наскільки ви задоволені вашою системою?

Питання, що стосуються робочого середовища: Як би ви могли описати ваше робоче середовище? Як би ви могли описати ізольованість вашого робочого середовища? Яка ступінь задоволеності вашим робочим середовищем?

Питання, які стосуються роботи: Як часто зустрічається ця задача? Наскільки важка ця задача? Наскільки ви задоволені тим, як виконується ця задача?

Питання, які припускають різні тлумачення: Які зміни в комп'ютерних системах могли б зробити вашу роботу більш легкою або більш ефективною? Які зміни в бізнес- процесах полегшили б вашу роботу або зробили б її більш ефективною?

Аналіз слід продовжувати неперервно, постійно уточнюючи його висновки та результати.

5.3. Концептуальне проектування

Всі задачі, пов'язані з проектуванням та розробкою КІ, а також забезпеченням практичності продукту, надзвичайно важливі. Перша конкретна проектна задача - концептуальна і передбачає вибір вірних основних положень для прийняття проектних рішень. В багатьох відношеннях концептуальний проект КІ - це, по суті, його архітектура. Послідовні рівні проектних рішень дозволяють нарощувати визначення і ступінь деталізації зовнішнього вигляду, поведінки, інформаційної підтримки та користувачкої взаємодії на більш пізніх етапах проектування.

Всі кроки проектування в рівному ступені важливі. Концептуальне проектування, яке стосується концептуальних моделей, стилю КІ та навігаційних структур, не зупиняється на деталях зовнішнього вигляду, поведінки, взаємодії і на функціональних можливостях продукту. Перший вірний крок важливий для збільшення шансів на найменш короткий та безболісний шлях до досягнення цілей проекту у відношенні КІ, практичності, узгодженості, інформаційної підтримки, часу відгуку, надійності та інших проектних факторів.

З одного боку, альтернативні концептуальні рішення формулюються, оцінюються і піддаються ітеративному перегляду, а також дуже швидкому нарощуванню при невеликих витратах - на концептуальному рівні. З іншого боку, помилка в концептуальних проектних рішеннях, не помічена на ранніх проектних етапах розробки, передається на наступний етап і потенційно впливає на множину екранів, взаємодій, підтримку користувачів, програмний код, навчання, тестування, план-графік та вартість розробки. Вартість виправлення такої помилки буде тим вище, чим довше помилка залишається невиявленою.

Концептуальний проект являє собою попередній ескіз, який показує основні функції КІ, візуальні, інформаційні та інші характеристики проекту продукту. Архітектура КІ - це буквально розподіл роботи між системою та користувачем, а також програмний інтерфейс між ними.

На основі аналізу характеристик користувачів, вимог, задач та ділових потреб можна представити в явній і наочній формі основоположну ідею проекту по відношенню до візуалізації та взаємодії, організаційні принципи та структуру, а також типові поведінкові моделі ПЗ.

Концептуальний проект включає наступні компоненти:

- загальний стиль КІ;
- прикладний стиль КІ;
- сутності та артефакти кінцевих користувачів;
- передбачувана користувальська модель;
- організація, структура, потоки та відношення між сутностями кінцевих користувачів;
- ключові принципи функціонання.

Найбільш важливий аспект концептуального проекту - його потенціал з точки зору формування і бачення. Концептуальний проект встановлює те, що називається "враженням і відчуттям" від програмного продукту, а також дозволяє виявити, що із спроектованого та сконструйованого під "обгорткою" проекту служить для підтримки бачення. Бачення задає відчуття того, наскільки продукт логічно узгоджений та працює на підтримку користувальських задач.

Альтернативи, закладені в концептуальний проект, одержують наочність в специфікації та прототипі. Для одержання початкової інформації від користувачів застосовуються методи залучення користувачів до участі в проекті. Для прогнозування перспективних шляхів розвитку проекту застосовуються методи оцінки практичності.

Шляхи формування бачення:

- попередній аналіз - вивчення та аналіз вимог профілів, задач, робочих середовищ користувачів;
- методи сумісної розробки - опис методів залучення користувачів до участі в проекті;
- проектна ітерація I₁ - концептуальне проектування являє собою першу з множини проектних ітерацій, яка виражається в наступних діях: формулювання множини проектних альтернатив; опис кожної проектної альтернативи; оцінка кожної альтернативи по відношенню до вимог, середовищ і користувачів; вибір найбільш багатообіцяючих альтернатив та відкидання небажаних альтернатив; повторення перерахованих дій, доки вимоги не будуть задоволені;
- альтернативи використання базових властивостей КІ;

- альтернативи використання прикладних стилів КІ - електронний стиль (подібний до електронної таблиці, показує користувачу комп'ютерні сутності або елементи предметної області без фізичних границь), фізичний стиль (комп'ютерні сутності з використанням понять, які виводяться на основі їх представлення в реальному світі, але в рамках обмежень артефактів КІ подібно вікну), візуальний стиль (аналогічний фізичному стилю, який просунувся на один крок уперед - оформлення вікна відсутнє);
- об'єднання перестановок та сполучень стильових варіацій - потрібно напрацювати, як мінімум, три проектних альтернативи, використовуючи варіанти стилю КІ та прикладного стилю КІ: традиційне проектування; нетрадиційне проектування, яке розсуваве межі можливого; проміжний варіант проектування.

Основна задача, пов'язана з концептуальним проектуванням, полягає в тому, щоб встановити, які функції покладаються на користувача, а які - на систему. Результат розв'язку цієї задачі визначається наступними аспектами:

- задачі, які підлягають виконанню;
- користувачі, які виконують задачі;
- сильні сторони та обмеження можливостей користувачів;
- сильні сторони та обмеження апаратного забезпечення, системи, засобів КІ, додатку та стилів взаємодії, які належать до задачі;
- перспективні характеристики ПЗ, які покращують, нарощують або автоматизують окремі аспекти розв'язку користувацьких задач;
- перспективний шлях найменшого опору, який веде до досягнення заданих часових та вартісних обмежень;
- перспективи виконання вимог по відношенню до КІ та практичності.

В загальному випадку колектив розробників продукту розподіляє відповідні компоненти задач по системі та користувачам. На систему приходить вся робота, яку не можуть виконати користувачі, а на користувачів покладаються задачі, розв'язок яких проблематичний для комп'ютерного ПЗ.

Сприйняття роботи системи користувачем називається користувацькою моделлю. Цей погляд користувача на те, як виглядають події, які відбуваються в системі; така модель представляє ефективний спосіб вивчення та прогнозування поведінки ПЗ. Важливий аспект задач концептуального проектування полягає в посиленні цієї особливості вивчення та поведінки користувача для свідомого формуванні КІ, щоб стимулювати появу користувацької моделі, яка відповідає намірам проектировальника.

Користувацька модель включає наступні компоненти:

- поняття;
- можливості;

- аналогії і метафори;
- системні та користувальські об'єкти;
- зовнішній вигляд;
- поведінка, порядок відображення та відношення;
- взаємодії.

Мета проектного колективу полягає в формуванні моделі, яка задовільняє декільком критеріям:

- орієнтація на користувача;
- природність;
- узагальнений погляд;
- інтеграція;
- узгодженість;
- простота;
- продуктивність;
- інтуїтивна наочність.

6. Макети, моделі та прототипи інтерфейсів користувача

6.1. Макети, моделі і прототипи

В багатьох галузях при розробці продуктів поширенна практика якомога більш раннього надання їм наочного вигляду - те, що в індустрії програмних засобів називається візуалізацією. Подібна практика відіграє велику роль також при проектуванні та розробці КІ. Користувачі, ділові спонсори, проектний колектив та інші учасники команди по розробці продукту можуть на ранній стадії його створення поглянути на те, що ж реально виходить на шляху до поставки. Цей перший погляд дає можливість зацікавленим сторонам внести проміжні корективи. Для проектного колективу це ще один метод розробки та можливість перевірити правильність прийнятих рішень, перш ніж вносити зміни буде надто пізно або надто дорого.

Створення прототипу чи прототипування - термін, яким часто зловживають, який часто вживають неправильно. Незалежно від використованої назви або технічних особливостей мета проектного колективу полягає в візуалізації проектних рішень, які приймають одну або декілька форм. Підхід на основі прототипування можна охарактеризувати за допомогою таких визначень, як швидкість, точність, можливість повторного використання та скрупулезність.

Макет - це матеріалізований у вигляді набору статичних зображень проект. Поведінкові та динамічні стани представляються статичними зображеннями. Сукупність статичних зображень, яка демонструє поведінку системи під час прогону сценарію користувацької взаємодії, називається розкладовою.

Модель (імітаційна модель, імітація) представляє собою матеріалізацію проекту, вона будується з використанням засобів реалізації, відмінних від тих, які передбачається використовувати для розробки продукту.

Прототип представляє собою матеріалізацію проекту, яка побудована з щ використанням передбачуваних засобів розробки для продукту. Засоби розробки включають апаратні засоби, операційну систему та мови програмування.

6.2. Цілі візуалізації проекту

Візуалізація продукту представляє собою цикл поступового уточнення та нарощування проектних рішень у відповідь на питання, які виникають в ході розробки. На ранніх етапах розробки вимоги, проектні концепції та альтернативи досліджуються за допомогою підходів, які не вимагають великих витрат і дають швидку віддачу. Низькорівневі деталі проекту не піддаються глибокому аналізу. На даних етапах не обов'язково використовуються

передбачувані засоби розробки, однак це не сприяє проясненню того, як виглядатиме і працюватиме проект застосовно до передбачуваного середовища його використання. Використання планованих інструментальних засобів допомагає оцінити потреби в навчанні й часі, необхідні для розробки.

Для оцінки спрощених та обмежених представлень складних проектних рішень використовуються евристичні методи і методи сумісної розробки. Від членів проектного колективу та користувачів вимагається подолати концептуальні обмеження, накладені існуючими на папері макетами, розкладовою, моделями, а також очікувані проектні проблеми та обмеження, пов'язані з використанням цих засобів. Бажано якомога швидше створити спрощений прототип КІ, використовуючи передбачувані засоби розробки і компоненти. Це допомагає прояснити проблеми користувацької взаємодії, пов'язані з логікою, кроками роботи і обмеженнями інструментарію.

Методи прискорення розробки:

- складання прискореного план-графіка;
- швидкий перехід до прототипування;
- внесення виправлень безпосередньо по ходу проекту;
- припустимість жорсткого програмування;
- концентрація на головних задачах і методах, нехтування неістотними деталями.

При використанні методів візуалізації слід спрямувати основні зусилля на досягнення наступних цілей:

- візуалізація основних складових, які визначають рівень задоволеності користувачів;
- увага до дрібниць, які можуть стати основними перешкодами на шляху до досягнення задоволеності користувачів;
- оцінка значущості розглядуваних питань з точки зору проекту.

6.3. Методи матеріалізації проектних рішень

Основна проблема проектного колективу полягає в необхідності прийняття рішення про те, які аспекти проекту підлягають матеріалізації з метою їх завчасної оцінки сумісно з користувачами і які методи для цього застосувати. Що стосується великих продуктів, а також аспектів можливостей і практичності, пов'язаних з функціями і КІ, вони вимагають розгляду багатьох питань.

Будь-яка матеріалізація проектних рішень вимагає розгляду наступних технічних основ:

- часто використовувані, нестійкі, проблематичні задачі і сценарії, які підлягають представленню;
- часто використовувані і нестійкі функції, методи і дані;

- загальне використання платформенного стилю КІ;
- специфіка прикладного стилю КІ, екранів і логіки;
- проблематичні елементи управління КІ;
- потрібний рівень користувачької взаємодії;
- необхідний обсяг точності відтворення середовища;
- глибина фактичної матеріалізації в порівнянні з потребами візуалізації;
- швидкість представлення;
- цілі по відношенню до повторного використання застосовно до кожного виду візуалізації.

