

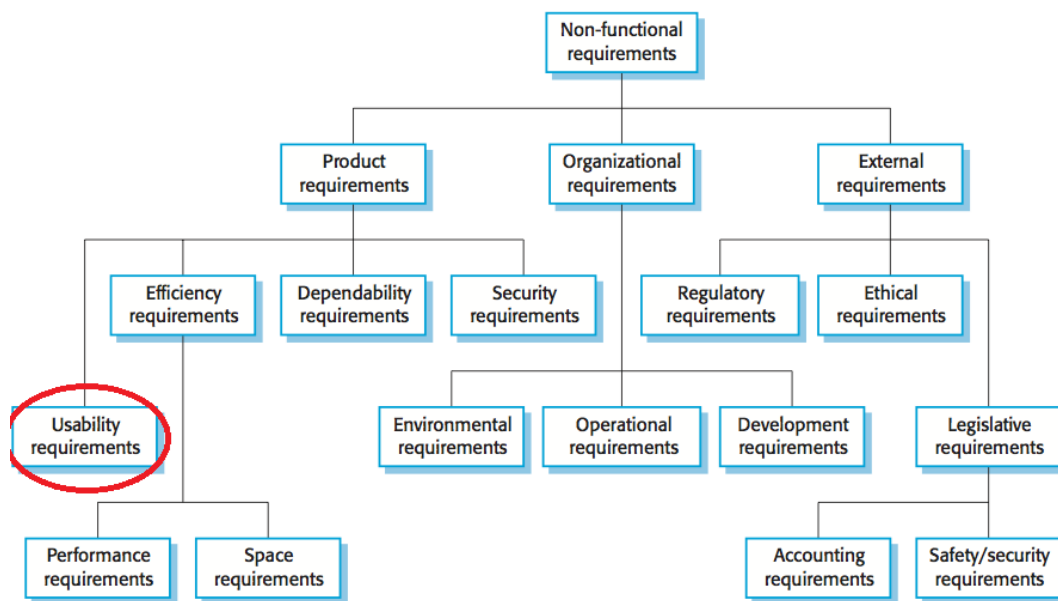
• Princípios de usabilidade e paradigmas de interação

○ Design de sistemas interativos

- Utilizador - bastante complexo, desconhecido, não controlável
- Interface de usuário (UI) – é o meio pelo qual o usuário e o sistema de computador interagem.
- O design da interface do usuário envolve um esforço considerável, pois os utilizadores não são todos iguais.
- Envolve o conhecimento:
 - Princípios de usabilidade: independentes da tecnologia
 - Paradigmas de usabilidade: dependentes da tecnologia

○ Usabilidade

- Diretamente relacionado com a capacidade que o sistema tem para permitir que os utilizadores atinjam os seus objetivos através da utilização do mesmo.
- Definido num contexto de uso: é uma propriedade do sistema de permitir que usuários específicos executem tarefas específicas de forma eficiente com eficácia e satisfação
- **Três aspetos fundamentais:**
 - Aprendizagem e Memorização: facilidade na aprendizagem do sistema e a relembrá-lo.
 - Eficiência e Eficácia: facilidade na utilização do sistema (rápido e com poucos erros).
 - Satisfação
- **Vantagens da usabilidade:**
 - Maior desempenho e satisfação do utilizador;
 - Custos de desenvolvimento mais baixos;
 - Menores custos de manutenção/suporte;



▪ Padrões de usabilidade

- ISO 9241-11 (1998) - Requisitos ergonômicos para trabalho em escritório com terminais de exibição visual
 - Explica como identificar as informações necessárias para especificar ou avaliar a usabilidade em termos de medidas de desempenho e satisfação.
- ISO 13407 (1999) - Processos de design para sistemas interativos centrados no ser humano
 - Quatro princípios de design: Envolvimento ativo dos usuários; Alocação adequada de função ao sistema e ao utilizador; Iteração de soluções de design; Design multidisciplinar

- Quatro atividades de design: Entender e especificar o contexto de uso; Especificar requisitos de utilizador e organização; Produzir mais de uma possível solução; Avaliar projetos contra requisitos;

