Interação Humano-Computador

Avaliação Heurística

Prof. Lesandro Ponciano

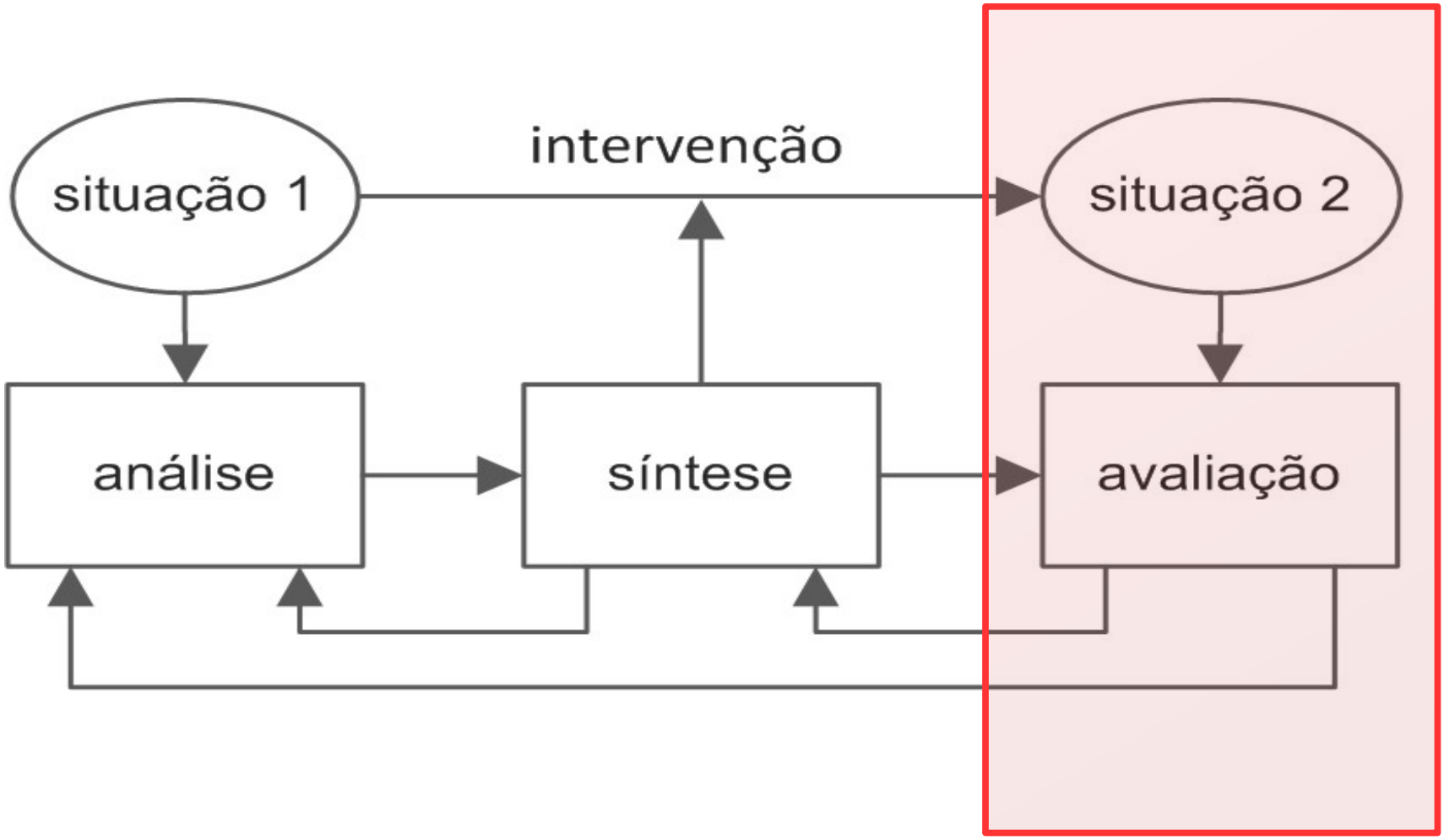
Departamento de Engenharia de Software
e Sistemas de Informação (DES)