Існують аспекти ПЗ КІ, які не підлягають матеріалізації до їх реалізації:

- рідко використовувані гілки функційних можливостей;
- рідко використовувані властивості КІ;
- інформація про продукт;
- повідомлення;
- обробка помилок;
- аспекти ефективності.

Методи візуалізації:

- олівець і папір;
- розкадровка;
- імітація;
- спрощений прототип КІ;
- прототип КІ;
- функційно-інтерфейсний прототип;
- реалізація.

6.4. Відкидання прототипів.

Іноді для складних проектів необхідна розширення оцінка, яку не можна здійснити недорогими методами. Таким чином, немає нічого незвичного у виборі імітації та прототипування, які вимагають розробки тисяч рядків коду, щоб оцінити складні програмні пакети з точки зору питань, які стосуються численних екранів, алгоритмів, інтеграції та узгодженості. В результаті керівники бажають бути впевненими в тому, що після моделювання відкидається мінімальний обсяг готового програмного коду.

В подібних випадках починати слід з повторного використання як мети. Виходячи з навичок персоналу, інфраструктури, календарного плану, необхідно ідентифікувати найбільш ймовірні компоненти, придатні для повторного використання. Іноді доречно відкинути деякі плоди важкої праці.

7. Оцінка практичності користувальського інтерфейсу

7.1. Особливості оцінювання практичності користувальського інтерфейсу

Основний елемент розробки програмного продукту і КІ - тестування за участю користувачів. Якщо продукт задовільняє вимогам до практичності та іншим критеріям, він передається клієнтам. Якщо продукт не відповідає критеріям, він допрацьовується в ході чергової ітерації.

Практичність та загальний рівень задоволеності користувачів залежить від багатьох факторів. Практичність ПЗ - це функція можливостей, КІ, продуктивності, надійності, простоти іnstalляції та інформаційної підтримки продукту. Перераховані змінні є базовими факторами. Збір користувальських відгуків за кожним з цих напрямків важливий для причинного аналізу проблем та загальної оцінки практичності.

Оцінка практичності висуває наступні основні цілі:

- прогнозування користувальської задоволеності;
- аналіз підходів, пов'язаних з зовнішнім виглядом, поведінкою та користувальською взаємодією;
- виявлення та вирішення проблем;
- перевірка критеріїв;
- конкурентна оцінка.

Оцінки призначенні для визначення рівня короткотривалої та довготривалої практичності.

Методи оцінки практичності:

- перегляд (review) - неформальний та евристичний;
- наскрізний контроль (walk- through);
- лабораторні випробування за участю типових користувачів (lab- based test);
- випробування на території замовника за участю реальних замовників (in- your- house test);
- експлуатаційні випробування (field test).

При проведенні оцінок найбільшу роль відіграють наступні фактори:

- наперед визначені та об'єктивні критерії;
- структурні та об'єктивні методи;
- залучення користувачів до проведення оцінок;
- методи, які дозволяють визначити, що вірно, а що ні.

Ретельна підготовка до проведення оцінки та її виконання сприяють обґрутованому результату, а уміння вислухати користувачів може забезпечити стійкий та якісний зворотний зв'язок із ними. Незалежно від використовуваного методу при проведенні оцінки необхідно слідувати базовій

послідовності кроків для залучення користувачів та одержання надійних результатів.

Етапи підготовки оцінювання:

- план;
- проект;
- брифінг (інформаційна нарада), який передує оцінці;
- анкетування, яке передує оцінці;
- сценарії та задачі;
- анкетування по завершенню сценарію;
- анкетування по завершенню випробувань;
- збір інших даних (час, необхідний для завершення; кількість звернень за допомогою; помилки та проблеми користувача; коментарії та питання користувача; місця в сценарії, де користувач був позбавлений можливості продовжувати виконання задачі або допускав серйозні помилки);
- визначення суб'єктів тестування;
- тестова платформа;
- призначення координаторів тестування.

Крім планування та підготовки, проведення оцінки включає інші основні кроки. Після того, як матеріали і об'єкт тестування готові, бригада по оцінці продукту виконує наступні задачі:

- організація;
- пілотне тестування;
- анкетування до оцінки;
- виконання сценарію;
- зауваження до оцінки;
- порівняння;
- анкетування після сценарію;
- всеобщна оцінка.

7.2. Поняття зручності застосування програмного продукту

Зручність застосування є "kleem", який скріплює всі частини, які повинні з'єднатись разом, щоб скласти якийсь продукт. Тобто зручність застосування збирає в єдине ціле наступні частини: бізнес-процес; технологію; користувальський інтерфейс; електронну підтримку виконання задачі.

Зручність застосування повинна бути частиною проекту і проходить тестування в процесі розробки та проектування. Зручність застосування повинна мати операційне визначення (щоб її можна було виміряти) та забезпечувати тестування. Головне при тестуванні - правильно обрати методи, за допомогою яких воно проводитиметься. Проблема розробки якісного КІ та

проведення тестування на зручність застосування надзвичайно важливі та взаємозв'язані. Вдалий проект інтерфейсу ще не гарантує, що продукт буде зручно використовувати, і в той же час тестування за участю користувачів в жодній мірі не заміняє якісної розробки. Обидва питання складають частину процесу розробки інтерфейсу, яка називається проектування зручності застосування.

Професори та дослідники сходяться на думці, що в комп'ютерній галузі термін "зручність застосування" трактується надто вільно. Ці автори пропонують декілька категорій загального характеру, за допомогою яких можна дати чітке визначення терміну "зручність застосування" з точки зору комп'ютерного ПЗ в різних користувальницьких та системних середовищах. До цих категорій слід звернутись під час розробки анкет, контрольних списків або керівних принципів для оцінки програмних продуктів.

Категорії зручності застосування:

- легкий у використанні;
- легкий у вивченні;
- легкий в повторному вивченні;
- легко забувається;
- не приносить шкоди;
- легко підтримується;
- легко контролюється;
- легко розподіляється всередині групи;
- легко впроваджується в існуючі операції.

Раніше проектувальники вдосконалювали програмні продукти, додаючи до них додаткові функції. Для вдалого продажу продукту основний акцент робився на кількість функцій, які він мав, при цьому мало кого цікавило, як користувачі їх застосовуватимуть. В наш час проектувальники програм орієнтуються на задачі, які стоять перед споживачами. Подібний проблемно-орієнтований підхід повинен виконуватись при розробці КІ, включаючи систему електронної допомоги та технічну документацію.

Вже на початковому етапі розробки тестування на зручність застосування дозволить нам зрозуміти, чи зможуть споживачі взагалі використовувати цей продукт. Для відповіді на це питання необхідно одержати підтвердження того, що створений додаток виконуватиме потрібні користувачам функції та забезпечить їм додаткові переваги в бізнесі. Основні причини, чому тестування на зручність застосування так важливе:

- інтуїція розробників та проектувальників іноді може підвести;
- термінологія розробників і проектувальників не завжди співпадає з термінологією, до якої звикли користувачі;
- всі люди різні, тому в природі не існує "середньостатистичного" користувача;

- інструкції та керівні принципи по розробці питань зручності застосування не повні;
- інформації, одержуваної від користувачів по телефону або електронній пошті, недостатньо для проведення оцінки якості продукту;
- витрачені на проведення тестування час, гроші та ресурси завжди окупаються;
- продукти, створені частинами, звичайно бувають несумісними на системному рівні;
- проблеми, виявлені на завершуючих стадіях розробки, складніше і найскладніше виправляти;
- усунення помилок під час проектування дозволить скоротити витрати на наступну підтримку програми;
- оцінка зручності застосування може дати переваги перед конкуруючими продуктами.

Міжнародна організація стандартизації (ISO) дає наступне визначення: "Зручність застосування - це ефективність, рентабельність та задоволення, з яким користувачі можуть виконати ті чи інші задачі в заданому середовищі". Тестування на зручність застосування проводиться для того, щоб оцінити якість роботи продукту і вияснити, наскільки він ефективний, рентабельний та чи задоволені ним користувачі.

Тестування на зручність застосування здійснюється на різних етапах розробки продукту, щоб забезпечити зворотній зв'язок з користувачами. Це допомагає вдосконалювати весь проект в цілому, скорочує кількість помилок, проводить порівняльний аналіз продуктів і всерій, а також підтверджує відповідність продукту вимогам, які до нього висуваються. Тестування є частиною загального процесу розробки зручності застосування.

Способи проведення тестування:

- спостереження;
- проведення опитів та досліджень;
- контекстуальні опитування;
- евристичні оцінки;
- робота з виділеними групами;
- лабораторне тестування.

Методи оцінки зручності застосування:

- методи оцінки роботи, які передбачають підрахунок дій, визначення повноти виконання задачі, підрахунок часу, помилок та звернень по допомогу. Такі методи називають чисельними;
- суб'єктивні методи, які включають збір усних та письмових повідомлень користувачів про їх сприйняття, думки, судження, переваги, а також ступінь задоволеності системою та їх власну виконану роботу. Ці методи носять назву якісних.

7.3. Особливості тестування користувальських інтерфейсів

Перш ніж планувати і проводити тестування на зручність застосування продукту слід чітко визначити цілі і задачі, які стоять перед ним.

Бут (Booth) виявив 4 фактори, що складають зручність застосування: корисність, ефективність, простота вивчення та відношення користувача. Шекель (Shackel) теж розбиває зручність застосування на 4 схожих категорії: простота вивчення, ефективність, гнучкість та відношення користувача.

Цілі і задачі, які стоять перед розробкою зручності застосування, повинні визначатись для всіх програмних продуктів. Цілі - це забезпечення переваг продукту перед конкурентами в області простоти вивчення, ефективності, гнучкості і т.д. Цілі напряму пов'язані з чотирма факторами зручності застосування. Цілі самі по собі не підлягають безпосередній оцінці - вони повинні бути розділені на задачі, які представляють собою уточнення цілей, вони більш конкретні і детальні, їх можна оцінити і виміряти. Досягнення однієї цілі може вимагати розв'язку багатьох задач. Задачі повинні бути вибудувані таким чином, щоб містити інформацію по конкретних діях або операціях.

Користувачі хочуть одержати в своє розпорядження продукти передових або новітніх технологій, високої якості, прості у використанні та дешеві в розробці. Виникає питання: скільки коштує зручність застосування?

Кожен розробник і проектувальник повинен сам приймати участь або спостерігати за процесом тестування свого продукта. Однак потрібно намагатись, щоб в нього не втручались проектувальники, які надто добре поінформовані про роботу продукту. Розробники продукту повинні відігравати допоміжну роль і не замінити собою спеціалістів по моніторингу тестування.

Надійність тесту полягає в тому, що один і той же тест може проводитись знову й знову і демонструватиме однакові результати. Це в корні відрізняється від достовірності тесту. Тест вважається достовірним, якщо точно вимірює те, що повинно бути виміряно. Порівняльне тестування на зручність застосування продуктів може бути мало надійним, але не приймайте його достовірність "по номінальній вартості". Нескладно розробити сценарії і задачі, які гарно висвітлять технологію або функції одного продукту в порівнянні з аналогічними характеристиками іншого. Різні типи вимірювань, використовуваних при тестуванні, можуть кардинально змінити результати.