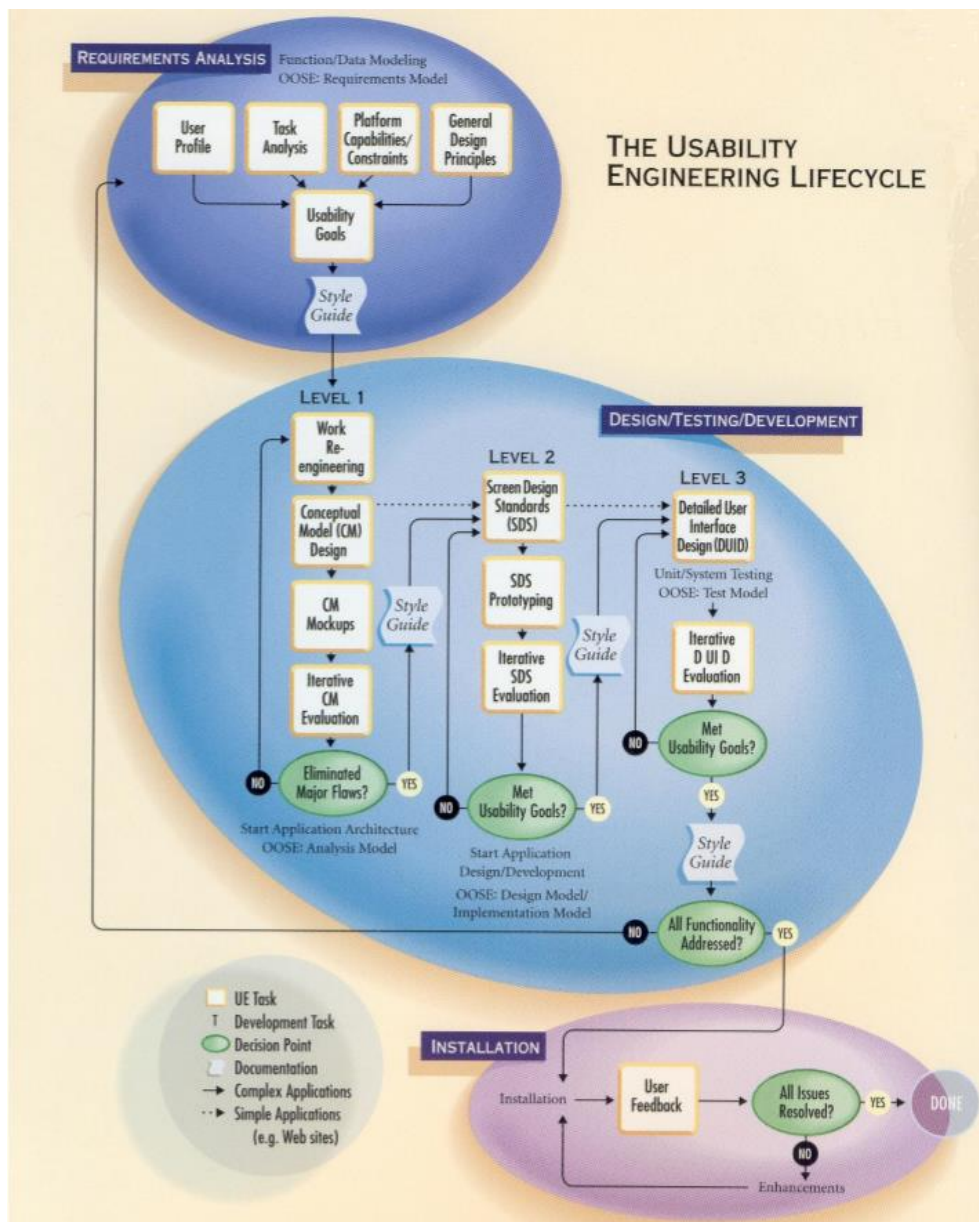
- **Princípios de usabilidade**

- Compatibilidade do utilizador; Compatibilidade das tarefas; Compatibilidade de fluxo de trabalho; Compatibilidade do produto; Feedback; Coerência; Familiaridade; Simplicidade; Flexibilidade; Controle; Invisibilidade da tecnologia; Robustez; Proteção contra erros

- **Experiência do utilizador**

- Envolve os comportamentos, atitudes e emoções sobre o uso de um determinado produto, sistema ou serviço do utilizador.
 - Inclui os aspetos práticos, experienciais, afetivos, significativos e valiosos da interação homem-computador, bem como o domínio sobre o produto.
 - Pode ser considerado subjetivo
 - É dinâmico, pois é constantemente modificado ao longo do tempo.

