A Human-Computer Interaction (HCI) é uma disciplina relacionada com o design, avaliação e implementação de sistemas informáticos interativos para uso humano e com o estudo dos principais fenômenos que os rodeiam. O que NÃO é HCI: Verdades universais sobre como não desenhar interfaces, fazer “coisas bonitas” e não é um conjunto de regras fixas e rígidas sobre construção de aplicações e interfaces. O que é HCI: compreender/conhecer o utilizador, otimizar desempenho e satisfação e avaliar qualidade dos sistemas e suas interfaces.

Usabilidade: De forma resumida é a “facilidade de utilização”. Capacidade de uma aplicação ser compreendida, aprendida, utilizada e atrativa para o utilizador, em condições específicas de utilização. Eficácia, eficiência e satisfação com que um produto permite a utilizadores específicos atingir objetivos específicos num contexto específico.

Eficácia – effective to use (effectiveness), eficiência – efficient to use (efficiency), prevenção de erros – safe to use (safety), utilidade – have good utility (utility), aprendizagem – easy to learn (learnability) e memorização – easy to remember how to use (memorability).

Norman refere os seguintes princípios a ter em conta para a usabilidade: Visibilidade, Retorno (feedback), Restrições, Mapeamento, Potencialidade (affordance) e Consistência

Desenho Centrado no Utilizador

Definição e Motivação: Abordagem ao desenho de sistemas interativos focada em fazer com que esses sistemas sejam usáveis. Objetivos: Melhorar a qualidade do trabalho, aumentar a produtividade, reduzir custos e melhorar a satisfação do utilizador. Circulo interativo: Analisar, Desenhar, Implementar, Avaliar.

Analisar: Identificar necessidades e estabelecer requisitos: Quem são os utilizadores? Que funcionalidades pretendem? Onde realizam as tarefas? Requisitos funcionais: Funcionalidades do sistema (exemplos: importar documentos no formato X, guardar informação sobre os campos Y e Z) e não funcionais: características particulares da utilização e da interação (exemplos: metáforas de utilização, organização semântica de determinados elementos).

Desenhar: Criar um modelo conceptual, estrutura geral da aplicação e da ideia, escolha de metáfora(s) e criação de esquemas, guiões.

Implementar: Criar protótipos que permitem identificar os problemas numa fase inicial.

Avaliar: Avaliar usabilidade, identificar problemas, e reagir adequadamente.

Analise de Utilizadores e Tarefas

Analisar: Objetivos: Descobrir as características dos potenciais utilizadores, perceber o modo como realizam as tarefas, identificar as tarefas que pretendem realizar e compreender o ambiente em que as tarefas são realizadas. Motivação: Identificar possíveis problemas o mais cedo possível e reduzir. Definição: Estudo de como as pessoas realizam as tarefas com os sistemas existentes, e do ambiente em que o fazem.

Utilizadores: Como se definem? Em que diferem? Dentro do grupo de utilizadores, em relação a outros grupos. Como utilizam ao longo do tempo e como são geridas as necessidades de perícia e a sua evolução? Perfis e Curvas de aprendizagem.

Tarefas: O que é que os utilizadores fazem? Que objetivos pretendem atingir? Que tarefas realizam agora para atingir esses objetivos? Que tarefas irá o novo sistema ajudar a realizar?

11 Perguntas que Sumarizam informação da Análise de Utilizadores e Tarefas: 1. Quem vai utilizar o sistema? Tipo de utilizador (principiante, perito) Faixa etária, Gostos e aversões particulares, Escolaridade e aptidões. 2. Que tarefas executam atualmente? 3. Que tarefas são desejáveis? (Futuro) 4. Como se aprendem as tarefas? Métodos comuns: manuais, perguntar, tentativa e erro, Pré-requisitos e formação 5. Onde são desempenhadas as tarefas? Caracterizar ambiente físico, social e cultural. 6. Qual a relação entre o utilizador e a informação? 7. Que outros instrumentos tem o utilizador? 8. Como comunicam os utilizadores entre si? Existência ou não de necessidade de comunicação entre utilizadores, Mecanismos de comunicação (telefone, e-mail, …) 9. Qual a frequência de desempenho das tarefas? 10.Quais as restrições de tempo impostas? Identificar durações aceitáveis para as tarefas, perceber o impacto das restrições temporais no tipo e quantidade de erros 11. Que acontece se algo correr mal? Ter em conta os tipos de reação dos utilizadores, identificar estratégias de contingência

Erros comuns: Os utilizadores foram descritos como deveriam ser ou como desejávamos que fossem, e não como realmente são. Os procedimentos da tarefa original têm problemas que foram replicados em vez de serem corrigidos. A análise foi incompleta e não captou aspetos fundamentais.