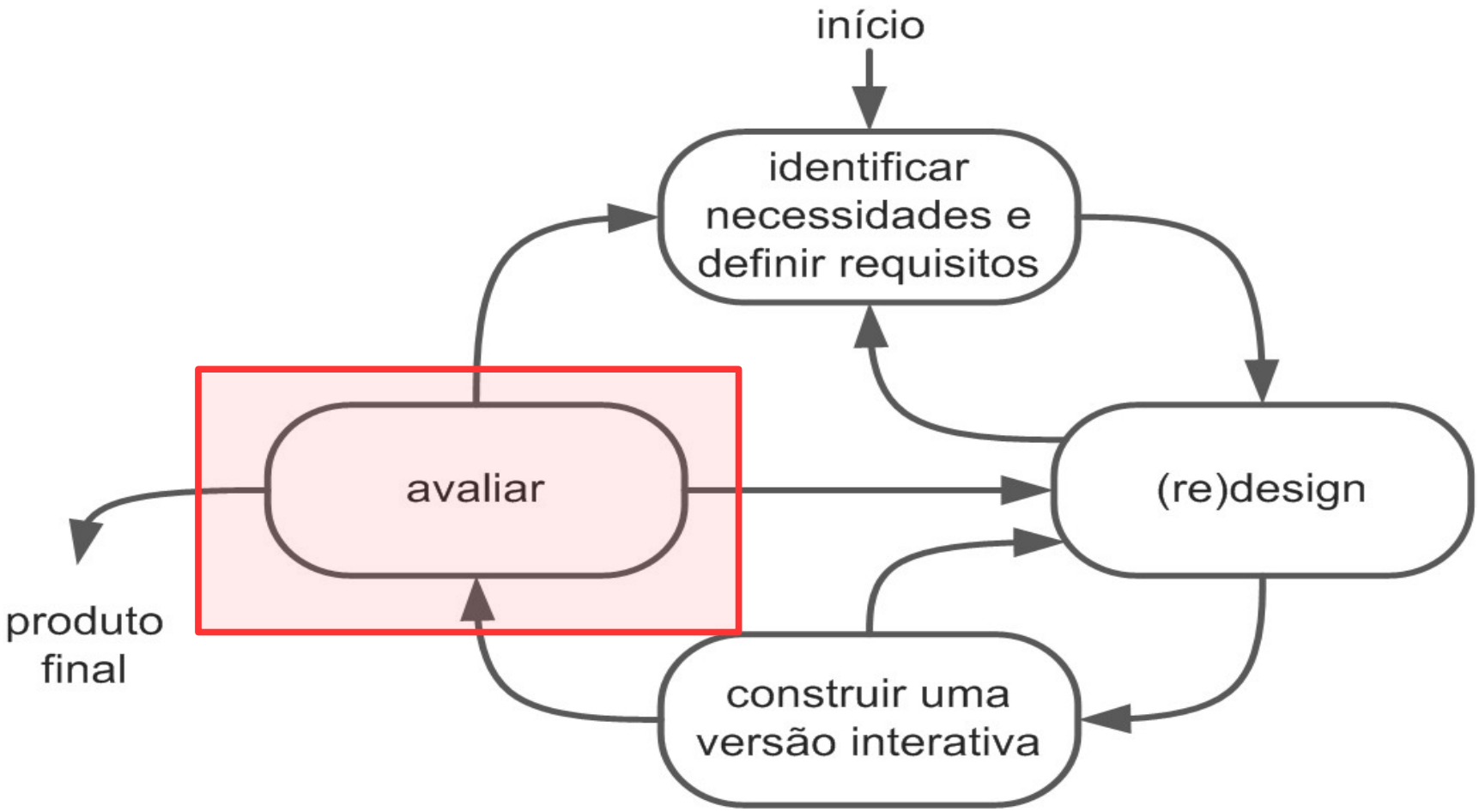
Interação Humano-Computador

Avaliação Heurística

Prof. Lesandro Ponciano

Departamento de Engenharia de Software
e Sistemas de Informação


situação 1
intervenção
situação 2
análise
síntese
avaliação


início
identificar necessidades e definir requisitos
avaliar
(re)design
produto final
construir uma versão interativa

Avaliação por Meio de Inspeção

- Características
  - Não envolve a participação de usuários
  - Um avaliador tenta se colocar no lugar do usuário enquanto examina (ou inspeciona) uma solução de IHC
  - Permite identificar problemas que os usuários podem vir a ter quando interagirem com o sistema
  - Permite propor soluções para os problemas

- Alguns métodos de inspeção em IHC são:
  - avaliação heurística
  - percurso cognitivo
  - método de inspeção semiótica

