

## Prototipagem e Cenários

Veja antes, decida depois

## Porque prototipar?

Falhas identificadas rapidamente

Possibilidade de explorar o "*design space*"

Protótipos: artefatos que simulam parte das características do sistema desejado

## Dilema

- Você não pode avaliar o *design* até que esteja construído
  - Mas...
- Depois de construir, fica difícil mudar
- Simular o *design*, a um custo baixo

## Prototipagem & Design Iterativo

Difícil obter um bom *design* no começo  
Problema do ovo e da galinha...

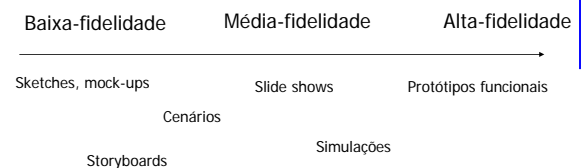
Construa sistemas *parciais*, avalie, repita

Prototipar o que??? Como?? Quando?

## Artefatos do Design

- Como expressar idéias iniciais do *design*?
  - Nenhuma codificação de software nesse estágio (no caso de SCC560: parte 2!)
- Noções chave
  - Rapidez!!!
  - Flexibilidade (para expressar *designs* radicalmente diferentes)
  - Barato
  - Assegurar *feedback*

## Prototipação



## Dimensões da Prototipagem

- 1. Representação
  - Como o *design* deve ser descrito/representado?
  - Pode ser uma descrição textual ou representações visuais e diagramas, ou mock-up da interface, ou um sistema...
- 2. Escopo
  - É só a interface (*mock-up*) ou inclui algum componente computacional?
  - Prototipação horizontal: implementação parcial (incompleta) de grande parte da funcionalidade do sistema;
  - Prototipação vertical: implementação total (completa) de um sub-conjunto de funcionalidades do sistema.

## Dimensões (cont)

- 3. Executabilidade
  - O protótipo é "executável"?
  - Se está sendo codificado, haverá períodos em que não será possível executar
  - Necessária para dar autenticidade à interação
- 4. Maturidade
  - Quais são os estágios do produto a medida que evolui?
    - Revolucionário - Joga fora o *design* anterior
    - Evolucionário - Muda gradativamente o *design* anterior (descartável vs incremental)

## Terminologia (1)

- Prototipação Precoce (*Early prototyping*)
  - Na fase de análise, para elicitar ou validar requisitos
- Prototipação Intermediária (*Middle prototyping*)
  - Durante o *design*, para confirmar o comportamento ou validar aspectos chave do design
- Prototipação Tardia (*Late prototyping*)
  - Na fase de implementação, para investigar parâmetros operacionais importantes, particularmente relacionados ao desempenho

## Terminologia (2)

- Fidelidade refere-se ao nível de detalhamento adotado

Protótipo de baixa-fidelidade (*Low-fi*)  
Esboços com muitos detalhes ausentes

Protótipo de alta-fidelidade (*High-fi*)  
Protótipo é semelhante ao produto final

## Terminologia (3)

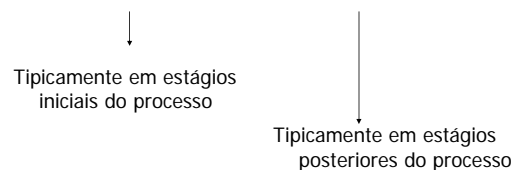
- Protótipo Horizontal

Muito amplo, simula ou mostra muito da interface, mas de uma forma grosseira
- Protótipo Vertical

Poucas características ou aspectos da interface são simulados, mas isso é feito em grande nível de detalhe

## Métodos de Prototipação Rápida

- Non-computer vs. computer-based*



## Métodos Non-Computer-Based

- Objetivo: expressar idéias sobre o *design* do sistema e obter opiniões de forma rápida e barata
- Métodos?

## Descrição do Design

- Pode ser simplesmente uma descrição textual de um possível *design*
  - Limitações óbvias, pois está muito longe de um sistema real
  - Inadequado para representar aspectos visuais da interface

## Cenários

- Situações de uso hipotéticas ou ficcionais
  - Tipicamente envolvendo alguma pessoa, evento, situação e ambiente
  - Deve fornecer o contexto de operação
  - Em geral em formato narrativo, mas pode ser na forma de esboços, ou mesmo vídeos

## Cenários

- Descrição informal, em linguagem corrente, das situações de uso de um sistema.
- Deve ser lido por todas as partes interessadas.
- São uma poderosa e rica forma de descrição, análise e exploração.
- Existem várias técnicas para cenários
  - Questionamento Sistemático
  - Casos de Uso

Para fazer uma compra o usuário precisa escolher um produto. Ele pode escolher navegando pelas páginas do site ou utilizando o serviço de busca. Após a escolha do produto, ele pode efetuar o pagamento eletrônico. Para isto é necessário fazer o seu cadastro. Os produtos são cadastrados no sistema pelo funcionário da loja.