Seleção de tarefas

Reais e representativas: contemplar as diversas funcionalidades e incluir combinações mais comuns. Orientadas ao objetivo e não ao método: levantar questão para potenciar o desenho de alternativas e não restringir. Específicas: criar um cenário real incluindo detalhes. Misturar complexidades: Exemplo (aplicação de videoclube): ver novidades de filmes. Procurar filmes de ação em que Nicolas Cage faz papel de vilão.

Caracterização de tarefas: Objetivo: A meta que se pretende atingir Ex: Partilhar fotografia com pessoa da lista de contactos. Pré-condições: Tarefas prévias sem as quais é impossível atingir o objetivo. Ex: Tirar fotografia. Sub-tarefas: Passos individuais que decompõem a tarefa principal

Tipos de planos: Sequências fixas (Fazer x, y e z). Tarefas condicionais (Quando c, fazer x). Esperar eventos (Quando e, fazer y). Ciclos (Repetir x e y enquanto c). Tarefas em paralelo (Fazer x e y simultaneamente). Tarefas opcionais (Se necessário, fazer x, y e z). Combinações.

Requisitos de Usabilidade: Descrição dos objetivos pretendidos. Descrição da totalidade do contexto: utilizador, tarefa e ambiente. Quantificar objetivamente os valores desejados de eficácia, eficiência e satisfação.

Eficácia: Qualidade com que o utilizador atinge os objetivos. Exemplo: 90% dos utilizadores conseguirão enviar uma fotografia a um dos seus contactos. Eficiência: Qualidade da utilização de recursos para atingir o objetivo. Exemplo: em média, após tirarem uma fotografia, os utilizadores conseguirão partilhá-la com um dos seus contactos em menos de 20 segundos. Satisfação: Qualidade do processo de utilização do sistema (sentida pelo utilizador). Exemplo: os utilizadores irão preferir utilizar esta aplicação em vez do Photo Smart Ultra XPTO Snap and Share V2.0.

Métodos de Análise: Entrevista: Método direto e rápido, tipicamente é preferível primeiro fazer observações, começar com questões gerais e ir aprofundando gradualmente. Observação direta: essencial para compreender o contexto das tarefas. Observação indireta: diários e registos de interação (logs), são pouco comuns. Entrevistas de grupo: obter visões consensuais e identificação de áreas de discordância e conflito. Esclarecimento de que “os utilizadores não são todos iguais”. Questionários: bom alcance num curto espaço de tempo. Útil para selecionar pessoas a serem incluídas noutros métodos de análise. Documentação existente: não requer tempo dos utilizadores. Pode não corresponder totalmente com a realidade. Análise de sistemas similares: identificação de funcionalidades e possível tipo de interação. Identificação rápida de problemas e soluções menos adequadas. Inquéritos contextuais: Entrevista, observação e discussão no local de trabalho (mistura de entrevista e observação direta ativa)

Fatores Humanos

Visão: O dispositivo de entrada, por excelência. Limitações: Daltonismo. Compensações mediante incerteza ou incompletude. Ilusões de ótica.

Audição: Fonte de informação importante, por vezes subvalorizada.

Lei de Fitts: Modelo do movimento humano para previsão do tempo para atingir ou selecionar um alvo no ecrã.

T = a + b \* log2(1+ D/S) T: tempo do movimento a,b: constantes D: distancia S: tamanho (a,b = 0.1)

Aplicações práticas: Menus circulares (número reduzido de opções). Menus de contexto – clique com botão direito. Cinco alvos “ideais”(os 4 cantos do ecrã e o próprio pixel (posição atual)). Outros alvos adequados: bordos do ecrã.