В останні роки було проведено ряд порівняльних аналізів, спрямованих на визначення переваг графічних КІ (ГКІ) над традиційними текстовими КІ (ТКІ) для виконання стандартних користувальських задач. Одним з найбільш відомих досліджень є звіт Wharton Report під назвою "The Value of GUIs", який пропонує 7 переваг ГКІ на основі одержаних результатів тестування. Вони полягають в тому, що користувачі ГКІ:

- працюють швидше;

- виконують більше задач;
- мають більш високу продуктивність;
- зазнають меншої розгубленості;
- менше втомлюються;
- мають більше можливостей для самонавчання та дослідження додатків;
- більш здатні до вивчення більшого числа можливостей додатку.

При порівняльному тестуванні операційних систем основну увагу приділяють наступним факторам:

- час виконання задачі в системі;
- показник успішності виконання задачі;
- продуктивність системи;
- ступінь задоволеності користувачів;
- переваги користувачів.

Висновки про порівняльні тести:

- порівняльні тести можуть надати корисну інформацію при дослідженії зручності застосування нових версій програмного продукту;
- слід використовувати задачі загального характеру і області відомих проблем в якості базових задач, щоб провести точні порівняльні вимірювання між версіями продуктів;
- слід застосовувати стандартні вимірювання зручності застосування;
- слід використовувати зовнішніх, незалежних виконавців для планування та проведення тестування на зручність застосування, щоб зменшити суб'єктивність сприйняття.

8. Графічний користувальський інтерфейс

8.1. Користувальський інтерфейс та операційні системи

Розглянемо історію розвитку КІ ПЗ - від командних інтерфейсів до меню і ГКІ, які ведуть до ООКІ. Розвиток стилів та технологій КІ йде паралельно еволюції ОС ПК, оскільки вони базуються на стилі та технології інтерфейса для апаратного забезпечення та операційної системи.

Сучасні ОС виходять далеко за межі ПЗ, розробленого лише для контролю за даними, які вводяться користувачами, та даними, які виходять з комп'ютера. На додаток до основного управління вони пропонують мережеві послуги, пристрой управління об'єктами, підтримку мультимедійних систем, електронну пошту і навіть оптимізацію апаратного забезпечення, анприклад, ущільнення диску.

Одним з найбільш важливих переваг сучасних ОС є підтримуваний ними стиль КІ, просуванню якого вони сприяють в рамках проектування продукту. ОС - програмний продукт, який має власний стиль інтерфейсу і є наочним товарним знаком системи. Інтерфейс ОС визначає парадигму КІ для ПЗ. Парадигма, за визначенням, представляє собою "виключно ясний або типовий приклад".

За кожним типом КІ розглянемо цілий ряд факторів. Важливо зрозуміти, як кожен з них слідує принципам розробки та керівним принципам. Такими факторами є:

- відповідність інтерфейсу користувальській моделі;
- відповідність можливостям пам'яті користувача;
- семантика стилю інтерфейса;
- взаємодія користувачів зі стилем інтерфейса.

8.2. Інтерфейс командного рядка

Інтерфейс командного рядка (ІКР) представляє собою оригінальний стиль взаємодії людини з машиною. Користувачі набирають запити або вказують дії формальною мовою, яка має власний словник, значення і синтаксис. Як правило, це набір команд, основних інструкцій до ОС. Будь- яка програма може використовувати ІКР для власного продукту. ІКР в найменшому ступені підтримує користувачів. Одна з основних проблем ІКР полягає в тому, що він не захищає ні ОС, ні програму від користувача. Для роботи з системою користувачам потрібно знати, як комп'ютер працює і де знаходяться їх програми і дані. Модель, застосовувана в ІКР, є моделлю програміста, а не користувача. Для багатьох засвоєння ІКР схоже на вивчення іноземної мови. Комп'ютерне ПЗ та АЗ має власну мову, знайому розробникам, але комп'ютери

повинні спілкуватись з користувачами мовою, доступною не лише програмістам.

Сучасні програмні системи можуть надавати інформацію способами, зрозумілими користувачу, який не вникає в питання функціонання комп'ютера. ІКР недосконалі в цьому смислі, але їй вони мають свої переваги. ІКР є дуже потужнім, гнучким і повністю контролльованим користувачем інтерфейсом. Однак цими перевагами можуть скористатись лише ті користувачі, які обирають модель програмістів. Якщо ІКР користуються проектувальники або професіонали в галузі програми або ОС, то вони здатні оцінити його по достоїнству.

ІКР не дотримується одного з головних принципів розробки КІ - зменшити завантаження пам'яті користувача. При проектуванні ІКР необхідно, щоб мова команд була зрозуміла та сприйнятлива для користувачів. Проектувальники повинні визначити функції, якими вони збираються забезпечити інтерфейс, і команди, які забезпечують доступ до них. Основна проблема ІКР - семантична (команди зрозуміти непросто).

Стиль ІКР передбачає, що користувачі мають клавіатуру для введення команд. Взаємодія користувачів з ІКР частіше передбачає операційно-об'єктно-орієнтований стиль синтаксису. Більшість команд являють собою дієслова або назви операцій, які користувачі планують виконувати. Спочатку користувачі набирають цю команду, потім назив об'єкта чи файла, над яким планують провести операцію. Подібний синтаксис команд забезпечує високоефективну взаємодію, хоча процент помилок при цьому також значий. Команди повинні набиратись точно у відповідності до керівництва. Ще одна проблема ІКР - непослідовність в зворотному зв'язку системи. Синтаксис взаємодії за допомогою командного рядка може бути вдосконалений за рахунок підказок, що зробить спілкування більш інтерактивним.

Переваги ІКР:

- потужний і гнучкий вид взаємодії для досвідчених користувачів;
- гнучкий інтерфейс, простота комбінування команд і параметрів;
- взаємодія, контролювана самими користувачами;
- використання мінімальної поверхні екрану;
- швидкодіючий та ефективний інтерфейс для знаючих користувачів;
- можливість використання скорочених назв команд;
- необхідність мінімального обсягу пам'яті;
- можливість використання поряд з іншими КІ.

Недоліки ІКР:

- мало (або повна відсутність) підказок та інструкцій на екрані;
- покращення інтерфейсу невидимі або невідомі;
- потрібно роздруковане керівництво або електронна допомога;
- необхідне знання користувачами системи, програм і даних;

- відсутній зворотній зв'язок та інформація про стан виконання задачі;
- необхідні навички в наборі тексту;
- необхідно запам'ятовувати команди та синтаксис;
- складність у вивченні;
- відсутність смислу в назвах команд;
- складність в розумінні синтаксису команд;
- неможливість відлагодити команди і синтаксис.

8.3. Інтерфейси меню

Меню являють собою перелік можливих дій, відображуваних на екрані або в вікні для користувачів, з метою вибору потрібних варіантів. Меню дозволяють користувачам пересуватись всередині системи, надаючи маршрути і можливість вибору потрібних пунктів або варіантів виконання операцій. Меню є дуже важливою частиною КІ, як графічного, так і об'єктно-орієнтованого. Стилі меню розвиваються.

Меню бувають різними за формою, розмірами і стилем. При повноекранному меню доступні функції або задачі представлені у вигляді списку на екрані. Основною визначальною характеристикою більшості повноекранних інтерфейсів меню є їх ієрархічна будова, тобто при виконанні задачі користувачам необхідно пересуватись деревовидною структурою меню.

Більшість сучасних графічних програм використовують панель меню, розташовану в верхній частині екрану або вікна. Такий тип меню забезпечує швидкий доступ до меню, а користувачі працюють всередині екранів або вікон програм. Панель меню представляє собою динамічний список основного набору операцій або пунктів, які ведуть користувача до інших пунктів, представлених в окремому спадному меню. Панелі меню є невід'ємною частиною всіх основних ГКІ і розроблених для них програм.

Основною характеристикою меню є їх здатність динамічно змінюватись. Це забезпечує користувачів підходящими варіантами і маршрутами для виконання поточних задачі або вибору потрібних об'єктів. Пункти меню можуть виділятись сірим кольором, якщо деякі операції на даний момент недоступні.

Іншими достатньо популярними типами меню є панелі інструментів і палітри. Панель інструментів представляє собою графічне меню операцій програми, інструментів і опцій, які користувачі можуть розташувати в будь-якому місці на екрані комп'ютера.

Останнім типом меню, розробленим для середовища КІ, є спливаюче або контекстне меню, оскільки його вміст залежить від контексту задач, які стоять перед користувачем. Вони можуть використовуватись для будь-якого типу додатків, як графічного, так і повноекранного інтерфейса. Спливаючі меню

містять лише ті варіанти, які відповідають поточному або обраному пункту. Розробляються вони для будь- якого елементу інтерфейса. Такі меню містять невеликий набір часто використовуваних дій, які можна знайти і на панелі меню системи чи програми.

Переваги меню:

- користувачам не потрібно пам'ятати набір команд;
- відмова від використання клавіатури зменшує кількість помилок;
- навігація по ієрархії проста для нових або випадкових користувачів;
- скорочує час навчання користувача;
- легко відстежувати відповіді і коригувати помилки;
- необхідний мінімальний обсяг друку;
- може бути використане сумісно з іншими типами інтерфейсів;
- елементи меню та їх розташування можуть налагоджуватись користувачем;
- гнучка система вибору команд (мнемоніки, миша, "гарячі" сполучення клавіш);
- підтримує тип пам'яті, орієнтований на визначення.

Недоліки меню:

- невідповідності або неефективність для деяких користувачів або задач;
- потрібна техніка навігації і вибору;
- необов'язково робить інтерфейс зручним;
- займає багато місця на екрані;
- змушує користувача пересуватись через багато рівнів підменю;
- вимагає часто оновлювати екран;
- користувач повинен розуміти принцип групування та ієрархії меню;
- вимагає деяких знань про систему;
- користувач може заблукати в ієрархії меню;
- терміни і найменування в меню незрозумілі користувачу;
- структура меню слідує за синаксисом "об'єкт- дія";
- синтаксис меню повинен бути послідовним;
- використання режимів змушує користувачів стежити за системою.

8.4. Основні властивості графічного користувацького інтерфейсу

Основною характеристикою ГКІ є використання ряду робочих елементів. ГКІ є графічним представленням на екрані комп'ютера інтерактивної взаємодії з програмами, об'єктами і даними. Він забезпечує користувачів потрібним інструментарієм і додатками, а не просто списком функцій.

Характеристики ГКІ:

- має растровий дисплей з високою роздільною здатністю;

- має вказуючий пристрій (миша);
- підтримує ідею сумісності між програмами;
- користувачі можуть бачити графічні зображення і текст на екрані в тому вигляді, в якому вони будуть роздруковані;
- слідує концепції інтерактивної взаємодії "об'єкт- дія";
- дозволяє переміщувати інформацію між програмами;
- надає можливість прямого маніпулювання об'єктами та інформацією на екрані;
- пропонує стандартні елементи інтерфейсу (меню та діалогові вікна);
- забезпечує візуальне відображення інформації і об'єктів (іконки і вікна);
- забезпечує візуальний зворотній зв'язок по ходу виконання користувачами дій та задач;
- дає візуальне відображення дій користувача/системи, а також режимів (меню, палітри);
- використовує графічні керуючі елементи, які дозволяють користувачам робити вибір та вводити дані;
- дає користувачам можливість налагодити та персоналізувати інтерфейс та інтерактивні дії;
- забезпечує гнучкість переходу від клавіатури до іншого пристрою введення даних.