## Porque usar cenários

- Atraente (até divertido)
  - Foco nas necessidades dos usuários
- Permite ao *designer* olhar o problema sob a perspectiva de diferentes pessoas
- Facilita *feedback* e opiniões
- Pode ser bastante futurístico e criativo
- Pode ser compartilhado com diferentes audiências
- Permite explorar erros e equívocos
- Pode ser usado com Storyboarding

## Envolver usuários

- Técnicas requerem input do usuário
- Explicar os *designs*, explicar o que vai fazer e como funciona
- Todos os designs sujeitos a revisão
- Importante ter representações visuais e/ou demos
- Pessoas reagem diferente com explicações verbais

## Storyboarding

- Simulação em papel e lápis, ou *walkthrough* da aparência e funcionalidade do sistema
  - Usa seqüência de diagramas/desenhos
  - Mostra *snapshots* chave
  - Rápido & Fácil

## Sketches, Mock-ups

- "Desenhos" das interfaces, em papel
- Bom para *brainstorming*
- Ajuda as pessoas a focar em noções de alto nível do *design*
- Menos adequados para ilustrar fluxo e detalhes
- Barato e rápido -> *feedback* útil

## Outras Técnicas

- Tutoriais & Manuais
  - Pode escrevê-los antecipadamente para identificar funcionalidade
  - Força o *designer* a ser explícito sobre decisões
  - Colocar no papel é sempre interessante

## Métodos Computer-Based

- Simular mais a funcionalidade do sistema
  - Usualmente apenas alguns aspectos ou características
  - Pode focar melhor em detalhes
  - Em geral, atraente
  - Perigo: usuários ficam mais relutantes em sugerir mudanças quando vêem um protótipo mais 'realístico'

## Ferramentas de Prototipação

- Pontos positivos
  - Fácil desenvolver & modificar telas
  - Suportam o tipo de interface que você está desenvolvendo
  - Suportam diferentes dispositivos de I/O
  - Fácil ligar telas e modificar links
  - Permitem chamadas a procedimentos e programas externos
  - Permitem importar texto, gráficos e outras mídias
  - Fáceis de aprender e usar
  - Bom suporte dos fornecedores

## Ferramentas de Prototipação

- Wizard of Oz - Pessoa simula e controla o sistema "atrás das cortinas"
  - Usa *mock* interface e interage com usuários
  - Bom para simular um sistema que seria difícil de construir

Pode ou não ser *computer-based*

## Wizard of Oz

- 'Wizard' humano simula a resposta do sistema
  - interpreta o user input de acordo com um algoritmo
  - controla o computador para simular a saída apropriada
  - Usa interface real ou mock-up
  - wizard pode ser visível ou invisível
    - "pay no attention to the man behind the curtain!"

## Wizard of Oz

- Método para testar um sistema que não existe

Fonte: Saul Greenberg slides

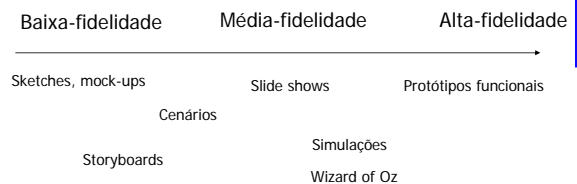
- the voice editor, IBM 1984



## Wizard of Oz

- Método:
  - Comportamento deve ser algorítmico
  - Bom para sistemas de reconhecimento de voz
  - Bom simular funcionalidade vertical complexa
- Vantagens:
  - Permite ao projetista 'mergulhar' na situação
  - Permite observar como as pessoas respondem, como especificam tarefas
  - Permite testar idéias futurísticas
- Ex. Ver  
[http://www.cs.uta.fi/research/hci/spi/reports/IPNMD\\_Makela\\_et\\_al.pdf](http://www.cs.uta.fi/research/hci/spi/reports/IPNMD_Makela_et_al.pdf)

## Prototipação



- Qual o seu plano de Prototipação?