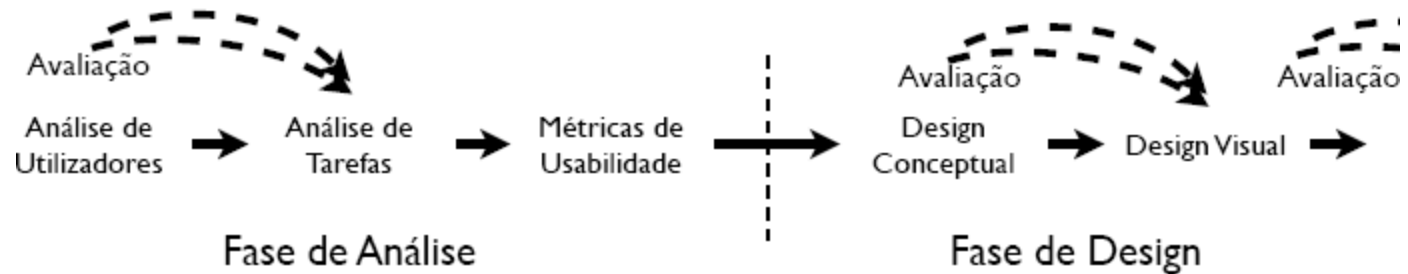


Avaliação de Usabilidade:



Objectivo é testar a usabilidade e funcionalidade do sistema

- Métodos Analíticos
 - Avaliação Heurística (avaliação por peritos)
 - Avaliação Preditiva (modelos): GOMS, KLM
- Métodos Empíricos (avaliação com utilizadores)
 - Requer um protótipo funcional

Avaliação Heurística:

Métodos para avaliar a IU de forma Rápida, Barata e Simples

> Jakob Nielsen, *Discount Usability Engineering*

- Rápida
 - Um dia ou menos para aplicar
- Barata
 - Não precisa de laboratórios ou equipamento
- Fácil de Aprender
 - Pode ensinar-se em duas horas ou menos
- Pequeno conjunto de avaliadores (≤ 5) examinam a IU
 - Verificam individualmente o cumprimento das heurísticas de usabilidade
 - Diferentes avaliadores encontrarão (provavelmente) problemas diferentes





Processo de Avaliação Heurística:

- Avaliadores “exercitam” a IU várias vezes
 - Inspeccionam vários elementos de diálogo
 - Comparam com lista de princípios de usabilidade
- Princípios de Usabilidade
 - Heurísticas de Nielsen
 - Lista suplementar de heurísticas específicas da categoria
- Usa-se as violações dos princípios para detectar e corrigir problemas



Heurísticas de Usabilidade

- 1 Tornar o estado do sistema visível
- 2 Falar a linguagem do utilizador
- 3 Utilizador controla e exerce livre-arbítrio
- 4 Consistência e adesão às normas
- 5 Evitar Erros
- 6 Reconhecer, em vez de lembrar
- 7 Flexibilidade e eficiência
- 8 Desenho de ecrã estético e minimalista
- 9 Ajudar o utilizador a reconhecer, diagnosticar e recuperar dos erros
- 10 Dar ajuda e documentação



Fases da Avaliação Heurística:

1. Treino pré-avaliação
 - Dar conhecimento aos avaliadores da funcionalidade
 - Informação sobre cenários de interacção
2. Avaliação
 - Individual, seguida de consolidação de resultados
3. Classificação de severidade
 - Determinar a gravidade de cada problema (prioridade)
 - Pode-se fazer primeiro individualmente e depois em grupo
4. Relato (*Debriefing*)
 - Discutir resultados com a equipa de projecto



Exemplo de Problemas

- Problema: Campo da data não indica formato
 - Viola H-5 (“Evitar erros”)
 - Correção: Substituir campo por calendário
- Problema: Tipografia mistura letra maiúscula e minúscula e tipos
 - Viola H-4 (“Consistência e adesão às normas”)
 - atrapalha utilizadores
 - talvez não fosse identificado por testes de utilização
 - Correção: Usar um só tipo em toda a interface

Como relatar (Debriefing):

- Sessão com avaliadores, observadores e equipa de projecto
- Discutir características gerais da IU
- Sugerir possíveis melhoramentos para resolver os principais problemas de usabilidade
- Equipa de projecto avalia os custos de corrigir cada problema
- Sessão de brainstorming
 - Minimizar críticas negativas durante o exercício