Conceptualização da Interação

Dois tipos de design: Conceptual: descrever o comportamento do sistema. Físico: estrutura da interface, ou seja, concretizar o modelo conceptual

Modelos Mentais: As pessoas (os utilizadores) criam modelos mentais acerca do funcionamento dos objetos, da ocorrência de eventos e dos comportamentos de pessoas. E fazem-no a partir da familiaridade com sistemas/situações similares, da observação, da experiência e treino, de Instruções/Documentação e da utilização/Interação.

Modelo Conceptual: Um modelo conceitual é uma explicação, geralmente altamente simplificada de como algo funciona. Não precisa ser completo nem preciso, desde que seja útil.

Componentes: Metáforas e analogias. Conceitos: objetos, atributos e operações. Relações entre conceitos. Mapeamento

Características adicionais importantes: ser tão simples quanto possível, mas com funcionalidade. Ser focado na tarefa.

Metáforas: Elemento central do design. Explicam as situações por comparação com algo familiar. Descrição que sugere semelhança Exemplo: Átomo representado como “sistema solar”.

3 passos (Erickson, 1990) para a escolha de Metáforas: Definição funcional: Perceber o que o sistema pode fazer e as condições em que certas funcionalidades estão disponíveis. Utilizar informação recolhida na AUT. Identificação dos problemas: Perceber que funcionalidades são novas para os utilizadores e particularmente diferentes, pois poderão ser mais críticas. Geração de metáforas: Metáforas utilizadas pelos utilizadores ao descreverem as tarefas (AUT) e/ou focar o domínio da tarefa.

5 perguntas para avaliar a escolha (Erickson, 1990): A metáfora fornece uma estrutura? Que partes da metáfora são relevantes para o problema? A metáfora é fácil de representar? Irão os utilizadores perceber a metáfora? A metáfora é extensível?

Benefícios da criação: ao criar uma lista de conceitos, estamos a definir o léxico a utilizar, tanto na interface como na documentação. Vai permitir a criação de cenários de atividade. Dá uma ideia mais clara dos elementos e da estrutura da interface. Serve como primeiro passo no apoio à implementação, pois os objetos e ações poderão corresponder às classes e métodos a implementar. Define uma referência para a criação da documentação.

Protótipos: Um protótipo é uma representação concreta, mas parcial, do sistema que pretendemos desenvolver e que permite aos utilizadores interagirem com este e explorarem a sua adequação. Vantagens: Redução de custos. Testar alternativas. Facilidade e rapidez de alteração. Desenhar centrado no utilizado.

Protótipos em Papel: Fácil e rápido de criar/alterar. Não requer competências especiais. Concentração da atenção nas questões importantes. Testes: Simular a utilização da aplicação por utilizadores. Papéis a desempenhar. Resultados: Compressão de metáforas e modelo conceptual. Adequação das funcionalidades. Compreensão da navegação e fluxo de realização das tarefas. Adequação da terminologia utilizada. Completude da informação disponibilizada nos ecrãs.

Dispositivos e Interfaces

Introdução de texto: Teclado QWERTY/Alfabético (Tentativa de tirar partido do conhecimento natural que temos da ordenação alfabética. Lento, tanto para utilizadores experientes como principiantes) /DVORAK (Desenhado para otimizar a velocidade de. Predominância da mão direita). Teclado de acordes (Rápidos de aprender. Compactos. Fadiga na utilização).

Introdução de coordenadas: Rato (Mapeamento indireto). Touchpad (Mapeamento indireto). Trackball (Movimentos difíceis. Operabilidade só com um dedo). Joystick (Populares no mundo dos jogos. Semelhança aos controlos de aviões). Mesas digitalizadoras (Coordenadas absolutas. Populares para ilustração). Ecrãs tácteis e multi-toque (Mapeamento direto. Conjugação da entrada em saída num só dispositivo. Baixa precisão. Fadiga).

Menus: Menu Bar, Pull-Down, Pop-up.