- **Introdução ao ciclo de vida do S/W iterativo**



Análise de requisitos	
Perfil do utilizador	Estabelecer as características importantes do utilizador para o design da UI
Análise contextual das tarefas	Obter um modelo de trabalho centrado no utilizador Extraír os requisitos de usabilidade do produto
Configuração das metas de usabilidade	Estabelecer metas de usabilidade quantitativas e qualitativas específicas para gerar o design da UI
Capacidades e restrições da plataforma	Estabelecer capacidades e restrições da plataforma tecnológica que limita as alternativas de design da UI
Princípios gerais de design	Identificar princípios e diretrizes que possam ser relevantes para o produto em desenvolvimento

Design, Testing, Development – Nível 1	
Trabalho de reengenharia	Reengenharia do modelo atual de trabalho do utilizador a fim de perceber o potencial da automação e apoiar de forma mais eficaz os objetivos comerciais
Modelo conceptual de design	Estabelecer uma estrutura de projeto de UI de alto nível coerente e baseada em regras para definir o cenário de design em níveis mais baixos
Modelo conceptual Mock-ups	Apoiar a avaliação, aperfeiçoamento e validação do Modelo Conceptual de Design
Avaliação do modelo conceitual iterativo	Avaliar, aperfeiçoamento e validação do Modelo Conceptual de Design

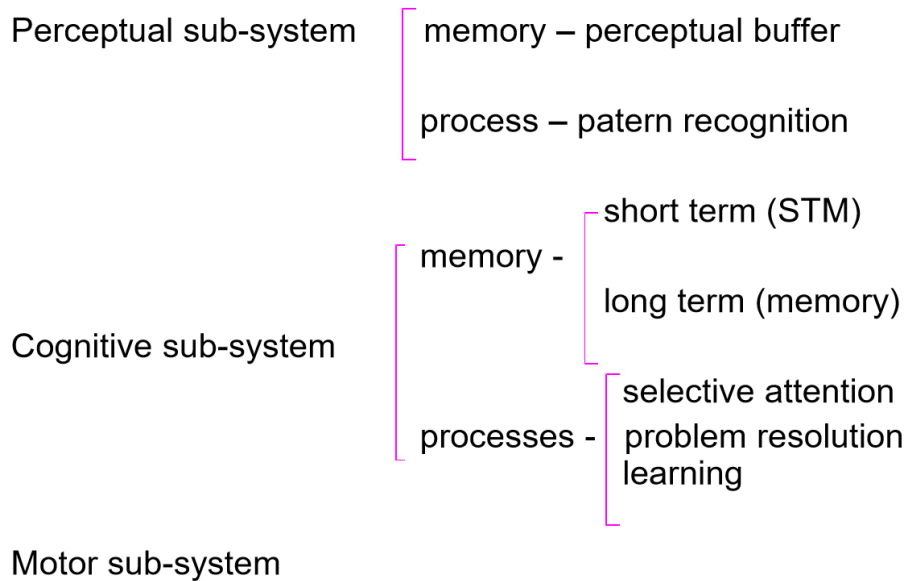
Design, Testing, Development – Nível 2	
Padrões de design de ecrã	Estabelecer um conjunto de padrões de design para preparar o cenário de design detalhado da UI
Padrões de design da prototipagem do ecrã	Apoiar a avaliação, refinamento e validação dos Padrões de Design do ecrã
Avaliação de padrões iterativos de design de ecrã	Avaliar, aperfeiçoamento e validação dos Padrões de Design do ecrã
Guia do desenvolvimento de estilo	Documentar o Modelo Conceitual de Design, os Padrões de Design do ecrã e o resultado da Análise de Requisitos

Design, Testing, Development – Nível 3	
Descrição detalhada do Design da UI	Design completo com detalhes da UI
Avaliação iterativa da UI	Avaliar, aperfeiçoamento e validação dos principais aspetos da UI detalhada

Implementação	
Feedback do utilizador	Obter os resultados dos testes de usabilidade, após este ser instalado e usado Informar o design da UI para possíveis atualização do produto

- **O utilizador**

- Sistema de processamento de informação Humano-SPIH (HIPS)



- **Patern Recognition** – utiliza fortemente o conhecimento anteriormente adquirido
- **Short Term Memory (STM)**
 - Curta duração: apenas alguns segundos
 - Capacidade limitada: 7 ± 2 elementos
- **Long Term Memory (LTM)**
 - “Infinita” capacidade e duração
 - Recuperação de informação lenta e não confiável

- **Vantagens e Desvantagens do HIPS**

- **Vantagens**
 - LTM capacidade infinita
 - LTM duração e complexidade
 - Capacidade de aprendizagem
 - Poderosa atenção seletiva
 - Processo de reconhecimento de padrões poderoso
- **Desvantagens**
 - STM capacidade limitada
 - STM duração limitada
 - Processo propenso a erros
 - Acesso não confiável no LTM
 - Processamento lento

- **Perfil do utilizador – características**

- **Experiencia e conhecimento**
 - Nível de educação e leitura
 - Experiência com o sistema e a tarefa
 - Língua materna
 - Literacia computacional