Avaliação Heurística

- Método de avaliação criado para encontrar problemas de usabilidade durante um processo de design iterativo
- Método simples, rápido e de baixo custo quando comparado aos métodos empíricos
- Tem como base um conjunto de heurísticas de usabilidade
  - Características desejáveis da interação e da interface

Heurísticas de Nielsen

- Conjunto de 10 heurísticas (podem ser complementadas)
  1) Visibilidade do estado do sistema
  2) Correspondência entre o sistema e o mundo real
  3) Controle e liberdade do usuário
  4) Consistência e padronização
  5) Reconhecimento em vez de memorização
  6) Prevenção de erros
  7) Flexibilidade e eficiência de uso
  8) Projeto estético e minimalista
  9) Ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros
  10) Ajuda e documentação

Visibilidade do Estado do Sistema

- O sistema deve sempre manter os usuários informados sobre o que está acontecendo
  - *feedback* adequado
  - *feedback* no tempo certo

Correspondência Sistema - Mundo Real

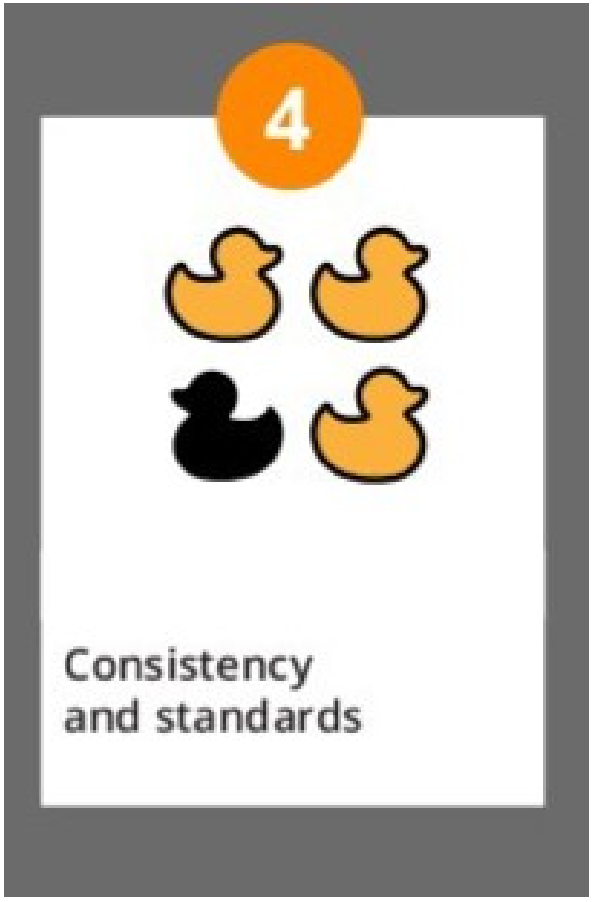
- O sistema deve utilizar palavras, expressões e conceitos que são familiares aos usuários,
  - Evitar termos orientados ao sistema ou jargão dos desenvolvedores
- O designer deve seguir as convenções do mundo real
  - Fazer com que a informação apareça em uma ordem natural e lógica, conforme esperado pelos usuários

Controle e Liberdade do Usuário

- Os usuários frequentemente realizam ações equivocadas no sistema e precisam de uma “saída de emergência” claramente marcada para
  - Sair do estado indesejado
  - Não ter de percorrer um diálogo extenso
- A interface deve permitir que o usuário desfaça e refaça suas ações

