

The Design of Everyday Things

Donald A. Norman
Sensibilizando para o *design*

DOET: caps. 1 a 4
Livro NIED: Cap 1

Princípios de Norman

Modelo conceitual
Visibilidade
Mapeamento
Restrição
Feedback
Affordances
Errar é humano
Projeto centrado no usuário

Princípios de Norman

Quais são?

- Princípios do bom *design* envolvem
 - Bom modelo conceitual:** dar ao usuário um bom modelo conceitual,
 - Visibilidade:** deixar visível o estado e as funções do dispositivo,
 - Bons Mapeamentos:** usar bons mapeamentos,
 - Feedback:** dar *feedback* apropriado.
 - explorar as *affordances*
 - impor *restrições*,

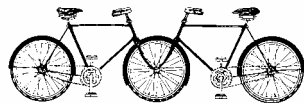
Modelos Conceituais

- Pessoas formam um *modelo conceitual* de como as coisas funcionam ...
 - ex: *bicicleta convergente*
 - Ex: carro
- Sugestionamentos (*clues*) ajudam a construir esse modelo
 - affordances*, ex: *tesoura*
 - restrições*, ex: *tesoura*
 - mapeamentos*, contra-ex: *relógio digital típico*

Modelos Conceituais



Carelman's Tandem "Convergent Bicycle (Model for Fiancés)
Carelman's "Coffee Pot for Masochists"
Jacques Carelman, *Catalog of Unfindable Objects*,
Balland.



Modelos Conceituais

Ex: tesoura... :

- Affordances
- Restrições
- Mapeamento – aparência física sugere como inserir os dedos
- Modelo Conceitual – Sugerido pela maneira como as partes se ajustam e se movem

Modelo Conceitual

Modelo do *Designer* x Modelo do Usuário!

Usuário forma seu modelo a partir da experiência e do uso...
(imagem do sistema)

Problemas se os modelos do usuário e do designer são diferentes!!

- Forneça um bom modelo conceitual
 - Um bom modelo permite prever os efeitos da ação
 - Na ausência de um modelo: operação é cega... pode até funcionar, mas ... o que acontece quando algo sai errado?
 - Objetos do dia a dia: modelos simples!
 - Ex:
 - Contra-ex:

Visibilidade

Alguns artefatos obscurecem a funcionalidade

comum quando o número de controles é muito menor que o número de funções

Garanta que funções sejam visíveis
recursos perceptíveis e interpretáveis
quando isso ocorre, a pessoa não tem que 'lembrar'

Demandas conflitantes : recursos, estética, abstrações

Visibilidade

- Exemplo: torneira?
- Contra-exemplo: telefone digital moderno
 - lista dos recursos desejáveis...
 - Como acessá-los?

Mapeamentos

- O princípio do mapeamento
 - Relacionamento entre dois objetos...
 - Mapeamento de intenções em ações
 - Mapeamento natural:
 - explora as vantagens das analogias físicas e padrões culturais
 - ex: ajuste do banco no Mercedes-Benz...
 - ex: **fogão**...

Restrições

Sinalize ações possíveis / apropriadas
físicas (disquetes, chaves, tesoura)
semânticas (sombreamento do menu)
culturais (vermelho/verde)
lógicas (espacial)

ex: tomada

Feedback

- O princípio do *feedback*
 - Enviar ao usuário informação sobre quais ações foram efetivamente executadas, que resultado foi obtido...
 - Pode ser sonoro, ou mudança no estado físico do objeto...
- ex:

Affordances:

- Lembrança de 3 objetos

Affordances

- *Propriedades percebidas de um objeto*
 - Portas:
 - Muros branquinhos:
- Sugerem um conjunto de ações possíveis; restrições limitam o conjunto de alternativas
- Combinação de
 - boa visibilidade
 - mapeamento natural
 - Restrições
 - *feedback*

Affordances:

- Objetos de uso diário...
 - Quantos você consegue identificar? Quarto, banheiro, casa, carro, sala de aula, campus...
 - Irving Biederman (percepção visual): "*30.000 readily driscriminable objects for the adult*"
 - O tempo necessário para aprender a usá-los a uma taxa de um objeto por minuto é...

Affordances:

- Irving Biederman (percepção visual): "*30.000 readily driscriminable objects for the adult*"
 - O tempo necessário para aprender a usá-los a uma taxa de um objeto por minuto é...

Errar é Humano

- "When simple things need pictures, labels, or instructions, the design has failed."