- **Trabalho e tarefas**
 - Frequência de utilização
 - Treino
 - Tipo de utilidade (opcional, obrigatória)
 - Utilização de outros sistemas
- **Características físicas**
 - Deficiências na distinção de cores (visão)
 - Deficiências físicas
 - Indiferenciação da mão dominante
 - Idade
- Modelos mentais e conceptuais
 - Para entender como desenhar um bom modelo conceitual, é necessário perceber modelos mentais
 - **Modelo mental**
 - Um modelo mental é a representação interna do conhecimento que o utilizador tem de um sistema.
 - Como criar um modelo mental? Usando o sistema, observando outros utilizadores, lendo documentação.
 - Neste modelo permite-nos: fazer previsões, determinar causas de eventos observados, determinar ações adequadas para produzir as alterações desejadas, compreender dispositivos semelhantes.
 - Este tipo de modelo é incompleto, instável, não científico e não tem limites específicos.
 - **Modelo conceptual**
 - Nível mais alto da interface do utilizador
 - Um modelo conceptual corresponde à forma como a funcionalidade da interface é apresentada.
 - Um modelo conceitual é a tentativa do designer de promover bons modelos mentais através de aspetos de IU
 - **Como criar um bom modelo conceptual**
 - Representar partes e processos invisíveis, de maneira visível (ex: copiar docs)
 - Dar feedback
 - Usar coerência (nomes de cores, sintaxes de comandos, estilos de diálogo, localização de informações na tela, etc., etc.)
 - Usar uma metáfora (opcional) - Explorar modelos mentais existentes do mundo real

• Modelos de Design

- Modelos de utilizador
 - **Modelos para obter os requisitos do utilizador**
 - Modelos socio-técnicos – reconhecer que o uso da tecnologia é feito com organização
 - USTM - User Skills and Task Match
 - OSTA – Open System Task Analysis
 - ETHICS – Effective Technical Human Implementation of Computer Systems
 - Metodologias “soft” do sistema– realçam a compreensão da situação em vez da obtenção da solução
 - Design participativo – abrange todo o ciclo de desenvolvimento da solução. Inclui os utilizadores como membros ativos da equipa de projeto e não apenas como participantes da avaliação.
 - **Modelos Cognitivos**
 - Representam a interação entre o utilizador e o sistema, ou seja, os aspetos de conhecimento, intenções e processamento do utilizador.

- Existem vários tipos de modelos cognitivos, entre os quais:
 - Hierarquia de objetos e tarefas (GOMS¹)
 - Modelos linguísticos
 - Físicos e de dispositivos (KLM²)

GOMS¹ – Goals., Operators, Methods and Selections (Moran e Newell, 1983)

- Este modelo consiste em decompor um objetivo de alto nível, numa sequência de tarefas (sub-objetivos)
- Goals (Objetivos) – o que o utilizador pretende alcançar
- Operators (Operadores) - operações básicas que o utilizador precisa de executar para usar o sistema (Pode afetar o sistema ou não, p.ex: pressionar uma tecla ou ler uma mensagem)
- Methods (Métodos) – possíveis decomposições do objetivo de alto nível em sub-objetivos
- Selections (Seleções) – regras para a seleção de possíveis métodos, tendo em atenção o tipo de utilizador e o estado do sistema
- **Capacidades:**
 - É utilizado na pesquisa do modelo cognitivo
 - Pode descrever adequadamente como é que os utilizadores experientes realizam as tarefas de rotinas do sistema
 - Quando associado a um modelo de dispositivos permite fazer estimativas de tempo
- **Limitações:**
 - Não fornece informações sobre o conhecimento do utilizador

KLM² – Keystroke-Level (Card, Moran e Newell, 1980)

- Este modelo prevê o desempenho do utilizador com base nas características motoras do sistema
- As tarefas têm duas fases: Aquisição (construção da representação mental da tarefa) e Execução (usando o sistema)
- O KLM permite apenas estimar o tempo que o utilizador despende na fase de execução da tarefa, recorrendo ao sistema
- A fase de Execução pode ser decomposta em 7 operadores:
 - K - Digitação (varia com a habilidade do utilizador)
 - B – Carregar no botão do rato
 - P – Acertar no alvo (Fitts' law)
 - H – Combinação entre o rato e o teclado
 - D – Desenhar usando o rato
 - M – Preparação mental ou ação física
 - R – Resposta do sistema

○ **Análise de Tarefas**

- A análise de tarefas é muito baseada na observação – ferramenta fundamental
- Pode ser utilizada para produzir documentação e material de ensino, em design de novos sistemas de alto nível.
- **Abordagens à análise de tarefas**
 - **Decomposição da Tarefa** – dividir a tarefa em sub-tarefas que devem ser realizadas numa sequência específica.
 - **Baseado no conhecimento** – considerar que os utilizadores precisam de saber tudo o que esteja relacionado com os objetos e ações envolvidas na realização da tarefa, bem como a organização do sistema
 - **Baseado na relação** – é focada nas relações entre atores e objetos, bem como nas ações que realizam.