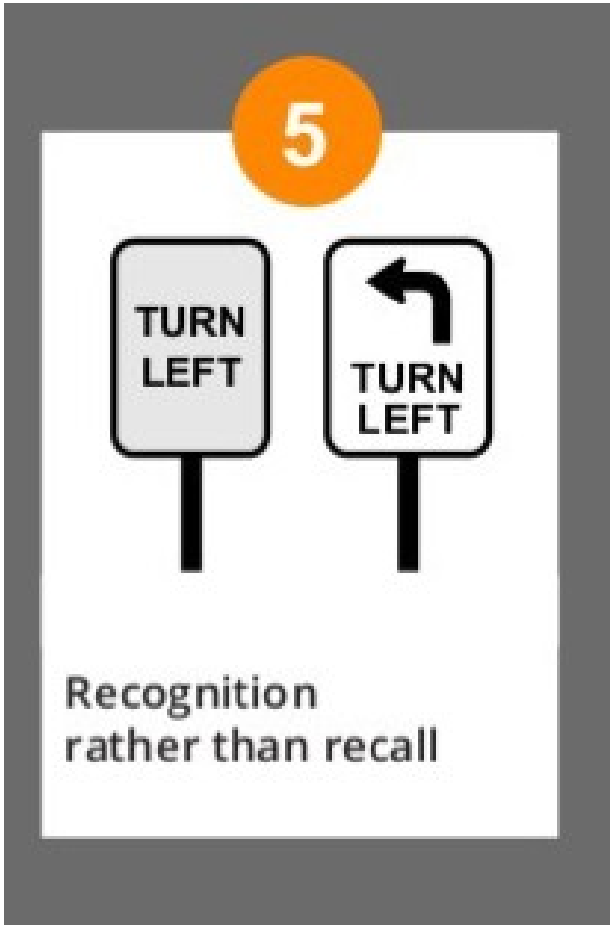
Consistência e Padronização

- Os usuários não devem ter de se perguntar se palavras, situações ou ações diferentes significam a mesma coisa
- O designer deve seguir as convenções da plataforma ou do ambiente computacional



Reconhecimento em Vez de Memorização

- O designer deve tornar os objetos, as ações e opções visíveis
- As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário



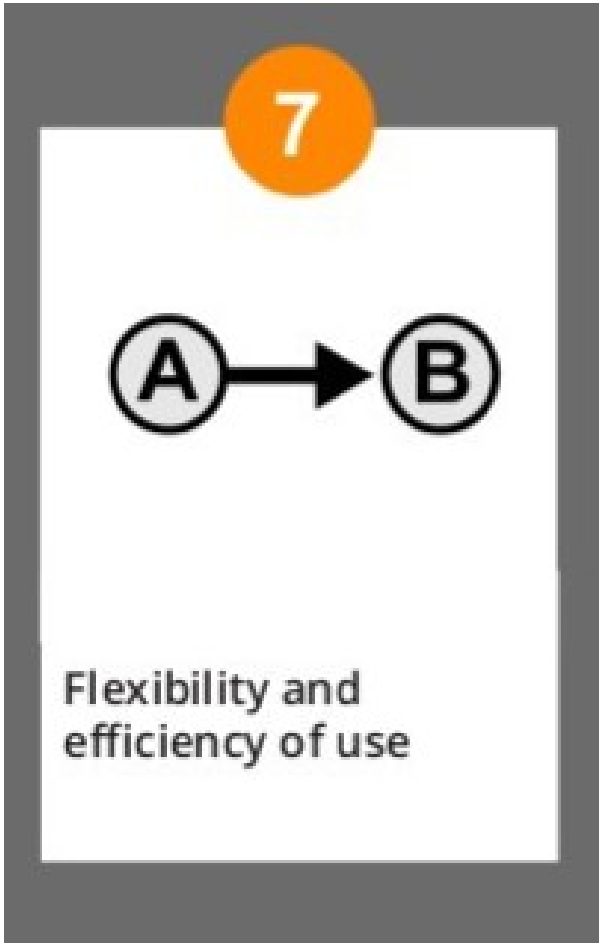
Prevenção de Erros

- Melhor do que uma boa mensagem de erro é um projeto cuidadoso que evite que um problema ocorra, caso isso seja possível

Flexibilidade e Eficiência de Uso

- Permitir que o sistema consiga servir igualmente bem os usuários experientes e inexperientes

- Aceleradores podem tornar a interação do usuário mais rápida e eficiente



Projeto Estético e Minimalista

- A interface não deve conter informação que seja irrelevante ou raramente necessária