Exemplo de Classificação

- Nomes diferentes para a operação “Guardar”
 - H-4: Consistência
 - Descrição: a interface usa “Salvaguardar” no primeiro ecrã para salvaguardar ficheiro do utilizador mas usa “Guardar Ficheiro” nos ecrãs subsequentes. O uso de terminologia diferente para a mesma função pode confundir os utilizadores.
 - Correção: definir uma terminologia e usá-la sempre
 - Severidade: 3



O que é um Defeito de Usabilidade?:

- Um defeito de usabilidade é um potencial problema na operação, aparência ou organização de um sistema que torna o produto final mais difícil de usar pela população de utilizadores alvo.
- Operacionalmente, um defeito é:
 - uma violação clara e evidente dos princípios de usabilidade dados como aceites (princípios como as heurísticas de Nielsen)
 - uma causa provável de atraso na execução das tarefas, confusão ou erros

Graus de severidade dos Defeitos:

1. **Nominal:** algo de aborrecido, ligeiramente incorrecto, atraso insignificante ou pouco frequente, ou pequena probabilidade de erro do utilizador
2. **Menor:** existe uma hipótese de que o problema irá afectar o desempenho, perturbar a aprendizagem ou aumentar de alguma forma os erros do utilizador
3. **Maior:** quando algumas tarefas são substancialmente mais difíceis de executar ou aprender; a probabilidade de erro é grande
4. **Crítico:** defeito óbvio, significativo, reduz a usabilidade geral do sistema ou torna o produto substancialmente mais difícil de usar

Sample Defect Logging Card:

Formulários ou checklists facilitam e aceleram o registo dos defeitos

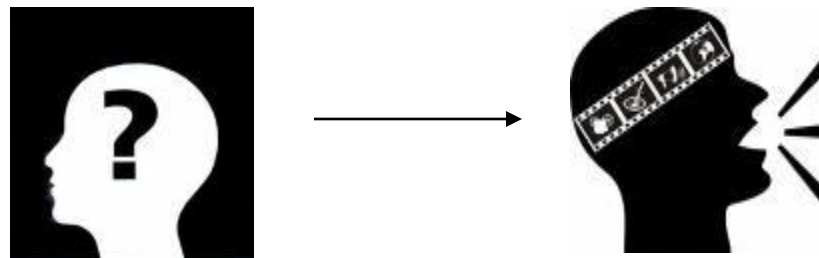
Page _____ Product _____ Recorder _____ Date _____

DEFECT ID

Location ¹	Interface Feature ²			Type of Problem ²			Description/Notes	Severity ³
-0	button color command dialogue field function OTHER:	graphic icon item keystroke label menu	message page palette screen tool window	awkward complex cluttered confusing distracting error handling OTHER:	hidden feature hidden behavior inconsistent missing nonstandard uninformative	behavior feedback layout tolerance visibility workflow		4-critical 3-major 2-minor 1-nominal ?-evaluate
-1	button color command dialogue field function OTHER:	graphic icon item keystroke label menu	message page palette screen tool window	awkward complex cluttered confusing distracting error handling OTHER:	hidden feature hidden behavior inconsistent missing nonstandard uninformative	behavior feedback layout tolerance visibility workflow		4-critical 3-major 2-minor 1-nominal ?-evaluate
-2	button color command dialogue field function OTHER:	graphic icon item keystroke label menu	message page palette screen tool window	awkward complex cluttered confusing distracting error handling OTHER:	hidden feature hidden behavior inconsistent missing nonstandard uninformative	behavior feedback layout tolerance visibility workflow		4-critical 3-major 2-minor 1-nominal ?-evaluate
-3	button color command dialogue field function OTHER:	graphic icon item keystroke label menu	message page palette screen tool window	awkward complex cluttered confusing distracting error handling OTHER:	hidden feature hidden behavior inconsistent missing nonstandard uninformative	behavior feedback layout tolerance visibility workflow		4-critical 3-major 2-minor 1-nominal ?-evaluate

Cenários de Utilização para Inspeções:

- Os cenários de inspecção são exemplos completos de utilização
- Organizam o processo de inspecção combinando tarefas básicas num coerente e correcto guião
- Podem misturar tarefas comuns ou representativas com interacções excepcionais ou não-usuais, ou eventos de interesse
- Asseguram que a avaliação é realizada no contexto de tarefas representativas
- Os utilizadores representam os cenários declarando ou descrevendo acções ou possíveis acções
- Cenários bem estruturados cobrem um subconjunto planeado da IU



Das Tarefas aos Cenários:

Tarefas discretas ou casos de utilização (*essential use cases*):

- avançando para a secção seguinte
- retrocedendo para a secção anterior
- escolhendo uma determinada cena
- ajustando a qualidade de imagem
- gravando um programa futuro
- acerto do relógio...

Cenários de Inspeção:

✓ **Você quer ver uma determinada cena de um filme favorito numa cassette usada. Obtenha uma imagem estável ajustando o tracker, encontre o início da cena e faça play.**

✓ **Quer gravar a Floribela 3 dias seguidos. Você repara que se esqueceu de acertar o relógio. Acerta-o e depois programe o vídeo para as gravações.**





Papéis das Inspeções:

• Lead Reviewer

- Coordena a inspecção, conduz a reunião, segue a agenda e o método
- Mantém o processo em movimento, faz todos participarem
- Protege os utilizadores, controla os programadores

• Inspection Recorder

- Regista os defeitos e inconsistências
- Atribui os graus de severidade iniciais (estimados)
- Separadamente regista features boas, possíveis soluções de design, objecções e opiniões minoritárias
- Organiza, distribui e arquiva os registos

• Continuity Reviewer

- Responsabilidade primária: identificar inconsistências na aparência ou comportamento de toda a interface
- Pode também identificar violações às normas
- Pode monitorizar critérios especiais, ex: regulações governamentais,



Papéis das Inspeções:

• **Usability Specialist**

- Designer IHM, especialista em usabilidade, especialista em ergonomia...
- Assiste o Recorder no que diz respeito à classificação dos defeitos de usabilidade e estimação dos graus de severidade
- Papel de consultor
- É mau sinal se for frequentemente ignorado

• **Utilizadores**

- Actuam nos cenários, comentam primeiro
- Há que encorajá-los, ouvir, tentar compreender
- Apontar ideias e comentários, seguir em frente
- Os utilizadores não são *designers* nem têm a palavra final

• **Programadores**

- Nunca devem explicar ou defender um design
- Nunca devem discutir com os utilizadores
- Nunca devem fazer promessas aos utilizadores

Utilizando os Utilizadores:

- Mesmo sob pressão, deve-se envolver os utilizadores
- Todo o tempo gasto construindo o sistema errado ou programando as funções erradas é desperdiçado
- Capturar os papéis dos utilizadores bem cedo
- Focar-se no **Porquê?** Não deixar que os utilizadores ordenem no **Que** parece a IU nem **Como** funciona
- Usar o tempo dos utilizadores e programadores de forma **Eficiente**
- Envolver os utilizadores, mas não em tudo! Apenas no que é importante:
 - requisitos, features e propriedades, inspeções de usabilidade, testes de campo





Testes Estatísticos:

- Métrica de desempenho: Execução < 30 min.
 - Teste com 6 utilizadores
 - Teste dá: 20, 15, 40, 90, 10, 5
 - Média = 30
 - Desvio padrão = 32
 - Parece OK !?
 - Errado, nada se pode afirmar!
 - Factores que contribuem para esta **incerteza**:
 - Pequeno nº de utilizadores no teste (N = 6)
 - Resultados muito variáveis (desvio padrão = 32)
 - Desvio padrão = dispersão do valor médio
-

• Experimentação Controlada:

- Responder a:
 - Solução A melhor que Solução B?
 - Solução cumpre os objectivos?

• Procedimento:

- Escolha da população significativa
- Formulação da hipótese nula (H0)
- Realização dos testes
- Conclusão

Exercício:**(Exemplo de Bilheteira)**

Estudo de alternativas entre adquirir bilhetes para o cinema numa bilheteira normal ou numa máquina.