	ANÁLISE DE TAREFAS	GOMMS
MODELOS	<ul style="list-style-type: none"> • Aspectos do mundo real que não fazem parte do sistema 	<ul style="list-style-type: none"> • Processos cognitivos enquanto as tarefas são efetuadas
PONTO DE VISTA	<ul style="list-style-type: none"> • Descreve a realização das tarefas feitas pelo utilizador num ponto de vista externo 	<ul style="list-style-type: none"> • Descreve a realização das tarefas feitas pelo utilizador num ponto de vista interno
S/W LIFECYCLE	<ul style="list-style-type: none"> • Usado em fases iniciais do S/W lifecycle 	<ul style="list-style-type: none"> • Usado na fase de avaliação do S/W lifecycle

○ Notações de Diálogo

- Várias notações são utilizadas para descrever diálogos entre o utilizador e o sistema.
- A utilização de uma notação para a especificação do diálogo de uma interface de utilizador torna mais fácil:
 - a separação entre os elementos da interface e a semântica da aplicação
 - a passagem de informação sobre a interface entre os diversos membros da equipa de projeto
- As notações textuais para a especificação do diálogo, em relação às notações em forma de diagrama são mais adequadas para uma análise formal
- AS diretivas existentes para o projeto dos diversos aspetos das interface de utilizador devem ser escolhidas com critério pois podem não ser aplicáveis ao nosso caso específico

● **Estilos de diálogo/interação**

○ Menus

- **Vantagens**
 - Autoexplicativo
 - Não carrega a memória (recordar em vez de relembrar)
 - Evita erros sintáticos
 - Melhora a visibilidade
- **Desvantagens**
 - Não é eficiente
 - Não é flexível
 - Não é prático quando temos muitas opções
- **Perfil dos utilizadores adequados**
 - Conhecimento e experiência
 - Baixa experiência do sistema
 - Uso frequente de outros sistemas
 - Baixa literacia computacional
 - Trabalho
 - Baixa frequência de utilização
 - Sem treino
 - Utilização opcional
 - Tarefas altamente estruturadas
- **Design de menus**
 - Aspetos relevantes: estrutura do menu; pedido das opções; seleção das opções; invocação do menu; navegação
 - Guidelines
 - Adequar a estrutura do menu à estrutura da tarefa
 - Minimizar a amplitude de profundidade do menu

- Usar um pedido de opções adequado
- Ser coerente (design, nomes das opções, etc)
- Dar feedback ao utilizador
- Incluir dicas de ferramentas caso os nomes ou os ícones não forem autoexplicativos
- Indicar as opções disponíveis
- Manipulação direta
 - As ações são realizadas diretamente nas representações visuais dos objetos (touchscreen)
 - **Distancia Semântica** - distancia subjetiva entre os objetivos do utilizador e os objetos e as ações oferecidas pelo sistema
 - **Distancia Articular** – distancia entre o significado das ações e o que realmente acontece quando estas são executadas
 - Caracterizado por:
 - Representação contínua dos objetos
 - Ações físicas em vez de comandos
 - Ações rápidas, incrementais e reversíveis com resultados visíveis
 - **Vantagens**
 - Fácil de aprender e relembrar
 - Direto
 - Ações flexíveis e facilmente reversíveis
 - Feedback visual imediato e de contexto
 - Menos propenso a erros
 - **Desvantagens**
 - Não é autoexplicativo
 - Pode ser ineficiente
 - Difícil de desenhar ícones reconhecíveis (especialmente para ações)
 - Ícones ocupam mais espaço na tela do que o texto
 - **Design de Manipulação direta**
 - Guidelines
 - Minimizar distância articulatória e semântica
 - Usar diretrizes gerais para projetar uma UI utilizável:
 - Coerência
 - Bom modelo conceptual
 - Feedback
 - Organização adequada de funcionalidades
 - Layout de tela adequado
 - Utilização adequada de cores
 - Manipulação de erros adequada

• Outros estilos de diálogo/interação

- Preenchimento de formulários
 - O preenchimento de formulário é particularmente útil em rotinas, trabalho de escritório ou para tarefas que exigem muita introdução de dados.
 - **Vantagens**
 - Autoexplicativo
 - Recordar em vez de relembrar
 - Permite entrada de dados distintas, contrariamente aos menus
 - Contextualiza e orienta o utilizador
 - **Desvantagens**
 - Implica conhecer quais os inputs válidos

