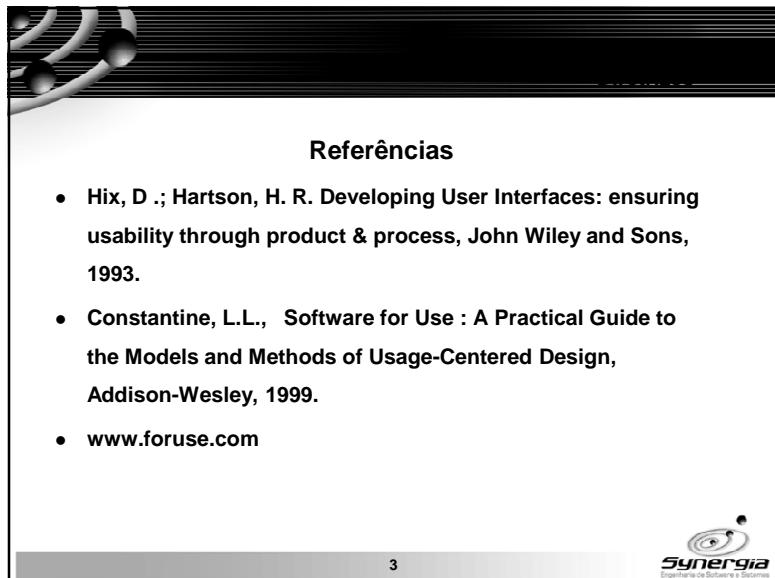
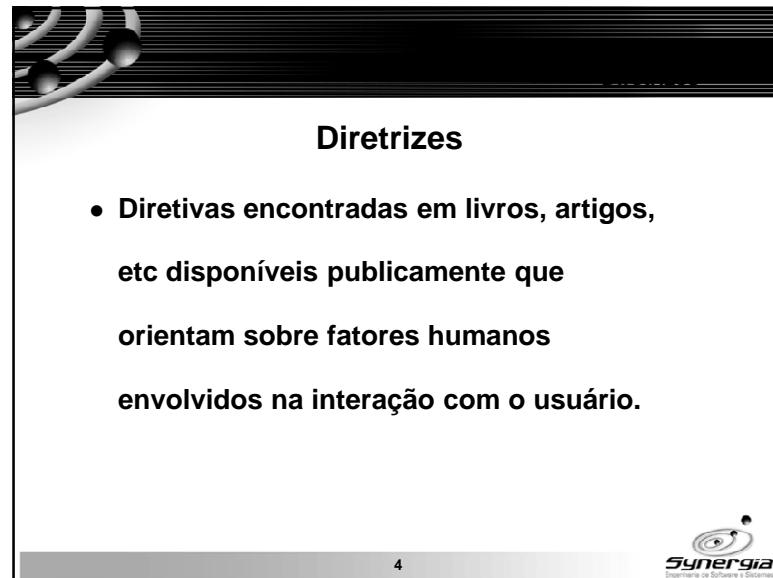


2



3



4

Diretrizes

- Mandamentos de Usabilidade: diretrizes em mais alto nível
- Princípios de desenho
- Gerais

5



Mandamentos de Usabilidade

1: Acesso: **o sistema deve ser acessível sem *help* ou documentação para o usuário que tem conhecimento do domínio da aplicação.**

6



2: Eficiência: **o sistema não deve interferir ou impedir no uso eficiente por usuários habilidosos e com experiência no sistema.**

7



3: Progressão: **o sistema deve facilitar o avanço contínuo em conhecimento e habilidade e acomodar mudanças progressivas à medida que o usuário ganha experiência com seu uso.**

8



4: Apoio: **o sistema deve apoiar o trabalho real que os usuários precisam executar, tornando-o mais fácil, simples, rápido ou mais divertido ou tornando possíveis coisas novas.**

9



5: Contexto: **o sistema deve ser adequado para as condições reais e presentes de trabalho do contexto operacional onde é utilizado.**