- Hipótese nula:
 - a forma de aquisição do bilhete não tem influência no tempo de tarefa
- Medidas
 - bilheteira: 28, 35, 23, 26, 30, 32 segundos
 - máquina: 32, 41, 37, 40, 30 segundos
- Médias
 - bilheteira: 29 segundos
 - máquina: 36 segundos

Exercício:
(Test T - Bilheteira)

	bilheteira	máquina
	28	32
	35	41
	23	37
	26	40
	30	30
	32	
média	29,0	36,0
teste-t	=TTEST(C3:C7;2;3)	

TTEST

Array1

B3:B8

= {28;35;23;26;30;32}

Array2

C3:C7

= {32;41;37;40;30}

Tails

2

= 2

Type

3

= 3

= 0,035767101

Returns the probability associated with a Student's t-Test.

Type is the kind of t-test: paired = 1, two-sample equal variance (homoscedastic) = 2, two-sample unequal variance = 3.

?

Formula result = 0,035767101

Cancel

OK



Exercício:

(Test T - Bilheteira)

- Constata-se que:
 - as duas amostras têm uma probabilidade de (apenas) **3,6%** serem a **mesma** amostra
 - rejeita-se H_0 pois $0.036 < 0.05$ (nível de significância p)
- Conclusão:
 - a compra de bilhetes em máquina é 20 % (36/29) mais lenta com uma probabilidade de **96,4%**



Exercício:

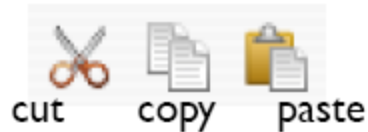
Desenhar uma experiência para testar se adicionar codificação com cores irá aumentar a precisão de uma interface.

- **Sujeitos experimentais:**
 - os mais próximos possíveis da população de utilizadores
- **Hipótese:** a codificação com cores irá tornar a selecção mais precisa
- **Variável Independente:** codificação com cores
- **Variável Dependente:** precisão da interface, medida como o número de erros
- **Design:** entre-grupos, para assegurar que não há transferência de aprendizagem (se não houver sujeitos suficientes, utilizar intra-grupos)
- **Tarefa:** interfaces idênticas nas 2 condições, só que na segunda a cor é adicionada. Apresenta-se aos sujeitos um ecrã com escolhas (ordenadas aleatoriamente) e indicamos o que devem escolher verbalmente. Há um limite de tempo para a selecção.
Conta-se um erro por cada selecção incorrecta ou não realizada.

Exemplo:

- Imagine que está a desenhar um novo programa de processamento de texto e pretende usar ícones. Está a considerar utilizar um de dois estilos de ícones: naturais vs. abstractos. Quer saber qual o design que fará com que os utilizadores melhor se recordem.

Naturais:



Abstractos:



- **Primeira coisa:**

- formular uma hipótese (a hipótese nula):
 - Os utilizadores irão lembrar-se dos ícones naturais mais facilmente do que dos abstractos

- **Variável Independente:**

- tem dois níveis: natural / abstracto

- **Variável Dependente:**

- número de erros na selecção e tempo de selecção de um ícone



Exemplo:

- Assumimos que a velocidade com que o utilizador selecciona um ícone é uma indicação da facilidade de lembrança do ícone!
- Design de uma tarefa e recolha dos tempos

Exemplo: tempos de conclusão das tarefas

sujeito nº	ordem de apresentação	Naturais (1)	Abstractos (2)	Média do sujeito (3)	Natural (1) - (3)	Abstracto (2) - (3)
1	AN	656	702	679	-23	23
2	AN	259	339	299	-40	40
3	AN	612	658	635	-23	23
4	AN	609	645	627	-18	18
5	AN	1049	1129	1089	-40	40
6	NA	1135	1179	1157	-22	22
7	NA	542	604	573	-31	31
8	NA	495	551	523	-28	28
9	NA	905	893	899	6	6
10	NA	715	803	759	-44	44
média		698	750	724	-26	26
desvio p.		265	259	262	14	14

estas médias podem ser comparadas com um teste-t!



Resumo

Factores a considerar na escolha de um método de avaliação:

- **when in cycle is evaluation carried out? design vs implementation**
- **what style of evaluation is required? laboratory vs field**
- **how objective should the technique be? subjective vs objective**
- **what type of measures are required? qualitative vs quantitative**
- **what level of information is required? High level vs low level**
- **what level of interference? obtrusive vs unobtrusive**
- **what resources are available? time, subjects, equipment, expertise**