- Propenso a erros
 - Não é muito flexível
 - Consume espaço na tela
 - **Perfil dos utilizadores adequados**
 - Conhecimento e experiencia
 - Alta/moderada habilidade de digitação
 - Alta/moderada experiência de tarefas
 - Alta/moderada experiência de aplicação
 - Características da tarefa
 - Frequência de utilização alta/moderada
 - Pouco treino
 - Tarefas altamente estruturadas
 - **Design de menus**
 - Aspetos relevantes: organização e layout, títulos e campos, formatos de inputs, instruções e ajuda, navegação, manipulação de erros
 - **Guidelines**
 - Evitar layouts não familiares
 - Alinhar os títulos
 - Fornecer um menu quando as possíveis entrada de informação são conhecidas
 - Indicar quais os campos que são obrigatórios
 - Os formatos de entrada de informação devem ser conhecidos e claros
 - AS instruções para o preenchimento dos campos devem ser claras
- Teclas funcionais
- **Hard Keys** – invocam sempre a mesma funcionalidade, tem abreviaturas ou default icons (teclas calculadora; menu start key...)
 - **Soft Keys** - invocam funcionalidades diferentes mediante o contexto de utilização, não tem abreviaturas ou default icons (F1...F12)
 - **Vantagens**
 - Autoexplicativo
 - Recordar em vez de relembrar
 - Fácil de utilizar
 - Flexível
 - Requer pouco ou quase nenhum espaço de tela
 - **Desvantagens**
 - Número limitado de teclas
 - Expansões de hardware são dispendiosas
 - **Perfil dos utilizadores adequados**
 - Conhecimento e experiencia
 - Alta/moderada experiência de tarefas
 - Alta/moderada experiência de aplicação
 - Características da tarefa
 - Frequência de utilização baixa/alta
 - Pouco treino ou sem treino
- Linguagens de comando
- As linguagens de comando também devem ser desenvolvidas de modo a serem o mais utilizável possível.
 - **Vantagens**
 - Poderoso
 - Flexível
 - Eficiente
 - Não requer muito espaço de ecrã
 - **Desvantagens**

- Difícil de aprender
- Não é autoexplicativo
- Propenso a erros
- As melhorias/alterações não são logo visíveis
- **Perfil dos utilizadores adequados**
 - Conhecimento e experiencia
 - Alta experiência de tarefas
 - Alta experiência de aplicação
 - Alta literacia computacional
 - Alta habilidade de digitação
 - Características da tarefa
 - Alta frequência de utilização
 - Treino formal
- **Guidelines**
 - Uso de uma sintaxe coerente
 - Uso de uma gramática de ação-objeto natural e fácil de lembrar
 - Permita os seguintes recursos de interação:
 - Padrões (default)
 - Edição de comandos
 - Interpretação inteligente
 - Auto complete
 - Feedback
 - Ajuda e documentação
 - Tornar o idioma “adaptável ao utilizador”

ESTILO DE INTERAÇÃO	VANTAGENS	DESVANTAGENS	EXEMPLOS
MANIPULAÇÃO DIRETA	<ul style="list-style-type: none"> • Interação rápida e intuitiva • Fácil de aprender 	<ul style="list-style-type: none"> • Difícil de implementar • Adequado apenas quando temos uma metáfora visual para tarefas e objetos 	<ul style="list-style-type: none"> • Videojogos • Sistemas CAD
MENUS	<ul style="list-style-type: none"> • Previne erros do utilizador • Pouca digitação 	<ul style="list-style-type: none"> • Experiência do utilizador lenta • Pode tornar-se complexo se os menus tiveram muitas opções 	<ul style="list-style-type: none"> • Maioria dos sistemas propostos
PREENCHIMENTO DE FORMULÁRIOS	<ul style="list-style-type: none"> • Fácil entrada de dados • Fácil de aprender • Verificável 	<ul style="list-style-type: none"> • Ocupa muito espaço do ecrã • Causa problemas quando o utilizador insere dados que não são aceites 	<ul style="list-style-type: none"> • Controlo de stocks • Informação pessoal
LINGUAGEM DE COMANDOS	<ul style="list-style-type: none"> • Poderosa • Flexível 	<ul style="list-style-type: none"> • Difícil de aprender • Gestão de erros deficiente 	<ul style="list-style-type: none"> • Sistemas operativos • Sistemas de controlo e comando
LINGUAGEM NATURAL	<ul style="list-style-type: none"> • Acessível a utilizador comuns • Facilmente extensível 	<ul style="list-style-type: none"> • Requer mais digitação • Não é muito confiável 	<ul style="list-style-type: none"> • Sistemas de recuperação de informações