10



Princípios de desenho

1: Estrutura: **organize a interface de modo que seja útil e faça sentido, consistente com modelos mentais dos usuários, colocando coisas relacionadas juntas e separando coisas não relacionadas, diferenciando coisas diferentes e fazendo coisas semelhantes lembrar uma às outras.**

11



2: Simplicidade: **faça tarefas simples e comuns poderem ser realizadas de forma simples, comunicando de forma clara e objetiva na linguagem do usuário.**

12



3: Princípio da visibilidade: **mantenha visíveis as opções necessárias para a realização de uma tarefa, sem distrair o usuário com informação redundante e fora de contexto.**

13



4: Feedback: **mantenha os usuários informados de ações e interpretações, mudanças de estado ou condição, erros ou exceções que sejam de seu interesse através de uma linguagem clara, concisa, não ambígua e familiar ao usuário.**

14



5: Tolerância: **seja flexível e tolerante, reduzindo os custos de erros e mau uso através da disponibilização de “desfazer” (*undos*) e “refazer” (*redos*), prevenindo erros, tolerando uma variedade de entradas e seqüências e interpretando todas ações razoáveis de uma maneira razoável.**

15



6: Reuso: **reuse componentes e comportamentos internos e externos, mantendo consistência com propósitos ao invés de fazê-la arbitrariamente, desse modo reduzindo a necessidade dos usuários pensarem e se lembrarem.**

16



- Desenho centrado no usuário: **nem sempre o que é bom para o desenvolvedor é bom para o usuário.**

17



- Conheça o usuário: **conheça o comportamento do usuário relacionado com o sistema a ser desenvolvido.**

18



- Envolva o usuário
 - **Desenvolvimento participativo: a participação do usuário desde o início é recomendável.**
 - **O conhecimento e o comprometimento do usuário são importantes.**
 - **Escolha bem o usuário.**

19



- Invista tempo em desenho
 - **O tempo investido em melhorar um desenho representa economia de tempo em interação com o usuário.**

20



- Previna erros do usuário
 - **Antecipe as reações do usuário.**
 - **Torne indisponível funções não pertinentes.**
 - **Peça confirmação de ações de riscos.**

21



- Aperfeiçoe as operações do usuário
 - **Permita o uso de atalhos.**
 - **Imponha processos otimizados via seqüências de ações.**

22



- Mantenha o usuário no controle da situação
 - **O sistema deve ser previsível.**
 - **Mensagens devem ser consistentes.**
 - **Cuidado com o 'decurso de prazo':**
 - aquelas mensagens indicando ações que são efetivadas se o usuário não se manifesta dentro de certo tempo.

23



- **Ajude o usuário a iniciar-se no sistema.**

24



- Garanta que o usuário forme um modelo mental compatível com o modelo real

25



- Consistência e simplicidade.

26



- Considere limitação humana de memória
 - **Utilize fechamentos.**
 - **Use lista de opções .**

27



- Considere questões de cognição
 - Use pistas cognitivas: ex. ctrl C para copiar.
 - Use analogias com mundo real, com cuidado porque pessoas diferentes podem entender de maneira diferente as analogias utilizadas.

28



- Use *feedback* informativo
 - **Feedback articulatório.**
 - **Feedback semântico.**

29

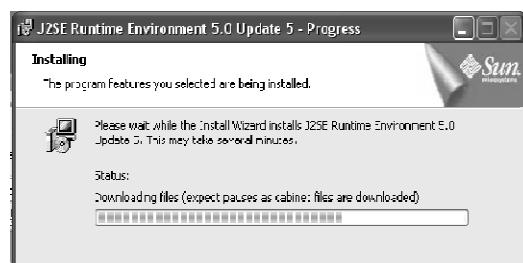


- Use indicadores de status e progresso apropriados
 - Ajuda na redução do tempo percebido pelo usuário.
 - **Tempo de resposta tolerado:**
 - 50 a 150 ms para clicks;
 - 1 seg para tarefas simples e freqüentes;
 - até 12 segundos para tarefas complexas.
 - Para tarefas com mais de 2 a 4 segs usar indicador de progresso .