Під час досліджень зручності застосування ГКІ було окреслено коло задач, які найбільш часто розв'язуються користувачами різних рівнів підготовки: запуск програми, виклик підказки, відкриття файлу, збереження файлу, копіювання файлу, зміна кольору робочого столу, знаходження файлу, запуск ще одного додатку, видалення файлу/папки, перейменування файлу/папки, вибір принтеру, створення папки, перегляд черговості завдань принтера, розташування документу на сервері, закриття програми, відхилення видалення/відновлення файлу, перевірка ресурсів системи.

Знання для роботи з ГКІ:

- уявлення користувачів про апаратну і програмну конфігурацію їх комп'ютерної системи;
- уявлення користувачів про графічні об'єкти та методи роботи з ними;
- знання користувачів про основні елементи вікон, щоб вміти маніпулювати самими вікнами та інформацією, яка в них міститься;
- знання користувачів про керуючі елементи ГКІ (панель меню, панель інструментів, радіокнопка, команда кнопка, палітра інструментів, прапорець, смуга прокрутки і т.д.);
- вміння користувачів маніпулювати мишею.

8.5. Архітектура графічного користувачького інтерфейсу

Концепція "об'єкт- операція" інтерактивної взаємодії користувача з інтерфейсом - це новація, втілена в ГКІ. Користувачі спочатку обирають об'єкт, а потім дію, яку необхідно виконати над днаим об'єктом. Це застосовано до всіх об'єктів інтерфейсу, включаючи іконки, вікна та об'єкти всередині вікон. Коли розробники дотримуються даного підходу до інтерфейсу, то вони розвивають користувачьку ментальну модель інтерфейса. Альтернативою йому є підхід "операція- об'єкт", який застосовувався в більшості ІКР та багатьох інтерфейсах на основі меню.

Одним з недоліків ГКІ є його орієнтація на додатки. При роботі з ГКІ користувачі біжать інформацію на екрані, в них складається враження, що вони дійсно працюють з об'єктами, однак увага користувачів сконцентрована на додатках. Вони обирають додаток, вказують файл даних, який хотять використовувати, організовують додатки і дані на комп'ютерах в формі графічних деревовидних структур, використовуючи диски і папки.

ГКІ передбачає проблемно- орієнтований підхід - перш ніж виконувати основну роботу з файлами, користувачі повинні запустити програму. Користувачі працюють з додатками у вікнах, які мають панель меню з варіантами маршрутів, відображеннями спадними меню. Такі меню містять варіанти дій і маршрутів, пов'язаних з об'єктами у вікні або додатком, який надає саме вікно.

ГКІ втілюють на екрані користувачькі ментальні моделі, якщо продукти добре продумані і якісно розроблені. ГКІ можуть навчати та розважати, а не лише допомагати в роботі.

Будь- який вдало розроблений КІ програми повинен знижувати навантаження на пам'ять користувача. ГКІ здатні запропонувати наочні підказки та необхідні відомості, використовуючи комп'ютерні потужності для зберігання та пошуку інформації.

Меню є головним фактором успіху ГКІ. Незважаючи на розмаїття ГІ, меню завжди будуть одним з основних елементів нагадування. Графічним меню повинні бути забезпечені більшість об'єктів ГКІ, в тому числі вікна і іконки.

Графічні керуючі елементи дозволяють обирати пункти та властивості, а також ініціювати виконання дій. Одним з найбільш важливих і креативних аспектів розробки інтерфейса є вибір необхідних керуючих елементів. Деякі з них представляють інформацію у вигляді списків і варіантів, що значно скорочує її обсяг. Інші змушують користувачів запам'ятати інформацію ьта потім вірно її друкувати.

Рядок статусу та поля інформації визначені в якості елементів вікна і повинні використовуватись для інформування споживачів про те, що

відбувається всередині програми, тобто служать візуальним зворотнім зв'язком. Рядок статусу інформує про стан додатку та вміст всередині вікна.

Існує чимало інших візуальних технологій, які нагадують користувачу про те, в якому режимі він знаходиться, або про поточний стан системи.

Кожен керуючий елемент, кожна іконка, кожен колір повинні мати певне значення, служити конкретній меті. Семантичний зворотній зв'язок нагадує про змістову характеристику об'єкта або виконання якоїсь дії. Схожі елементи повинні поводитись аналогічним чином.

Методи інтерактивної взаємодії ГКІ включають як клавіатуру, так і вказуючі пристрої для переміщення, вибору та прямого маніпулювання інформацією на екрані. ГКІ сполучають 3 основних стилі інтерфейсу: командний рядок, меню, пряме маніпулювання. У використанні клавіатури і миші повинна бути золота середина. Тільки користувачам повинно належати право вибору.

До методів інтерактивної взаємодії ГКІ належать:

- право користування клавіатурою і мишею в залежності від переваг користувача;
- робота з різними типами меню;
- 2 режими редактування: вставка або заміна;
- технологія drag- and- drop;
- буфер обміну.

ГКІ залишатиметься масовим інтерфейсом ще декілька років. Орієнтовані на користувача вдосконалення будуть продовжувати покращувати операційні системи та програми по мірі зростання конкуренції серед виробників ПЗ.

9. Етапи розробки користувальського інтерфейсу

9.1. Етапи розробки користувальського інтерфейсу. Ітераційна природа розробки

Розробка, навіть з точки зору простої зручності, вимагає команди спеціалістів, які мають таланти в абсолютно різних областях.

Якщо просто слідувати принципам проектування, керівництвам і стандартам, то це абсолютно не означає, що буде створено зручний інтерфейс. Не існує універсального способу розробки та проектування, який гарантує успішний кінцевий продукт. В будь-якому випадку потрібно дотримуватись колективного підходу.

Без сумніву, проектування і розробка вимагають навичок в області конструювання і створення ПЗ. Крім того, не здивими в команді будуть графічні та промислові розробники; спеціалісти по психології, які розбираються в пізнавальних та моторних здібностях людини; професіонали, які займаються написанням технічної документації; спеціалісти по тренінгу, знайомі з проблемами організації праці; а також люди, компетентні в питаннях пристройів введення, технологій відображення, інтерактивних методів, діалогового проектування та методології розробки. А оскільки в інтерфейсах все частіше застосовуються звук, голос, відео, анімація та тривимірна графіка, доводиться залучати й спеціалістів з цих областей.

Ідеальна команда для розробки програми повинна мати наступні навички:

- проблемний аналіз;
- програмування;
- розробка КІ;
- розробка команд;
- графічне проектування;
- написання технічної документації;
- тестування на зручність застосування.

В минулому розробка ПЗ та КІ розвивались лише за рахунок еволюції технологій та систем, на базі яких програми будувались. Це називалось системно-керованою або технологічно-керованою розробкою. Інтереси користувачів абсолютно не враховувались. Їм пропонувались програмні функції з інтерфейсом, який розробники були в стані запропонувати. З початку 80-х років акцент було перенесено на розробку, орієнтовану на користувача, причому до розробки залучаються і самі користувачі. Однак їм відводилася пасивна роль: у них виясняли, які вимоги вони висувають до комп'ютера і які задачі вони вирішуватимуть за його допомогою. Зараз більшість розробників дотримуються нових методологій, названих розробкою з залученням

користувачів та розробкою, орієнтованою на користувачів, що навчаються. Новий підхід полягає в тому, що на користувачів потрібно дивитись як на активних учасників самого процесу розробки. Подібне залучення користувачів до розробки сприяє демократизації, а також служить гарантією, що одержана комп'ютерна система буде відповідати запитам та вимогам користувачів.

Розробка, орієнтована на користувачів, що навчаються, спрямована на те, щоб в процесі вирішення своїх задач людина навчалась новим навичкам роботи з комп'ютером, тобто на її інтелектуальний розвиток, тренування її уявлення і одержання знань в різних областях.

Розробка, орієнтована на користувача, заснована на наступних керуючих принципах:

- розуміння потреб користувачів є рушійною силою всього проекту;
- все, що користувачі бачать і до чого торкаються, повинно проектуватись сумісними зусиллями;
- інноваційний проект завжди є результатом інтенсивної роботи команди спеціалістів в різних областях;
- конкурентоздатний проект вимагає постійного акценту на змагання;
- проект, затверджений користувачем, керує розробкою коду;
- рішення, що приймаються, повинні базуватись на зворотному зв'язку з користувачами;
- інформація від зворотного зв'язку з користувачами повинна збиратись часто, з науковою точністю та швидкістю;
- зворотній зв'язок здійснюється як з потенційними, так і з вже існуючими клієнтами;
- послідовно повинна стандартизуватись і впроваджуватись розробка, орієнтована на користувача;
- розробка, орієнтована на користувача, повинна постійно вдосконалюватись.

Проектування КІ може здійснюватись як окремо, так і сумісно з останнім процесом розробки продукту.

Зараз більшу увагу приділяють елементам інтерфейсу і об'єктам, які сприймаються і використовуються користувачами, а не функційності програм. В багатьох проектах власне розробка КІ та програмування продукту ведуться паралельно, особливо на ранніх стадіях. На більш пізніх етапах враховуються вимоги КІ та зворотнього зв'язку, які виявляються в результаті тестування на зручність застосування.

Процес складається з 4 основних етапів:

- збирання і аналіз інформації від користувачів;
- розробка КІ;
- побудова КІ;
- підтвердження якості створеного КІ.

Даний алгоритм може використовуватись як при розробці ООКІ, так і при проектуванні традиційних проблемно-орієнтованих інтерфейсів або ГКІ. Цей процес не залежить від матеральної та програмної платформ, ОС, застосованого інструментарію.

Будь-який вдалий процес розробки КІ повинен бути ітераційним. Словник Webster New Collegiate Dictionary дає наступне визначення слову “ітераційний” - “комп'ютерна процедура, де повторення циклу операцій дає результат, все близчий до шуканого результату”. Отже, вдалий інтерфейс неможливо одержати без періодичного повернення до попередніх етапів. Критерієм для завершення ітераційної розробки повинен служити той факт, що всі вимоги користувачів задоволені, а сам продукт відповідає запланованим цілям.

Може здаватись, що ітераційний процес займає багато часу через чисельні проходження етапами розробки. Але, якщо діяти вірно, то початкові проходження етапами допомагають створити варіанти розробок та прототипів, які в наступних ітераціях зекономлять час на впровадження та тестування.

9.2. Перший етап: збір та аналіз інформації від користувачів

Починати потрібно саме з цього — з користувачів. Перш ніж приступити до розробки та побудови будь-якої системи, потрібно вияснити, які проблеми споживачі або користувачі хочуть вирішити, і як вони звикли працювати. Потрібно спостерігати за користувачами, розпитувати їх. Слід звернути увагу на те, які обмеження накладаються їх комп'ютерними системами на технічне та програмне забезпечення. Потрібно весь час пам'ятати, що запропоноване рішення повинно відповідати не лише теперішнім, але й майбутнім потребам користувачів. Існує ряд ключових питань, які слід поставити на етапі аналізу інформації від користувачів. Проектування та постановка питань, а також проведення аналізу є справжнім мистецтвом. Потрібно бути дуже уважними при опитуванні користувачів та аналізі їх відповідей.