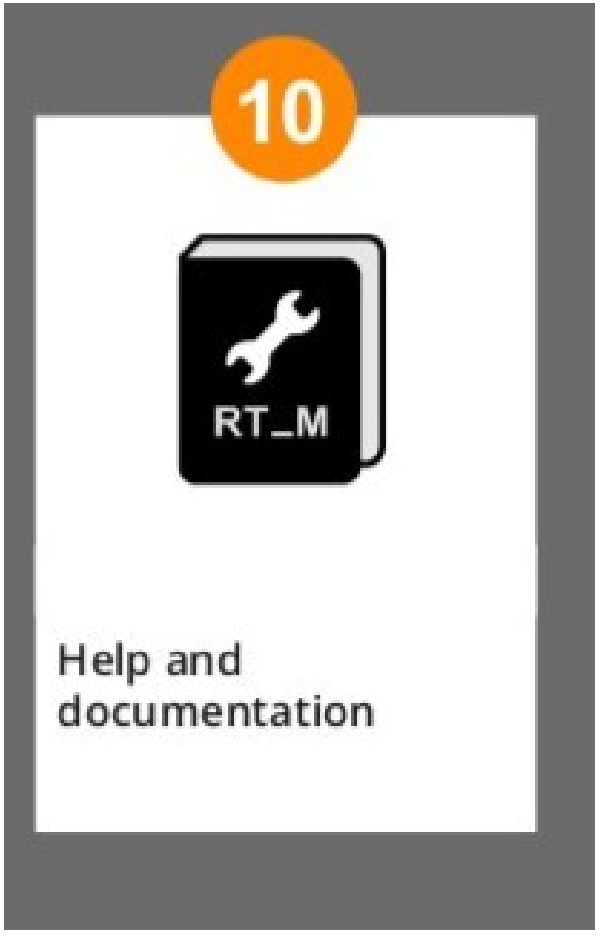
- Cada unidade extra de informação em uma interface reduz sua visibilidade relativa, pois compete com as demais unidades de informação pela atenção do usuário

Ajudar os Usuários com os Erros

- Reconhecer, diagnosticar e recuperar de erros
- As mensagens de erro devem ser expressas em linguagem simples
  - sem códigos indecifráveis
- Indicar precisamente o problema e sugerir uma solução de forma construtiva

Ajuda e Documentação

- É necessário oferecer ajuda e documentação de alta qualidade
  - Tais informações devem ser facilmente encontradas
  - Devem ser focadas na tarefa do usuário
  - Devem enumerar passos concretos a serem realizados
  - Não devem ser muito extensas



Atividades da Avaliação Heurística

avaliação heurística	
atividade	**tarefa**
Preparação	*Todos os avaliadores:* ▪ aprendem sobre a situação atual: usuários, domínio etc. ▪ selecionam as partes da interface que devem ser avaliadas
Coleta de dados Interpretação	*Cada avaliador, individualmente:* ▪ inspeciona a interface para identificar violações das heurísticas ▪ lista os problemas encontrados pela inspeção, indicando local, gravidade, justificativa e recomendações de solução
Consolidação dos resultados Relato dos resultados	*Todos os avaliadores:* ▪ revisam os problemas encontrados, julgando sua relevância, gravidade, justificativa e recomendações de solução ▪ geram um relatório consolidado

Relato de Problemas

- Para cada problema identificado, o avaliador deve anotar:
  - qual heurística foi violada
  - em que local o problema foi encontrado (em que tela e envolvendo quais elementos de interface)
  - qual a gravidade do problema
  - uma justificativa de por que aquilo é um problema
  - também pode anotar ideias de soluções

Severidade de Problemas

- A severidade de um problema envolve três fatores:
  - a **frequência** com que o problema ocorre
    - é um problema comum ou raro?
  - o **impacto** do problema, se ocorrer
    - será fácil ou difícil para os usuários superarem o problema?
  - a **persistência** do problema
    - o problema ocorre apenas uma vez e será superado pelos usuários, ou atrapalhará os usuários repetidas vezes?

Escala de Severidade

- Nielsen sugere a seguinte escala de severidade
  - **problema cosmético**: não precisa ser consertado a menos que haja tempo no cronograma do projeto
  - **problema pequeno**: o conserto pode receber baixa prioridade
  - **problema grande**: importante de ser consertado e deve receber alta prioridade
    - Esse tipo de problema prejudica fatores de usabilidade tidos como importantes para o projeto
  - **problema catastrófico**: é extremamente importante consertá-lo antes de se lançar o produto
    - O problema pode impedir que o usuário realize suas tarefas e alcance seus objetivos

Obrigado!

Lesandro Ponciano

Referências

BARBOSA, Simone D. J; SILVA, Bruno Santana da. Interação humano-computador. Rio de Janeiro (RJ): Elsevier, 2010. 384 p. (capítulo 6, 7)

BENYON, David. Interação humano-computador. 2. ed. São Paulo: Pearson Prentice Hall, 2011. xx, 442 p. ISBN 9788579361098

ROGERS, Yvonne; SHARP, Helen; PREECE, Jennifer. Design de interação: além da interação homem-computador. 3. ed. Porto Alegre: Bookman, 2013. xiv, 585 p. ISBN 9788582600061