30



- Exemplo de indicador de progresso criando expectativas corretas.



31



- Use mensagens apropriadas
 - Use mensagens centradas no usuário e nas tarefas, que ele possa entender.
 - Use mensagens positivas, não ameaçadoras: evite coisas como “erro fatal, execução abortada”.
 - Evite ser “espirituoso”.

32



- Use mensagens apropriadas
 - Use termos específicos apropriados: ex. ao invés de “dado ilegal” use “dado fora do limite permitido de 0 a 99”.
 - Ponha a “culpa” de erros no sistema: ex. ao invés de “comando ilegal” use “o sistema não reconhece este comando”.

33



- **Cuidado com antropomorfismo: computador não é gente nem amigo da gente!**

34



- Use diálogos modais com cuidado
 - Elementos modais que exigem atenção e impedem outras ações do usuário até que alguma tarefa seja completada: usar quando realmente necessário.

35



- Permita o usuário reverter ações facilmente.

36



- Prudência ao exigir a atenção do usuário
 - **Evite pisca-pisca e alertas sonoros.**
 - **Cuidado com negritos e sublinhados.**
 - Caixa alta tem tempo de leitura aumentado em 10%.
 - **Cuidado com o uso de cores, especialmente com seu significado em uma dada cultura.**

37



- Uso apropriado de telas
 - Mantenha a inércia das telas: reduzir mudanças entre telas.
 - Posição de objetos deve ser consistente entre telas.
 - Uso do espaço branco: produtividade de leitura cai muito quando se usa menos de 25% de espaço branco; 50% seria recomendado para tela mais textuais.
 - Organize telas para gerenciar a complexidade: exemplo de ganho de 40% em produtividade conseguido com uma melhor organização da informação em uma tela.

38



- Considere diferenças de usuários
 - **Permita personalização, que deve manter-se mesmo entre sessões de uso.**

39



- Considere diferenças de usuários
 - novo: tem **desconhecimento sintático e semântico**.
Demandas:
 - clareza;
 - simplicidade;
 - pequeno número de funções mais úteis;
 - mensagens claras;
 - *feedback* informativo;
 - manuais completos, tutoriais e demonstrações.

40





- Considere diferenças de usuários
 - intermitente: **mantém conhecimento semântico mas se esquece do conhecimento sintático.** Demanda:
 - comandos simples e consistentes;
 - seqüência lógica de passos;
 - funções fáceis de se lembrar;
 - assistência e help *on-line*
 - *manuais concisos.*

41


Synergia
Engenharia de Software e Sistemas



- Considere diferenças de usuários
 - freqüentes: **tem bons conhecimentos sintáticos e semânticos.**

Demandas:

- interação rápida;
- comandos poderosos;
- digitação reduzida;
- mensagens breves com acesso a detalhes só se necessário;
- feedback conciso;
- personalização da interface.

42


Synergia
Engenharia de Software e Sistemas



- Cuidado com diretrizes conflitantes
 - Exemplo de uso do “daytime” no sistema operacional e no correio eletrônico:
 - d: delete
 - a: all
 - y: yes
 - t: type
 - i: inverted order
 - e: exit

43


Synergia
Engenharia de Software e Sistemas



- Cuidado com diretrizes conflitantes
 - Apresentação e remoção de mensagens na tela: deve-se manter sob o controle do usuário ou fazer a mensagem desaparecer após um tempo? (sugere-se utilizar remoção automática mas com tempo de exibição apropriado)
 - Exemplo de caixa de mensagens com os botões “Help”, “Cancel” e “OK”. Qual seria o *default*? Poderia ser o mais utilizado, ex. “OK”, ou o mais seguro, ex. “Cancel”

44


Synergia
Engenharia de Software e Sistemas