Перший етап — дії по збиранню та аналізу інформації — може бути розділений на 5 кроків:

- визначення профіля користувача — профіль користувача відповідає на питання “Що представляє собою ваш користувач?”, він дозволяє скласти уявлення про вік, освіту, переваги користувача, одержати іншу необхідну інформацію;

- аналіз задач, які стоять перед користувачем — це визначення того, чого хочуть користувачі і яким чином вони збираються вирішувати свої задачі;

- збирання вимог, наданих користувачами — відповідає на питання “Яку, з точки зору користувача, користь принесе їм пропонований продукт чи інтерфейс?”; практично в усіх проектах ПЗ враховуються вимоги користувачів, що допомагає визначити особливості проекту та структуру КІ. Щодо збирання вимог, то існують деякі спільні для всіх користувачів вимоги: скорочення

роботи з папером, зменшення помилок користувача, автоматизація існуючих ручних процесів, підвищення швидкості здійснення транзакцій;

- аналіз робочого середовища користувачів — відповідає на питання “Де користувачі вирішують задачі, які стоять перед ними?”, тобто потрібно визначити характеристики середовища, які можуть впливати на виконання користувачами своєї роботи. Потрібно зібрати інформацію щодо фізичного боку робочого середовища (освітлення, шум, робочий простір, температура, кількість персоналу), місця роботи користувача та ступінь його мобільності (офіс, квартира, стаціонарно, з пересуваннями), питань ергономіки та умов праці (чи задіються зір, слух, робота ведеться стоячи/сидячи), особливих запитів (рівень підготовки, фізичний стан, інтерес до пізнавального процесу, особливості мови та можливі недоліки), інтернаціоналізації та інших культурологічних умов (переклад, кольори, іконки, текст, повідомлення);

- відповідність вимог користувачів задачам, які стоять перед ними — перевірка на реалістичність вимог; якщо вимоги користувача на співрозмірні виконуваним задачам, то потрібно запропонувати їм оптимальний варіант; слід перевірити, чи не перевищують можливості продукту дійсні потреби клієнта.

Ітераційний метод передбачає повернення до етапу аналізу вимог користувача, щоб перевірити, чи не змінились в процесі проектування і розробки профіль користувачів, задачі, характеристики середовища або самі вимоги. Для побудови якісного продукту потрібно періодично повертатись до первого етапу, щоб поновлювати відомості про користувачів.

9.3. Другий етап: розробка КІ

Розробка КІ для програмного продукту звичайно вимагає значних витрат часу і ресурсів. Етап розробки складається з певних кроків, виконуваних в заданій послідовності. Існує велика спокуса почати програмування фінальної версії продукту вже зараз, не займаючись розробкою інтерфейсу. Однак потрібно пройти всі етапи процесу розробки, перш ніж перейти до програмування.

Розробка включає в себе наступні кроки:

- визначення цілі з точки зору зручності застосування продукту — на ранніх стадіях розробки продукту потрібно точно визначити, що він зможе зробити для користувачів; цілі розробки найкраще сформулювати в термінах, які характеризують дії користувача. 4 області, найбільш підходящі для встановлення цілей і задач: придатність, ефективність, легкість в засвоєнні, оцінка користувачами якості продукту;

- розробка задач і сценарію дій користувачів — сценарій є описом дій, виконуваних користувачем, це послідовність задач, які стоять перед користувачами, або подій, спрямованих на досягнення єдиної цілі; слід розробити декілька сценаріїв користувача, причому чим більше їх буде, тим

менше ймовірність, що буде пропущено ключові об'єкти чи операції, необхідні в інтерфейсі;

- визначення цілей та операцій інтерфейсу — найскладніший і найважливіший крок; на даному кроці потрібно: виділити об'єкти, дані і дії з сценаріїв та задач, які стоять перед користувачами; переглянути і уточнити список об'єктів і дій сумісно з користувачами; накреслити діаграму взаємодії між об'єктами; заповнити матрицю прямого маніпулювання об'єктами;

- визначення іконок об'єктів та візуального представлення — визначення, як найкраще представити визначені об'єкти на екрані, в якому вигляді їх побачать користувачі, яку інформацію вони міститимуть. При визначення представлень об'єктів слід враховувати спосіб спілкування користувачів з кожним з об'єктів і з інформацією, яка міститься в них;

- розробка меню об'єктів і вікон — вияснення, як користувачі будуть спілкуватись з визначеними і розробленими об'єктами і вікнами. Слід відповісти на наступні питання: Які дії властиві кожному об'єкту і типу?, Що міститься у спливаючих меню?, Яким вікнам потрібна панель меню?;

- оптимізація візуальної розробки.

9.4. Третій етап: побудова КІ

При першому зверненні до ітераційного процесу розробки потрібно швидше створювати прототип, ніж будувати КІ. Прототипування є виключно цінним способом створення перших проектів і демонстрації продукту, особливо на ранніх етапах тестування на зручність застосування.

Мета прототипування — швидко і легко візуалізувати різні альтернативні варіанти розробки, а не створювати код, який повинен стати частиною продукту.

Необхідно слідувати 3 “золотим” правилам при використанні прототипів як частини процесу розробки інтерфейсів:

- прототипуйте на ранніх стадіях і не забувайте про ітераційний принцип розробки;

- створюйте різні альтернативні варіанти;
- будьте готові викинути код прототипу.

9.5. Четвертий етап: підтвердження якості КІ

Тестування на зручність застосування є ключовим елементом ітеративного процесу розробки. Воно полягає в тому, щоб видати продукт на руки великій кількості користувачів і подивитись, чи зможуть вони з ним працювати. Мета тестування на зручність застосування повинна полягати в оцінці поведінки, дій і ступеня задоволеності користувачів. Більшість розробників звертаються до такого виду тестування близче до кінця проектування. Однак це надто пізно,

щоб на основі його результатів вносити зміни. Навіть якщо вони й вносяться, не можна бути впевненими в тому, що виправлений продукт можна використовувати без проведення повторного тестування.

Розробники повинні обов'язково бути присутніми при проведенні тестування. Тоді вони зможуть побачити, як користувачі працюють з їх продуктами. Однак вони не повинні мати змогу здійснювати технічну підтримку користувачів при тестуванні.

10. Інструментарій розробника інтерфейсів

10.1. Інструментарій розробника. Передача інформації візуальним способом.

До інструментарію розробника можна віднести: способи передачі інформації, використання кольору, звуку і анімації, термінологія та міжнародне проектування, визначення керуючих елементів інтерфейсу, проблеми, пов'язані з прямим маніпулюванням, а також розробка і схеми вікон.

Призначення будь- якого інтерфейсу — забезпечити комунікацію між двома системами, в даному випадку людьми та комп'ютерами. Практичні поради і рекомендації по розробці ПЗ та КІ засновані на традиційних способах, використовуваних при спілкуванні людей. Області візуального та графічного проектування базуються на знанні теорії мистецтва та візуального сприйняття.

Графічне відображення інформації надзвичайно важливе. Кожен рядок, кожен керуючий елемент, блок, частина тексту, колір і рисунок, які з'являються на екрані, впливають на користувача як поокремо, так і в комбінації з усім іншим.

З додаванням на екран кожного елементу візуальний вплив на користувача посилюється.

При роботі з візуальним представленням інформації слід розглядати 2 фактори:

- графічна перевага — складається з комплексних ідей, зв'язаних з ясністю, точністю та ефективністю;
- графічна цілісність — представляє собою використання графічних засобів для точного відображення даних.

Колір є найменш використовуваним елементом в розробці КІ, хоча проведені дослідження і дані рекомендації по застосуванню кольору як в апаратному, так і в програмному забезпеченні комп'ютерних систем.

Колір в КІ повинен застосовуватись дуже акуратно, адже люди мають різноманітні фізіологічні, психологічні, культурні та емоційні реакції на кольори. Часто він використовується лише в якості декоративного елементу. Це обмежує його здатність відображати значущу інформацію в інтерфейсі. Один з основних принципів використання кольору — не нашкодити.

В нашій свідомості кольори асоціюються з емоційним станом. Люди мистецтва знають, що теплі кольори (червоний, оранжевий та жовтий) збуджують і активують, а холодні (синій, фіолетовий, бордовий, сірий) мають заспокійливу та миротворчу дію.

Використання кольору в КІ може викликати труднощі у людей, які мають деякі проблеми із зором (кольорова сліпота — дальтонізм). Даний тип зорового дефекту присутній в 8% чоловічого населення та менш 1% жіночого населення. Враховуючи, що користувачами комп'ютерів є люди різного віку та стану

здоров'я, розробникам доводиться враховувати багато нюансів при створенні апаратного забезпечення монітора та розробці ПЗ екрану.

Колір часто застосовується для якісного виділення, представлення різниць в характеристиках та для кількісного виділення, щоб показати різниці в кількості. Кольори надзвичайно корисні, оскільки надають інформації додаткового значення. Оскільки колір є сильним засобом фокусування уваги, велика кількість відтінків змусить користувачів звернути увагу на екран. Це допомагає зробити інтерфейс більш дружнім та легким у використанні. Однак даний «ефект Лас- Вегаса» може відволікати користувачів при роботі.

ОС пропонують стандартні кольорові схеми і палітри, які й слід використовувати при розробці інтерфейсу. Вони були створені, тому що кольори складно узгоджувати, і люди часто помиляються при виборі кольору або кольорової комбінації.

Колір фокусує увагу людини. Ця властивість інтенсивно використовується в області комунікацій і реклами. Статистика показує, що колір на 32% ефективніше для фокусування уваги і на 25% - для виклику зворотніх дій користувача. Це дій практично на рівні підсвідомості. Колір часто застосовується для фокусування уваги, і при цьому не враховується, чи підходить він для тривалої роботи за комп'ютером.

Знання теорії сприйняття та особливостей зору призвело до створення дуже важливих фундаментальних рекомендацій по використанню кольору незалежно від засобу представлення інформації. Дані рекомендації застосовні до використання кольору в друкованих матеріалах, а також на комп'ютерних дисплеях.

Робота Мерча широко відома як авторитетне дослідження в області психологічного та фізичного аспекту кольору. Мерч пропонує 3 групи рекомендацій:

- з точки зору фізіології;
- з точки зору сприйняття;
- з точки зору пізнавальності.

Маркус — відомий розробник інтерфейсів і консультант в області комп'ютерної індустрії — застосовує наступні принципи до використання кольору :

- організація;
- економія;
- виділення;
- взаємодія;
- символізм;
- комунікація.

10.2. Використання звуку та анімації

По питанню використання звуку в якості зворотного зв'язку завжди ведеться багато дискусій. Якщо все добре продумано, ненав'язливо, можна за бажанням його вмикати та вимикати, то це нормальню. Перш за все слід вияснити, коли варто застосовувати звукову інформацію замість візуальної. Потім слід визначити тип звукового представлення тієї чи іншої ситуації. Система повинна бути контролюваною: користувач повинен мати можливість регулювати гучність або зовсім вимикати звук.

Рекомендації по використанню звуку, як і кольору, в більшому ступені засновані на особливостях людського сприйняття та здатності до навчання, ніж на ПЗ.

Сучасні робочі середовища є відкритими, тобто користувач оточений іншими людьми, телефонами, працюючими комп'ютерами, тому звукова форма передачі інформації малоекективна, адже людина часто не в стані відрізняти, чий комп'ютер надає якісь звуки.

Більшість сучасного комп'ютерного ПЗ використовує звуковий зворотній зв'язок — короткі «біпи», які сигналізують про помилку чи невірно обраний варіант. Але навіть такі незначні звуки можуть набридати.

Анімації, як і звуку, однакову увагу приділяють і розробники, і користувачі. При переміщенні миші по різним областям екрану її вказівник змінює свою форму для відображення типу дії, яку ви можете виконати над даним об'єктом або в даній області. Існує багато форм, які може приймати вказівник миші (стрілка, переміщення, розмір, очікування, заборона). Іконки і вказівники з анімацією теж можуть бути корисними.