Profundidade vs. Largura: Tipicamente, deve dar-se preferência a largura em relação a profundidade. Profundidade máxima recomendada: 3 ou 4 níveis. Largura máxima recomendada: 20 opções (Recapitular Lei de Hick, tempo que uma pessoa leva a tomar uma decisão com base no nº de opções).

Linguagens de comando: Flexibilidade e controlo pelo utilizador. Eficientes para utilizadores experientes. Difíceis de utilizar por utilizadores não avançados. Difíceis para aprender e relembrar.

Princípios de Design de Norman

Coerência: Estética, repetição do estilo e aparência ao longo da interface. Funcional, semelhanças do significado dos elementos (tipos de ícones, textos…). Interna, repetição de significados na aplicação. Externa, significados idênticos a outras aplicações (principalmente aplicações complementares).

8 Regras de ouro de Shneiderman: Manter a coerência. Oferecer usabilidade universal. Fornecer retorno informativo. Desenhar diálogos que indiquem o fecho de sequências. Evitar erros. Permitir a reversão de ações. Fornecer controlo e iniciativa ao utilizador.Reduzir a carga da memória de curta duração.

Heurísticas de Nielsen: H2.1 – Tornar o estado do sistema visível H2.2 – Correspondência entre o sistema e o mundo real H2.3 – Utilizador controla e exerce livre-arbítrio H2.4 – Coerência e adesão a normas H2.5 – Evitar erros H2.6 – Reconhecimento em vez de lembrança H2.7 – Flexibilidade e eficiência H2.8 – Desenho estético e minimalista H2.9 – Ajudar o utilizador a reconhecer, diagnosticar e recuperar de erros H2.10 – Dar ajuda e documentação

Desenho de Ecrãs

Alinhamento de texto: Cultura ocidental - Alinhamento principal à esquerda. Texto justificado: alternativa válida, mas deve-se ter em atenção a criação de descontinuidades. Alinhamento de números: Alinhamento preferencial à direita. Facilita cálculos mentais o Facilita interpretações (menores, maiores, …). Não colocar zeros à esquerda. Em números reais, alinhar na vírgula. Posicionamento: Centro do ecrã. Proporção áurea. Regra dos terços

Avaliação

Analíticos: Não envolve utilizadores: avaliação pericial (simulação do papel dos utilizadores) ou preditiva (análise das operações físicas e mentais). Empíricos: envolve utilizadores. Medição do desempenho e da satisfação dos utilizadores. Formativos: realizada durante o ciclo iterativo de desenvolvimento centrado no utilizador. Identificação de aspetos da interface que apresentem problemas de usabilidade. Sumativos: avaliação ao produto final. Medição do sucesso do produto acabado (em comparação com os objetivos estabelecidos no processo de design).

Avaliação Preditiva: Método analítico. Permite avaliar muito cedo a interface, antes desta existir (protótipo em papel é suficiente). Utilidade principal na comparação de soluções alternativas (mais do que na extração dos valores expectáveis propriamente ditos)

Vantagens: Não é necessário construir o sistema. Útil para comparações em soluções alternativas. Mais barato que testar com utilizadores ou peritos

Desvantagens: só aplicável a tarefas previsíveis. Previsões são simplesmente aproximações.

Goals: Definir os objetivos da interação. Podem ser divididos em sub-objetivos. Operators: Ações básicas para atingir o objetivo. Methods: Sequência de operadores para atingir um objetivo. Podem existir diversos métodos. Select: Regras para selecionar o método a utilizar.

KLM: Avaliação preditiva. Modelo físico para previsão do tempo necessário para um utilizador realizar uma determinada tarefa.

Operadores: D: Desenhar R: Resposta do sistema K: Premir uma tecla B: Carregar nos botões do rato P: Apontar (Lei de Fitts) H: Alternar entre teclado e rato M: Preparação mental

Avaliação com Utilizadores: Participantes o Considerações éticas (duração, conforto, informação, privacidade e controlo). Representativos (novamente: concretizado, mas diversificado). Distribuição (testes intergrupo vs. Intragrupo). Para tratamentos estatísticos válidos são necessários 30 utilizadores (com 20 poderá ser possível).