• Dispositivos de informação no ecrã e utilização da cor

- Design de ecrã
 - O design do ecrã é uma parte importante do desenvolvimento da UI
 - Um design de ecrã inadequado pode degradar o desempenho do utilizador

- **Informações de layout - guidelines**
 - Deve incluir apenas e toda a informação necessária
 - Começar no canto superior esquerdo e alinhamento para a esquerda
 - Agrupar os itens de acordo com o seu tipo
 - Deixar bastante espaço em branco
 - Usar traços entre as várias colunas (índice)
- **Texto - guidelines**
 - Evitar usar frequentemente ou totalmente letras maiúsculas
 - Não usar demasiadas fontes para enfatizar
 - Em várias colunas usar traços ou riscas (tabelas com riscas)
 - Mensagens devem:
 - Ter um nível de detalhe adequado ao conhecimento e experiência do utilizador
 - Serem específicas e compreensíveis
 - Serem breves e concisas
 - Serem positivas
 - Serem úteis
- **Números - guidelines**
 - O uso de números inteiros deve ser bem justificado
 - Os números reais devem ser alinhados pelo ponto decimal
 - Evitar zeros desnecessários (à esquerda)
 - Os números longos devem ser divididos em grupos de 3 ou 4
- **Técnicas de codificação - guidelines**
 - Usar parcialmente: negrito, sublinhado, itálico, tamanho e fonte de estilo adequado, proximidade, fronteiras, som, cor...
- Utilização da cor
 - A cor de um objeto depende da
 - Características do material
 - Iluminação
 - Cor ambiente
 - Sistema visual humano
 - **Vantagens**
 - Mostrar organização da exibição da informação
 - Representar valores
 - Cativar a atenção
 - Facilitar a pesquisa em telas complexas
 - Despoletar emoções
 - Quando usadas em excesso ou inadequadamente podem degradar o desempenho do utilizador.
 - **Guidelines**
 - Usar a cor de forma parcimoniosa
 - Usar um número limitado de cores
 - O sistema deve funcionar, primeiramente sem cores
 - Usar cores de forma coerente
 - Evitar usar simultaneamente várias cores saturadas
 - Não transmitir informações apenas através da cor
 - Fazer com que a codificação seja o mais simples possível
 - Permitir que o utilizador controle o código de cores
 - Ter e, atenção o significado cultural das cores

- **Dispositivos de entrada**

- Keyboards

- Pointing devices – Mouse

- **Vantagens**

- Relação direta entre a mão e o movimento do cursor (distância, velocidade, direção)
 - Permite o controlo da velocidade
 - Permite movimentos contínuos em todas as direções

- **Desvantagens**

- Requer movimento da mão entre o rato e o teclado
 - Espaço adicional (footprint)
 - Coordenação mão-olho

- Pointing devices – Trackball

- **Vantagens**

- Relação direta entre a mão e o movimento do cursor (velocidade, direção)
 - Permite o controlo da velocidade
 - Permite movimentos contínuos em todas as direções
 - Não precisa de espaço adicional footprint)

- **Desvantagens**

- Requer movimento da mão entre o trackball e o teclado
 - Coordenação mão-olho

- Pointing devices – Touch Screens

- **Vantagens**

- Relação direta entre a mão e o movimento do cursor (distância, velocidade, direção)
 - Não precisa de espaço adicional footprint)
 - Permite movimentos contínuos em todas as direções

- **Desvantagens**

- O dedo pode ser demasiado grande para a precisão necessária
 - Torna-se cansativo após longo período de utilização
 - O dedo pode esconder informação importante
 - O ecrã pode ficar sujo

- Sistema de reconhecimento de voz (Bell Lab, 1972)

- **Vantagens**

- Facilita pessoas com deficiências físicas
 - Permite que o utilizador se movimente enquanto utiliza este sistema
 - Permite que o utilizador utilize este sistema quando tem os olhos ocupados
 - Permite que o utilizador utilize este sistema num ambiente de baixa visibilidade

- **Desvantagens**

- A voz é transitória
 - Não há feedback natural
 - Pode incomodar outras pessoas
 - Pode retirar privacidade aos utilizadores
 - Pode ser mais lento e mais cansativo

- **Dispositivos de saída (matéria inútil – cultura geral)**

- Graphics/visual Displays

- **Monitor**

- HMDs (VR/AR)
- Binoculars (...)
- Voice synthesizers

• Métodos de avaliação de IU

○ Métodos Analíticos (sem utilizadores)