User-Centered Design

- O Design deveria:
 - Deixar claro quais ações podem, e quais não podem, ser executadas com um objeto,
 - Deixar o modelo conceitual visível ao usuário, incluindo ações alternativas e resultados das ações,
 - Deixar o estado do sistema visível
 - Usar mapeamentos naturais sempre que possível

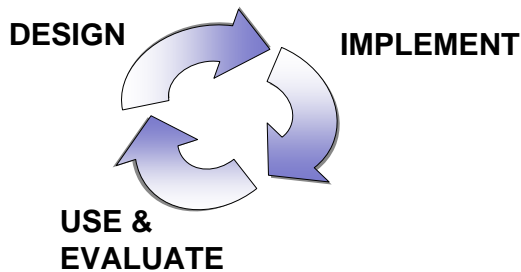
Design Centrado no Usuário

User Centered Design

Design Centrado no Usuário

- Abordagem de *design* que fundamenta o processo em informações sobre as pessoas que irão usar o produto
 - Uma maneira de forçar os projetistas a identificar e considerar os fatores humanos relevantes para o projeto
 - Ajuda a reduzir o número de decisões não fundamentadas e a focalizar as atividades de projeto
 - Ajuda a documentar e justificar decisões que deverão ser revistas mais tarde

Design Centrado no Usuário



UCD: 9 passos

1. Defina o Contexto
2. Descreva o Usuário
3. Analise as Tarefas
4. Aloque as Funções
5. *Layout* do Sistema/ *Design* Básico
6. *Mockups* & Protótipos
7. Testes de Usabilidade
8. Teste *Iterativo* & *Redesign*
9. Atualizações & Manutenções

1. Defina o Contexto

- Contexto: os "tipos" de usos, aplicações
 - Sistemas críticos, aplicações
 - Industriais, comerciais, militares, científicos, consumidor
 - Escritório, casa, entretenimento
 - Exploratório, criativo, cooperativo
- Mercado
- Cliente (não é necessariamente o Usuário)
...Impactos no Design?

2. Descreva o Usuário (!!)

- Atributos Físicos
(idade, gênero, tamanho, alcance, ângulos visuais, etc...)
- Ambiente de Trabalho Físico
(altura da mesa, níveis de ruído, condições de iluminação, versão dos softwares...)
- Habilidades Perceptuais
(audição, visão, sensibilidade ao calor, ...)
- Habilidades Cognitivas
(capacidade de memorização, habilidade de leitura, treinamento musical, habilidades matemáticas...)
- Personalidade e habilidades sociais
(gostos, preferências, aversões, paciência, ...)
- Diversidade cultural e internacional
(línguas, fluxo de diálogo, símbolos ...)
- Populações especiais (limitações físicas, mentais)

3. Analise as Tarefas

- Converse com e observe usuários (NÃO clientes) executando suas tarefas
- Liste todas e cada uma das TAREFAS
- Subdivida as tarefas em PASSOS

🔗 **ABSTRAIA** em tarefas típicas (monitorar, diagnosticar, prever, controlar, inspecionar, transmitir, receber, decidir calcular, armazenar, escolher, operar, etc.)

4. Aloque as Funções

- Considere o sistema como um todo!
- Decida quem, ou o que, é mais adequado para executar cada tarefa (ou cada passo)
 - e.g., sistema lembra o login id, e lembra o usuário, mas o usuário lembra o password
- Baseie suas decisões no conhecimento do hardware e software do sistema, e das habilidades, cultura, protocolos de comunicação, privacidade, etc. dos usuários

...não esqueça as implicações para o design!...

5. Layout do Sistema / Design Básico

- Sumarize os componentes e o seu *design* básico
- Verifique em relação aos Documentos de Requisitos; Referências sobre Fatores Humanos; Especificações de Hardware; Orçamentos; Legislação; etc.
- Garanta que o sistema satisfaz o *design* e atende às restrições
(Verificação e Validação, na linguagem de Engenharia de Software)

6. Mockups & Protótipos

- "Informed Brainstorming"
- RAPIDAMENTE gere 'mock ups' das interfaces com os usuários para teste com pessoas reais
- Caneta e papel ou quadro branco para começar
- *Iterate, iterate, iterate!*
- Gradualmente aumente a funcionalidade e veracidade
- Liste e descreva recursos auditivos e visuais no mesmo nível de detalhe nos protótipos (i.e. não ignore um ou outro)

7. Teste de Usabilidade

- Peça para usuários reais (ou representativos) executarem suas tarefas usando os protótipos
- Busque *feedback* subjetivo e objetivo. As vezes os usuários "querem" recursos que na verdade se mostram inadequados
Video tape, muitas anotações, questionários
- Seja rigoroso sempre que possível (estatísticas, etc.)
- Use o *feedback* nas avaliações iterativas & no *redesign* do sistema
- "Discount" *usability testing* pode ser bastante efetivo: pode demandar menos sujeitos e gerar resultados mais rápidos

8. Teste Iterativo & Redesign

- Repita os ciclos de teste e redefinição do sistema, sujeito às restrições de tempo e custo
- Focalize na Funcionalidade Primeiro (*Functionality First!*)
- Planeje várias versões ao longo do desenvolvimento
 - Com teste, redesign, teste...

9. Atualizações & Manutenção

- Obtenha *feedback* no ambiente de uso, dados de usuários, logs, *surveys*, etc.
- Analise e faça as recomendações para redesign/testes iterativos
- Planeje atualizações e manutenção como parte do *design*!
design deve permitir que o sistema seja modificado ou atualizado

UCD: Maximizando o Esforço

- No mundo real existem limitações e restrições
- Eliminar passos não é a melhor forma de economizar!
- Otimize a eficiência de cada passo
- *Em resumo:* Focalize no contexto e no usuário, para obter o máximo benefício do tempo dispendido