Під анімацією розуміється зміна в часі візуального представлення графічного елементу. Аналогічно звуку головна перевага анімації полягає саме в розважальності дії. Курсор з анімацією спрощує його пошук на екрані, особливо це важливо для невеликих комп'ютерних записників з малим розрішенням. Анімація може використовуватись для вдосконалення візуального зв'язку між комп'ютерами та користувачами. Анімація може застосовуватись для виділення важливих іконок, відображення стану певного об'єкту і навіть пояснення його поведінки.

10.3. Термінологія та міжнародне проектування. Ключові питання розробки

Розробники не завжди говорять однією мовою з користувачами. В цьому і криється причина того, чому вони рідко спілкуються одне з одним — у них просто мало спільног! Ось чому розробники повинні визначитись з термінологією і нести відповідальність за кожне слово, яке з'являється на екрані. Треба бути особливо уважними при організації інформації в

міжнародних інтерфейсах.

До ключових питань розробки можна віднести:

- керуючі елементи;
- рядок меню та панель інструментів;
- метод drag- and- Drop;
- компоновку і розробку вікна.

В очах користувача кожна деталь, кожен керуючий елемент повинні служити якісь меті. З точки зору розробника кожна частина даних може бути представлена декількома способами. Оптимальність керуючих елементів для збирання певних типів інформації, яка вводиться користувачем, залежить від типу даних, які потрібно зібрати, кількості доступних даних, виду представлення іншої інформації на екрані, а також способу взаємодії користувачів з керуючими елементами екрану. Одним з ключових факторів, який слід враховувати при виборі керуючих елементів, є масштабованість.

Рядки меню, панелі інструментів і кнопки — всі вони можуть використовуватись для представлення аналогічних дій. Рекомендації для ефективної розробки щодо меню та панелей інструментів:

- в усіх додатках використовувати іконки, які узгоджуються по представленню;
- забезпечувати узгоджену взаємодію кнопок;
- групувати кнопки, які виконують взаємопов'язані команди;
- розставляти кнопки в логічному порядку;
- розташовувати кнопки для забезпечення швидкої роботи;
- забезпечувати роботу за допомогою клавіатури;
- дозволити користувачам займатись конфігуруванням;
- не допускати загромадження дисплею;
- іконки повинні бути візуально різні;
- використовувати колір, але з обережністю;
- кнопки повинні мати візуальні підказки до їх активації.

Інтерфейси стають все більш об'єктно-орієнтованими, більш графічними і візуальними, тому перевага все частіше надається прямому маніпулюванню drag- and- drop. Проблема цього методу полягає в тому, що відсутня візуальна вказівка на те, що об'єкти можуть або повинні переміщені або скинуті на інші об'єкти.

Компоновка і розробка елементів вікна — мистецтво і наука одночасно. Колір, шрифт, розмір, тип керуючих елементів, їх розмір, орієнтування керуючих елементів, питання симетричності, виділення, багато інших факторів — все впливає на кінцевий вигляд навіть найпростішого вікна.

10.4 Додаткові рекомендації по розробці КІ

На основі проведених досліджень та накопиченого практичного досвіду спеціалісти зі зручності застосування розробили 10 найбільш загальних проблем ГКІ та ООКІ:

- двох змістовні меню та іконки;
- орієнтація на одну мову;
- обмеження по введенню та прямому маніпулюванню;
- обмеження по виділенню та вибору;
- неясний порядок кроків;
- управління інтерфейсом вимагає більше кроків, ніж власне розв'язок задач;
- складні зв'язки між і всередині додатків;
- неадекватний зворотній зв'язок та підтвердження;
- недостатньо раціональна та упереджуvalьна система допомоги;
- недосконалі повідомлення про помилки, довідкові елементи, керівництва і документація.

Основні рекомендації по розробці інтерфейсу і компоновці вікна:

- асиметрія = активний стан, симетрія = стан спокою;
- ніколи не треба гнатись за помірністю;
- слід уважно стежити за послідовністю та узгодженістю;
- слід пам'ятати, що узгоджені інтерфейси означають кращу впроваджуваність системи;
- слід чітко вирівнювати керуючі елементи;
- якщо треба розташовувати деякі елементи за межами вирівнювання, то слід робити це логічно;
- слід створювати екранні об'єкти за зразком об'єктів з реального фізичного світу;
- не слід забувати, що абсолютна симетричність ускладнює читання інформації з екрану;
- елементи, які не відповідають шаблону, виділяються з загальної маси;
- елементи однакового розміру та кольору сприймаються як принадлежні до однієї групи.

Поради по розробці інтерфейсів:

- зосередитись на змісті;
- очікування повинні бути реалістичними;
- навчитись розпізнавати, що працює, а що — ні;
- використання підходящих інструментів для роботи;
- пам'ятати про свої сильні та слабкі технічні сторони.

ТЕСТОВІ ПИТАННЯ

1. ...— це сукупність засобів, за допомогою яких користувач спілкується з різними пристроями (з комп'ютером або побутовою технікою) або іншим складним інструментарієм (системою)

- інтерфейс користувача
- технології введення- виведення
- елементи управління системою
- інтернет

2. Інтерфейс користувача комп'ютерного додатку включає:

- всі відповіді вірні
- засоби відображення інформації, відображувану інформацію, формати і коди
- командні режими, мову «користувач–інтерфейс»
- діалоги, взаємодію та транзакції між користувачем та комп'ютером, зворотній зв'язок з користувачем

3. Користувацький інтерфейс об'єднує в собі всі елементи і компоненти програми, а саме:

- набір задач користувача, які він розв'язує за допомогою системи, елементи управління системою, навігація між блоками системи
- процес проектування КІ
- задоволеність користувача
- процес конструювання КІ

4. Задоволеність користувача – це функція невеликої кількості факторів, а саме:

- функція від можливостей користувацького інтерфейсу, часу відзвиву, надійності, пристосованості до інсталяції, інформаційної підтримки, пристосованості до супровождження та інших факторів
- функція пристосованості до інсталяції та надійності
- функція пристосованості до супровождження та інших факторів
- функція інформаційної підтримки

5. Підхід “зсередини- назовні” спочатку розглядає:

- внутрішні властивості системи
- зовнішні властивості системи
- зовнішні і внутрішні властивості системи
- властивості продукту, видимі кінцевому користувачу

6. Підхід “зовні- всередину” спрямований на:

- КІ та властивості продукту видимі кінцевому користувачу
 - внутрішні властивості системи
 - надійність системи
 - поведінку системи
7. ... –це планований обсяг робіт по конструюванню продукту, особливо КІ
- ітерація
 - план
 - специфікація
 - розгортка

8. Підхід типу “великий вибух” представляє собою

- спробу розробити “все або нічого”
- спробу розробити все
- практичність розробки
- зниження ризику при розробці КІ

9. Найкращий підхід до розробки КІ – це...

- еволюційний ітеративний підхід “ззовні всередину”
- підхід типу “великий вибух”
- підхід “зсередини- назовні”
- підхід “піти на прорив”

10. Ключова частина процесу проектування і розробки КІ – це

- етап конструювання
- етап планування
- етап розгортки
- етап вимог

11. ... – план створення продукту сконцентрований на побудові КІ та забезпеченні практичності враховує календарні терміни для кожного з етапів процесу, визначає основні фактори ризику, об’єднує всі можливі методи, встановлює цілі та критерії по відношенню до КІ

- етап планування
- етап вимог
- етап проектування
- етап специфікації

12. –на цьому етапі виконуються наступні задачі: опис користувачів, постановка задач користувачів, оцінка поточного рівня практичності, аналіз можливостей КІ, аналіз тенденцій

- етап вимог
- етап прототипування

- етап специфікації
- етап конструювання

13. ... – етап представляє сукупність характеристик програми, які сприймаються користувачем (вхідні сигнали, взаємодія користувача, відзив системи на вхідні сигнали та взаємодію користувача

- етап проектування
- етап конструювання
- етап вимог
- етап планування

14. ... – матеріалізація проекту програмного продукту в документальній формі, яка описує і показує дії користувачів, а також вигляд та поведінку ПЗ в специфічних ситуаціях

- етап специфікації
- етап планування
- етап прогнозування
- етап конструювання

15. ... – написання коду та автономне тестування

- етап конструювання
- етап вимог
- етап проектування
- етап концептуального проектування

16. ... – загальні критерії досягнення цілей створення КІ повинні бути чітко визначені, зрозумілі й прийняті керівництвом та розробниками досягнення поставлених цілей може вимагати багатократних ітерацій

- ітеративний підхід
- етап розгортки
- етап оцінки
- етап специфікації

17. ... – створення прототипів та моделювання – ефективні засоби ранньої оцінки проекту прототип – це матеріалізація побудованого проекту з використанням його передбачуваної платформи реалізації, включаючи обладнання, ОС, мови і засоби реалізації

- етап прототипування
- етап тестування
- етап конструювання
- етап вимог

18. ... – після того, як продукт задовольняє вимогам та потребам користувачів, він розгортається для використання за призначенням з цього моменту починається ряд наступних дій – оцінка продукту за участю користувачів, які не залучались до розробки, пілотне тестування, виконання користувацьких задач, які не оцінювались або не були передбачені під час проектування та розробки

- етап розгортки
- етап проектування
- етап конструювання
- етап тестування

19. ... – представляє собою сукупність високорівневих описів, абстракцій та оглядової інформації, яка дає розробникам та кінцевим користувачам загальне уявлення про програмний продукт, його структуру та КІ

- етап концептуального проектування
- етап прототипування
- етап тестування
- етап реалізації

20. Стиль взаємодії “користувач- комп’ютер”, в якому застосовуються 4 елемента: вікна, піктограми, меню та покажчики називають:

- графічний користувацький інтерфейс (GUI)
- користувацький Web- інтерфейс(WUI)
- користувацький інтерфейс кишенькових пристройів(HUI)
- об’єктно- орієнтовані КІ

21. Користувацький інтерфейс, для введення даних в якому застосовують “жестикуляційний” стиль з пером та сенсорним екраном називають:

- об’єктно- орієнтовані КІ
- графічний користувацький інтерфейс (GUI)
- користувацький Web- інтерфейс(WUI)
- користувацький інтерфейс кишенькових пристройів(HUI)

22. Користувацький інтерфейс, в якому необхідна навігація виконується в рамках одного або декількох додатків з використанням текстових або візуальних гіперпосилань називають:

- об’єктно- орієнтовані КІ
- графічний користувацький інтерфейс (GUI)
- користувацький Web- інтерфейс(WUI)
- користувацький інтерфейс кишенькових пристройів(HUI)

23. Об'єктно-орієнтований прикладний КІ повинен мати наступні властивості:

- забезпечувати безпосереднє маніпулювання
- забезпечувати безпосереднє введення даних
- забезпечувати контекстну незалежність від об'єктів
- введення великих обсягів інформації

24. Область пристрою відображення, використовувана для представлення взаємодії з об'єктами, інформації про об'єкти або для виконання дій над об'єктом називають:

- піктограма
- меню
- вікно
- покажчик

25. Список варіантів вибору гіперпосилань, які забезпечують доступ до інформації називають:

- навігаційна панель
- меню
- вікна
- покажчики

26. Варіант вибору, який відображає наступну сторінку інформації або пересуває фокус відображення на іншу область тієї ж сторінки називають:

- навігаційна панель
- меню
- гіперпосилання
- покажчик

27. Можливості прикладного рівня КІ програмного забезпечення включають наступні аспекти:

- введення великих обсягів інформації
- концептуальне проектування
- наочне представлення та стиль поведінки об'єктів в межах клієнтської області
- збільшення часу відзвіду при завантаженні та відкритті файлів