▪ **Avaliação Heurística (Nielsen and Molich, 1990)**

- *Como realizar a avaliação heurística? Cada avaliador:*
 - Primeiro, faz uma análise geral para conhecer a IU
 - Faz uma análise sistemática tendo em conta as heurísticas
 - Anota os eventuais problemas de usabilidade, heurística que viola e o grau de severidade
 - Finalmente, compila todos os potenciais problemas
- *Dez Heurísticas de Nielsen*
 - Visibilidade do estado do sistema
 - Coerência entre o sistema e o mundo real
 - Controlo do utilizador e a liberdade
 - Consistência e padrões
 - Prevenção de erros
 - Reconhecer em oposição a relembrar
 - Flexibilidade e eficiência de uso
 - Design estético e minimalista
 - Ajudar os utilizadores a reconhecerem diagnosticar e recuperar de erros
 - Ajuda e documentação
- *Graus de gravidades de Nielsen*
 - Grau 0 - não há consenso quanto a problema de usabilidade
 - Grau 1 - problema cosmético
 - Grau 2 - problema menor
 - Grau 3 - problema de importante usabilidade - é necessário corrigir
 - Grau 4 - Catástrofe de usabilidade - é imperativo corrigir antes de continuar o desenvolvimento
- *Vantagens*
 - Deve produzir resultados uteis com um investimento moderado
 - Simples de aplicar mesmo quando os avaliadores não são experientes
- *Limitações*
 - Subjetivo (particularmente quando há muito avaliadores)
 - Não consegue selecionar todos os problemas de usabilidade
- **Apos uma avaliação heurística e sempre necessário uma avaliação com utilizadores**

▪ **Cognitive Walkthrough (Wharton, 1992)**

- *Como realizar o cognitive walkthrough:*
 - Análise de tarefas: sequência de passos ou ações solicitadas pelo utilizador a fim de concluir a tarefa e obter resposta do sistema.
 - Designers e developers trabalham como um grupo, interrogando-se num conjunto de perguntas a cada fase.
 - Recolha de dados feita durante o walkthrough: os problemas de usabilidade são detetados quando se interrogam sobre determinadas questões a cada sub tarefa.
 - Reportar potenciais problemas
 - Redesign da IU tendo em atenção os problemas identificados.
- *Cognitive walkthrough – 4 questões:*

- 1 - O utilizador vai conseguir alcançar o objetivo da sub-tarefa?
- 2 – O utilizador apercebe-se que a correta ação esta disponível?
- 3 - O utilizador vai conseguir perceber que realizando a correta ação consegue alcança o objetivo da sub-tarefa?
- 4 – O utilizador tem feedback?
- **Streamlined Cognitive Walkthrough (Spencer, 2000)**
 - *Apenas apresenta duas questões:*
 - 1 – O utilizador sabe o que deve saber naquela (sub)tarefa?
 - 2 – Caso o utilizador proceda corretamente, como é que ele sabe isso e se está na direção certa para alcançar o seu objetivo?
- *Limitações dos Métodos Analíticos*
 - São subjetivos
 - Envolvem vários especialistas em usabilidade
 - Não conseguem encontrar todos os problemas de usabilidade
- **Estas limitações implicam que haja, posteriormente, uma avaliação usando métodos empíricos!**
- Métodos Empíricos (com utilizadores) - implicam teste de usabilidade³
 - **Observação**
 - *Várias formas de observação:*
 - Direta – o observador tira notas
 - Indireta – através de vídeo/áudio, mais complexo e demoroso
 - Pensando em voz alta - os utilizadores vão explicando o que vão fazendo
 - Logging – a atividade dos utilizadores é registada pelo sistema
 - Combinação destas várias formas, etc
 - **Inquéritos**
 - Feitos através de questionários (abrangem mais pessoas; menos flexíveis) ou de entrevistas

Testes de Usabilidade³

- Envolvem observação e inquéritos
- Os principais aspetos dos testes são: Participantes; Tarefas; Testes à implementação e ao sistema; Protocolo; Medidas de usabilidade; Análise de dados
- Para os relatórios dos testes é usado como padrão: CIF (Common Industry Format)
 - CIF – é um modelo standard de relatório de avaliação sumativa. Uma avaliação sumativa produz métricas de usabilidade que descreve quão usável é um produto quando usado num determinado contexto. Contrariamente à avaliação formativa que serve para identificar problemas que podem ser resolvidos.
- **Experiências controladas**
 - Define uma hipótese
 - Define o input, output e variantes secundárias
 - Define o design experimental (dentro de grupos / entre grupos)
 - Seleciona os participantes
 - Prepara toda a documentação (lista de tarefas e correspondentes grau de dificuldade, questionário final, lista de tarefas para o observador tirar notas)