28. Принцип хорошого КІ при якому, програмні об'єкти забезпечують підвищення продуктивності і нарощування можливостей реальної системи без необхідного внесення складності називається :

- простота
- естетичність
- продуктивність
- пристосуваність

29. Принцип хорошого КІ при якому, програмні об'єкти повинні мати естетичну та ергономічну привабливість, при чому широко використовується графічний дизайн та візуалізація називається :

- естетичність
- пристосуваність
- продуктивність
- простота

30. Ключові положення, які впливають на принцип естетичності:

- додаток повинен в меншій мірі походити на комп'ютерні артефакти, а в більшій мірі – на користувальські об'єкти
- додаток повинен бути виродженим
- слід прагнути до максимальної візуалізації інформації
- додаток повинен відповісти задачі

31. Принцип хорошого КІ при якому, використання програмних продуктів вимагає мінімальних зусиль для виконання задач кінцевого користувача називається :

- естетичність
- пристосуваність
- продуктивність
- простота

32. Принцип хорошого КІ при якому, програмні об'єкти доступні в різних формах для задоволення індивідуальних потреб називається :

- естетичність
- простота
- пристосуваність
- продуктивність

33. Принцип хорошого КІ при якому, програмні об'єкти доступні в різних формах для задоволення індивідуальних потреб називається :

- естетичність
- простота
- пристосуваність
- продуктивність

34. Модель КІ, в якій наявне лише внутрішнє відображення того, як користувач розуміє і взаємодіє з системою називається :

- модель проектувальника
- модель програміста
- користувальська

- ментальна, або концептуальна

35. Модель КІ, в якій єдиний спосіб визначення вигляду користувацької моделі – поговорити з користувачем та подивитись, як він працює називається :

- модель проектувальника
- модель програміста
- користувацька
- ментальна, або концептуальна

36. Модель КІ, яка є функціональною специфікацією програмного продукту і може бути формально і недвозначно описана називається :

- модель проектувальника
- модель програміста
- користувацька
- ментальна, або концептуальна

37. Модель КІ, яка описує об'єкти, з якими працює користувач, і техніку маніпулювання ними називається :

- модель проектувальника
- модель програміста
- користувацька
- ментальна, або концептуальна

38. Під методом сумісної розробки КІ розуміють:

- паралельне виконання КІ двома командами розробників
- послідовне виконання розробки програмного продукту
- розподіл обов'язків розробки між програмістами
- участь користувачів в розробці програмних продуктів

39. Вкажіть способи збору інформації про користувача:

- аналіз задач користувачів
- інтерв'ю з справжніми та потенційними користувачами
- вивчення резюме користувачів
- тести по придатності

40. Застосування ергономіки до ПЗ - це неперервний процес, який

- визначає пристосування роботи до людини
- розвивається і спрямований на оптимізацію ПЗ у відповідності до задач і процесів робочого середовища
- рівносильний навчанню і набуттю практики

41. Довгочасна пам'ять зберігає інформацію, яка

- належить до колишнього досвіду
- впливає на думку людини
- належить до теперішнього досвіду
- містить комірки з інформацією

42. Ергономіка —

- пристосування роботи до людини
- вивчає взаємозв'язки між людиною та його роботою
- вивчає взаємозв'язки між людьми

43. Взаємозв'язки між людиною та його роботою вивчаються науково:

- ергономіка
- агрономіка
- піалогія
- фізіологія

44. Виділіть принципи ергономіки

- пристосування людини до роботи
- пристосування роботи до людини
- пристосування людини до ПК
- пристосування ПК до задач

45. З точки зору ергономіки пристосування роботи до людини означає

- проектування засобів і задач, які відповідають можливостям людей
- пристосування людини до роботи
- взаємозв'язки між людиною та його роботою
- характеристики процесу людського пізнання

46. Найбільш виражені характеристики процесу людського пізнання, яким повинен приділяти увагу проекувальник ПЗ, включає наступні властивості:

- відчуття та сприйняття
- навчання та запам'ятовування
- увага та мова
- інтуїція

47. Сенсорні буфери...,

- тимчасово зберігають фіксовані образи
- постійно зберігають фіксовані образи
- мають декілька типів сенсорної пам'яті
- спрямовані на усунення невизначеності

48. Звукоудавана пам'ять

- зберігає звукові дані
- постійно зберігають фіксовані образи
- зберігає обмежений обсяг інформації
- зберігає часові відношення

49. Образ –

- представляє собою зоровий запис
- зберігає просторові відношення
- зберігає звукові дані
- постійно зберігають фіксовані образи

50. Звуковідбиття

- зберігає часові відношення
- постійно зберігають фіксовані образи
- зберігає обмежений обсяг інформації
- зберігає звукові дані

51. Короткочасна пам'ять –

- частина пам'яті людини, яка зберігає обмежений обсяг інформації, яка належить до поточного часу
- частина пам'яті людини, яка зберігає обмежений обсяг інформації, яка належить до минулого часу
- зберігає інформацію, яка належить до колишнього досвіду
- зберігає історію підходів до вирішення поточних проблем

52. Довгочасна пам'ять

- зберігає інформацію, яка належить до колишнього досвіду
- частина пам'яті людини, яка зберігає обмежений обсяг інформації, яка належить до поточного часу
- зберігає історію підходів до вирішення поточних проблем
- зберігає звукові дані

53. Проміжкова пам'ять...

- зберігає історію підходів до вирішення поточних проблем
- зберігає тимчасові результати
- відображає зміни в планах на майбутнє
- зберігає історію підходів про вирішенні проблеми

54. До найбільш очевидних соціальних факторів належить

- тиск на користувача з боку клієнтів
- рівень зарплати

- поведінка в колективі
- ставлення з боку керівників

55. Застосування ергономіки до ПЗ – це...

- неперервний процес, що розвивається і спрямований на оптимізацію ПЗ у відповідності до задач і процесів робочого середовища
- процес спрямований на оптимізацію КІ у відповідності до задач і процесів робочого середовища
- процес удосконалення КІ
- процес проектування ефективного ПЗ з інтуїтивно зрозумілим КІ

56. Керуючі принципи (з точки зору ергономіки)...

- використовуйте ергономічні стилі в апаратному забезпеченні та КІ
- пам'ятай про користувачів
- постійно навчайтесь
- використовуйте шаблони

57. Керуючі принципи (з точки зору ергономіки)...

- не вимагайте надто багато або надто мало від користувачів
- використовуйте звичне коло знань
- підключайте до роботи колег
- знаходьте аналогічні рішення задач

58. Керуючі принципи (з точки зору ергономіки)...

- заохочуйте навчання
- не перевантажуйте короткочасну пам'ять
- використовуйте довгочасну пам'ять
- діліться своїми результатами

59. Золоте правило проектувальника -

- ніколи не робили іншим того, що вони зробили тобі
- ніколи нічого не робити вручну
- автоматизувати всі процеси
- ніколи нічого не робити

60. Принципи розробки користувацького інтерфейса

- контроль користувачем інтерфейса
- зменшення завантаження пам'яті користувача
- ітеративність користувацького інтерфейса
- створення дружнього інтерфейсу

61. Які з цих принципів входять у список принципів проектування (за Хансеном):

- знати користувача
- скоротити запам'ятовування
- здійснювати контроль над системою
- полегшення в користуванні

62. Які з цих принципів входять у список принципів проектування (за Хансеном):

- оптимізувати операції
- позбутися помилок
- знизити завантаження пам'яті
- збільшити швидкість проектування

63. Основне правило проектування

- дати користувачу контроль над системою
- знизити завантаження пам'яті
- збільшити швидкість проектування
- покращити графічний інтерфейс

64. Принципи за якими користувачу дається контроль над системою

- безрежимність
- гнучкість
- можливість переривання
- запам'ятовуваність

65. Принципи за якими користувачу дається контроль над системою

- корисність
- поблажливість
- ітеративність
- навчання

66. Принципи за якими користувачу дається контроль над системою

- доступність
- полегшення в користуванні
- запам'ятовуваність
- ітеративність

67. Принципи за якими користувачу дається контроль над системою

- пристосуваність
- інтерактивність
- застосуваність

- візуалізація

68. Принципи, які дозволяють знизити завантаження пам'яті користувача

- запам'ятовування
- розпізнавання
- застосуваність
- візуалізація

69. Принципи, які дозволяють знизити завантаження пам'яті користувача

- інформування
- терпимість
- ітеративність
- навчання

70. Принципи, які дозволяють знизити завантаження пам'яті користувача

- швидкість
- терпимість
- ітеративність
- навчання

71. Сумісність –

- ключовий аспект для використання користувацького інтерфейсу
- інтерфейс, який заохочуватиме користувача досліджувати
- визначає автоматичні процедури, а також підтримує прийняття рішень та потік робіт
- дозволяє використовувати установки прийняті в системі за замовчуванням

72. Принципи створення сумісного інтерфейсу –

- послідовність
- досвід
- прогнозування
- аналіз

73. Задача проектувальників КІ –

- створити дружній інтерфейс, який заохочуватиме користувача досліджувати його складові і властивості без страху зробити щось неправильно
- визначити автоматичні процедури, а також підтримувати прийняття рішень та потік робіт
- заохочувати користувача досліджувати КІ
- бути зв'язковою ланкою між розробником і замовником

74. Процес проектування та розробки програмних продуктів включає етапи:

- планування
- виконання ітерацій
- прогнозування

75. Найбільш важкий бік роботи над КІ пов'язаний з людськими факторами, такими як

- навчання
- досвід
- організаційна поведінка
- рівень заробітної плати

76. Процес проектування та розробки програмних продуктів не включає:

- опис предметної області
- планування
- збір вимог
- реалізацію

77. Найкращий спосіб приступити до роботи над складним, непід可控ним помилкам та таким, що відрізняється високою ймовірністю ризиків, проектом полягає в:

- описі технічного завдання
- зборі вимог
- складанні плану
- описі предметної області

78. Чи беруть безпосередню участь в процесі розробки продукту користувачі в ідеальному випадку:

- так;
- ні

79. Обсяг програмного коду пов'язаний з КІ:

- до 10%
- 10- 25%
- 30- 50%
- більше 60%

80. Операція drag- and- drop виконує:

- переміщення- вставку
- копіювання- вставку
- вирізання- вставку
- жоден із варіантів є неправильним

81. Розгляд обсягів робіт, витрат та ризиків, пов'язаних з розробкою продукту та складання докладного плану - це:

- збір вимог
- планування
- опис технічного завдання
- опис предметної області

82. Чи потрібно використовувати план для проекту:

- використовувати тільки для великих проектів
- ніколи не можна починати проект без плану
- план можна взагалі не використовувати
- використовувати тільки при потребі

83. Організація обробки даних, при якій дії повторюються багато разів, не приводячи при цьому до викликів самих себе:

- методологія
- трансформація
- ітерація
- ціклічність

84. Ітеративний орієнтований на користувачів план визначає принаймні основних ітерацій:

- одну
- дві
- три
- чотири

85. Технічний підхід до проекту визначається:

- описом технічного завдання
- технічним персоналом
- використовуваними апаратними засобами
- предметною областью проекту

86. Проект з великою тривалістю відрізняється значним ризиком то його хід важче прогнозувати, доцільно оформити ____ - денні відрізки робіт та більш деталізовані проміжні етапи:

- 7
- 30
- 60
- 90

87. Тривалість концептуального проектування для великих проектів та складного ПЗ:

- тиждень
- 2 тижні
- 4 тижні
- 8 тижнів

88. Тривалість почтакової ітерації з використанням прототипу для великих проектів та складного ПЗ:

- тиждень
- 2 тижні
- 3- 4 тижні
- 6- 8 тижнів

89. Тривалість проектування, прототипування та підготовка до тестування для великих проектів та складного ПЗ:

- тиждень
- 3 тижні
- 6 тижнів
- 8 тижнів

90. Тривалість тестування, аналізу та планування наступної ітерації для великих проектів та складного ПЗ:

- тиждень
- 2 тижні
- 4 тижні
- 8 тижнів

91. Тривалість високорівневого проектування/прототипування/оцінки/специфікації для великих проектів та складного ПЗ:

- 2 тижні
- 3 тижні
- 1- 2 місяці
- 2- 3 місяці

92. Тривалість низькорівневого проектування/прототипування/оцінки/специфікації для великих проектів та складного ПЗ:

- 2 тижні
- 3 тижні
- 1- 2 місяці
- 2- 3 місяці

93. Тривалість кожного користувачького елемента управління для великих проектів та складного ПЗ:

- 2 тижні
- 3 тижні
- 1- 2 місяці
- 2- 3 місяці

94. Тривалість тестування, аналізу та планування наступної ітерації для великих проектів та складного ПЗ:

- тиждень
- 2 тижні
- 4 тижні
- 8 тижнів

95. Тривалість високорівневого проектування/прототипування/оцінки/специфікації для великих проектів та складного ПЗ:

- 2 тижні
- 3 тижні
- 1- 2 місяці
- 2- 3 місяці

96. Тривалість низькорівневого проектування/прототипування/оцінки/специфікації для великих проектів та складного ПЗ:

- 2 тижні
- 3 тижні
- 1- 2 місяці
- 2- 3 місяці

97. Тривалість кожного користувачького елемента управління для великих проектів та складного ПЗ:

- 2 тижні
- 3 тижні
- 1- 2 місяці
- 2- 3 місяці

98. Частина загальної концепції проектування, яка включає принципи розробки, методологію розробки, тестування на зручність застосування, а також розуміння потреб користувачів - це

- стандарти та керівні принципи
- вимоги
- план
- проектування

99. Головною умовою міжнародного успіху програмних продуктів є

- націоналізація
- інтернаціоналізація
- космополітизм
- монополізація

100. Проблеми, які необхідно розв'язати при проектуванні міжнародних інтерфейсів:

- абревіатури
- іконки
- колір заливки
- форми предметів

101. Керівні принципи побудови інтерфейса розраховані на сьогоднішні системи

- друку інформації
- сканування інформації
- розпізнавання інформації
- введення та виведення інформації

102. Одна з проблем розробки інструкцій, які відповідають новим технологіям

- це
- розшифрування способів взаємодії користувача з системою
- форматування тексту
- узгодження мов
- узгодження форматів і мов

103. Розкриває сутність кожного об'єкту та елементу інтерфейсу в термінах та способах представлення на екрані, їх поведінку, механізм взаємодії з ними користувачів

- повний набір вимог
- простий набір вимог
- повний набір керівництв
- простий набір керівництв

104. Підбірка індустріальних керівництв по проектуванню - була розроблена

- Беллом
- Беллкором
- Дейвом
- Дейтом

105. Найпростіші принципи по розробці ПЗ, сформульовані Берстом:

- для свого програмного середовища оберіть відповідні промислові керівні принципи
- створіть корпоративне керівництво за стилем оформлення вашого програмного інтерфейса
- оберіть стиль КІ
- оберіть стиль й принципи розробки КІ

106. Організаціями по розробці стандартів є:

- ANSI
- DIN
- ISO
- AVV

107. Стандартними метриками, які можуть бути корисними при фіксації практичності ПЗ:

- встановлення або оновлення
- вивчення
- ефективність
- продуктивність

108. В якості причин невдалої розробки програмних продуктів, які не мають відношення до вимог є:

- недолік ясних, визначених та вимірюваних вимог
- обмежений контроль за вимогами під час розробки
- нестача ясних вимог по відношенню до практичності, КІ, узгодженості, комплектації, користувальських та ділових потреб та очікувань.
- не розуміння вимог розробниками

109. Основна проблема, пов'язана з розроблюваним продуктом (незалежно від етапа розробки), полягає в

- збиранні та управлінні введенням нових вимог в процесі розробки
- не розуміння вимог розробниками
- проблемами в колективі
- низькою обізнаністю

110. Орієнтований на користувачів колектив розробників формує наступні вимоги:

- аналіз задач та прецедентів;
- можливості (функції, користувальський інтерфейс і т.д.);
- компоненти задач по системі та користувачам
- щодо вивчення та поведінки користувача

111. Реконструкція інсуючих систем з наступною переробкою та реалізацією нових проектних рішень – це

- реїнжініринг ПЗ
- інженіринг ПЗ
- доповнення ПЗ
- переробка ПЗ

112. Якщо функція або характеристика КІ передбачає вимірювання, звітність, відстежування та служить основою для дій, то її слід визначати

- як головну вимогу
- як другорядну вимогу
- як основу для проектування
- як частину вимог до продукту

113. Частина загальної концепції проектування, яка включає принципи розробки, методологію розробки, тестування на зручність застосування, а також розуміння потреб користувачів - це

- стандарти та керівні принципи
- вимоги
- план
- проектування

114. Одна з головних причин створення невдалих продуктів полягає в:

- недостатній залученості користувачів до проекту
- високій залученості користувачів до проекту
- практичності використання
- детальній інформації про користувачів

115. Які є методи залучення користувачів в проект?

- метод групового інтерв'ю
- метод інтеграції
- метод простоти
- метод інтуїтивної наочності

116. До якого методу відноситься спостереження за реальними користувачами, які виконують реальну роботу?

- метод спостереження
- метод опитування
- метод простоти
- метод інтерв'ю

117. До якого методу відноситься опитування всіх учасників проекту, які виконують комплекс задач або входять в робоче середовище?

- метод опитування
- метод спостереження
- метод інтерв'ю
- метод простоти

118. До якого методу відносяться неформальні інтерв'ю з реальними користувачами?

- метод індивідуальних інтерв'ю
- метод опитування
- метод спостереження
- метод простоти

119. До якого методу відноситься формальні або неформальні інтерв'ю з колективом або групою реальних користувачів?

- метод групового інтерв'ю
- метод опитування
- метод спостереження
- метод індивідуальних інтерв'ю

120. До якого типу питань відноситься питання : «Яку операційну систему ви використовуєте?»

- питання про існуючі системи
- питання, що стосуються роботи
- питання про користувачів
- питання, що стосуються робочого місця

121. До якого типу питань відноситься питання: «Як часто зустрічається ця задача?»

- питання, які стосуються роботи
- питання про користувачів
- питання про існуючі системи
- питання ,що стосуються робочого місця

122. Архітектура КІ – це

- розподіл роботи між системою та користувачем, а також програмний інтерфейс між ними
- розподіл роботи між системою ПЗ
- розподіл роботи між стилями КІ
- загальний стиль КІ

123. Концептуальний проект включає наступні компоненти:

- загальний стиль КІ, прикладний стиль КІ, сутності та артефакти кінцевих користувачів, передбачувана користувацька модель
- загальний стиль КІ
- передбачувану користувацьку модель
- прикладний стиль

124. Найбільш важливий аспект концептуального проекту:

- його потенціал з точки зору формування і бачення
- альтернативи використання
- прогнозування альтернативних шляхів
- гнучка структура

125. Для одержання початкової інформації від користувачів застосовуються:

- методи залучення користувачів до участі в проекті
- методи оцінки практичності
- методи точності
- методи простоти

126. Для прогнозування перспективних шляхів розвитку проекту застосовуються:

- методи оцінки практичності
- методи залучення користувачів до участі в проекті
- методи альтернатив
- методи точності

127. Основна задача, пов'язана з концептуальним проектуванням, полягає в тому:

- щоб встановити, які функції покладаються на користувача, а які - на систему
- щоб встановити загальні функції
- щоб переглянути функції
- щоб редагувати функції

128. Сприйняття роботи системи користувачем називається...

- користувацькою моделлю
- альтернативною моделлю
- простою моделлю
- концептуальною моделлю

129. Користувацька модель включає наступні компоненти:

- поняття, можливості, аналогії і метафори
- системні та користувацькі об'єкти, зовнішній вигляд

- поведінка, порядок відображення та відношення, взаємодії
- всі відповіді вірні

130. Мета проектного колективу полягає в формуванні моделі, яка задовільняє декільком критеріям:

- орієнтація на користувача, природність, узагальнений погляд
- інтеграція, узгодженість
- простота
- всі відповіді вірні

131. Одна з головних причин створення невдалих продуктів полягає в:

- недостатній залученості користувачів до проекту
- високій залученості користувачів до проекту
- практичності використання
- детальній інформації про користувачів

132. Які є методи залучення користувачів в проект?

- метод спостереження, метод опитування, метод індивідуальних інтерв'ю, метод групового інтерв'ю
- метод інтеграції
- метод простоти
- метод інтуїтивної наочності

СПИСОК ЛІТРАТУРИ

1. Тео Мандел. Разработка пользовательского интерфейса: Пер. с англ. – М.: ДМК Пресс, 2001. – 416 с.
2. Р.Дж.Торрес. Практическое руководство по проектированию и разработке пользовательского интерфейса: пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 400 с.
3. Ганеев Р.М. проектирование интерфейса пользователя средствами Win32 API: Учебное пособие для ВУЗов. – М.: Горячая линия-Телеком, 2007. – 358 с.
4. Дадашов М. Проектирование пользовательского интерфейса на персональных компьютерах. – М.: DBS LTD, 1990. – 196 с.
5. Іванова Л.М., Авраменко О.А. Системне програмування та операційні системи. Створення GUI-додатків з використанням API WIN32: Методичні рекомендації – К.: НАУ, 2007. – 68 с.
6. Baecker, Ronald M., Jonathan Gurdin, William A.S. Buxton, Saul Greenberg. Readings in Human-Computer Interaction. - San Francisco, 2000. – 456 p.
7. Boling, Elizabeth. Usability testing for Web sites. – Bloomington, 1995. – 178 p.
8. Del Galdo, Elisa and Jacob Nielsen. International user interface. – New York, 1996. – 638 p.
9. Fernandes, Tony. Global Interface Design. – Chestnut Hill, 1995. – 871 p.
10. Horton, William, Lee Taylor, Arthur Ignacio, Nancy Hoft. The Web Page Design Cookbook: All the Ingredients You Need to Create – star Web pages. – New York, 1996. – 345 p.
11. Laurel, Brenda. Computers as Theater. – New York, 1991. – 289 p.

Навчально - методичне видання

Сучасні інструментальні засоби розробки користувачького інтерфейсу

Опорний конспект лекцій

Відповідальний за випуск:

*Дивак М. П., д. т. н., професор, декан факультету комп'ютерних інформаційних
технологій*

Підписано до друку 20.06.2012 р.

Формат 60x84/16. Папір офсетний.

Друк офсетний. Зам. № 4- 1579

Умов.- друк. арк. 6,43.

Тираж 100 прим.

Видавництво Тернопільського національного
економічного університету "Економічна думка"
46000, Тернопіль, вул. Львівська, 3, тел. 43- 22- 18
E-mail: edition@tane.edu.ua

*Свідоцтво про внесення суб'єкта видавникої справи
до державного реєстру видавців ДК № 3467 від 23.04.2009 р.*

Віддруковано ФО- П Шпак В. Б.

Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.

Свідоцтво платника податку: Серія Е № 897220

м. Тернопіль, вул. Просвіти, 6.

тел. 8 097 299 38 99

E-mail: tooums@ukr.net