IPM

**Design rules**

Design rules são usadas para aumentar a usabilidade de um produto de software. Quanto mais geral for uma regra de design, maior a possibilidade dela entrar em conflito com outras regras.

**Dimensões**

– **Authority**: indicação se uma regra deve ser seguida ou se é apenas uma sugestão.

– **Generality**: indicação se a regra pode ser aplicada a muitas situações de projeto ou se pode ser aplicada apenas em algumas situações particulares.

#### Regras de design

**Standards** (+ Authority, - Generality)

* Definido por instituições para garantir um conjunto de regras de design por uma comunidade (uniformidade, compatibilidade).
* Requer teoria subjacente consistente (estável).
* Padrões de hardware/software.

**Guidelines** (- Authority, + Generality)

* Sugestivo e mais geral
* Vários livros e relatórios técnicos contêm enormes catálogos de diretrizes.

#### Métricas da ISO 9241

* Usabilidade
* Eficácia
* Eficiência
* Satisfação

# Schneiderman’s 8 Golden Rules

1. Esforce-se para ter consistência
2. Busque usabilidade universal
3. Ofereça feedback informativo
4. Projete diálogos para encerrar
5. Evitar erros
6. Permitir fácil reversão de ações
7. Mantenha os usuários no controle
8. Reduzir a carga de memória de curto prazo

# Norman’s 7 Principles

1. Use tanto o conhecimento no mundo quanto o conhecimento na cabeça.
2. Simplifique a estrutura das tarefas.
3. Torne as coisas visíveis: elimine os abismos de Execução e Avaliação.
4. Obtenha os mapeamentos corretos.
5. Explore o poder das restrições, naturais e artificiais.
6. Projeto para o erro.
7. Quando tudo mais falhar, padronize

**HCI Design Patterns**

Uma abordagem para reutilizar o conhecimento sobre soluções de design bem-sucedidas

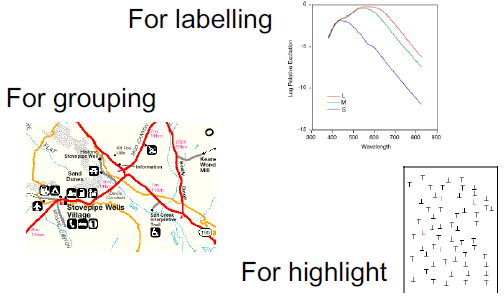
Um padrão é uma solução invariável para um problema recorrente dentro de um contexto específico.

**Exemplos:** Luz nos dois lados de cada sala (arquitetura)

### Informações visuais

A informação consiste em diferenças que fazem a diferença.

# Use of color



* Usar cores com moderação
* Usar cores consistentemente
* Evite o uso excessivo de cores saturadas e para campos pequenos, isso causa fadiga visual, porque o olho deve manter o foco em diferentes comprimentos de onda

## Interação de cores

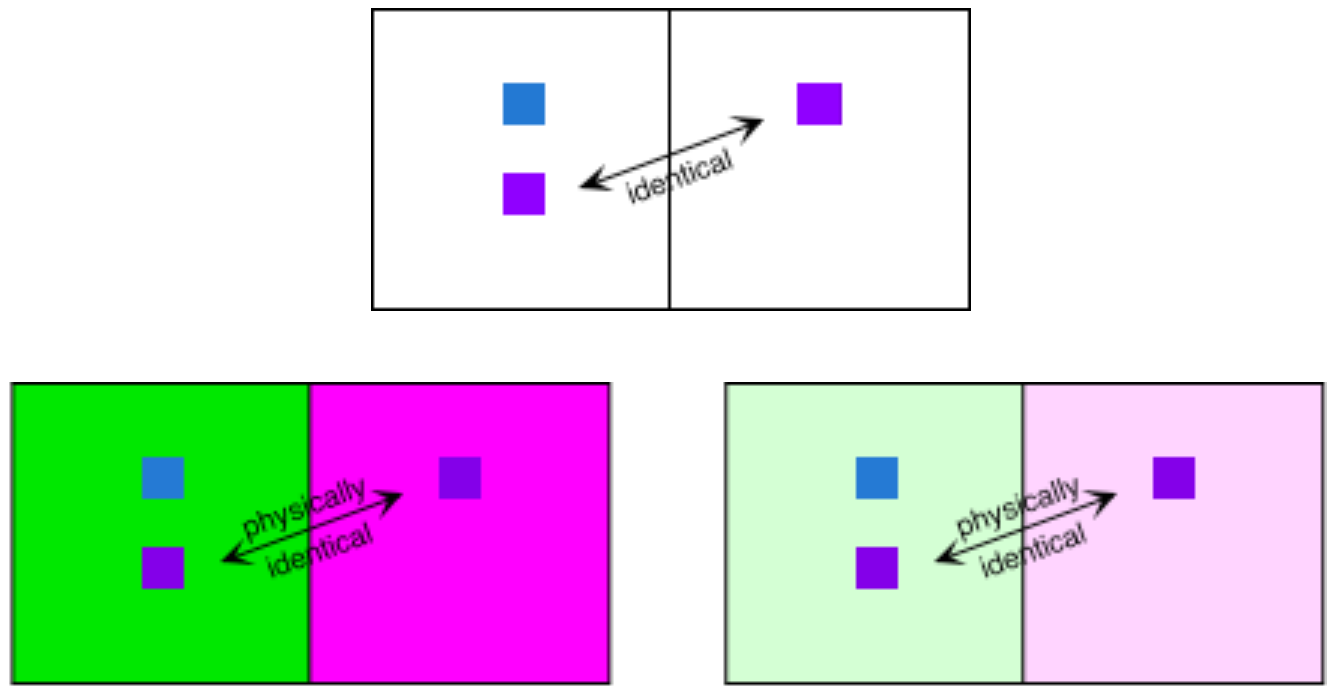
A cor aparente de um alvo é sempre afetada pela cor física das áreas adjacentes, no espaço e no tempo.

Os símbolos desenhados com luz de onda longa (vermelhos) aparecem com profundidade diferente dos símbolos de onda curta (azuis).

#### Contrastes simultâneos

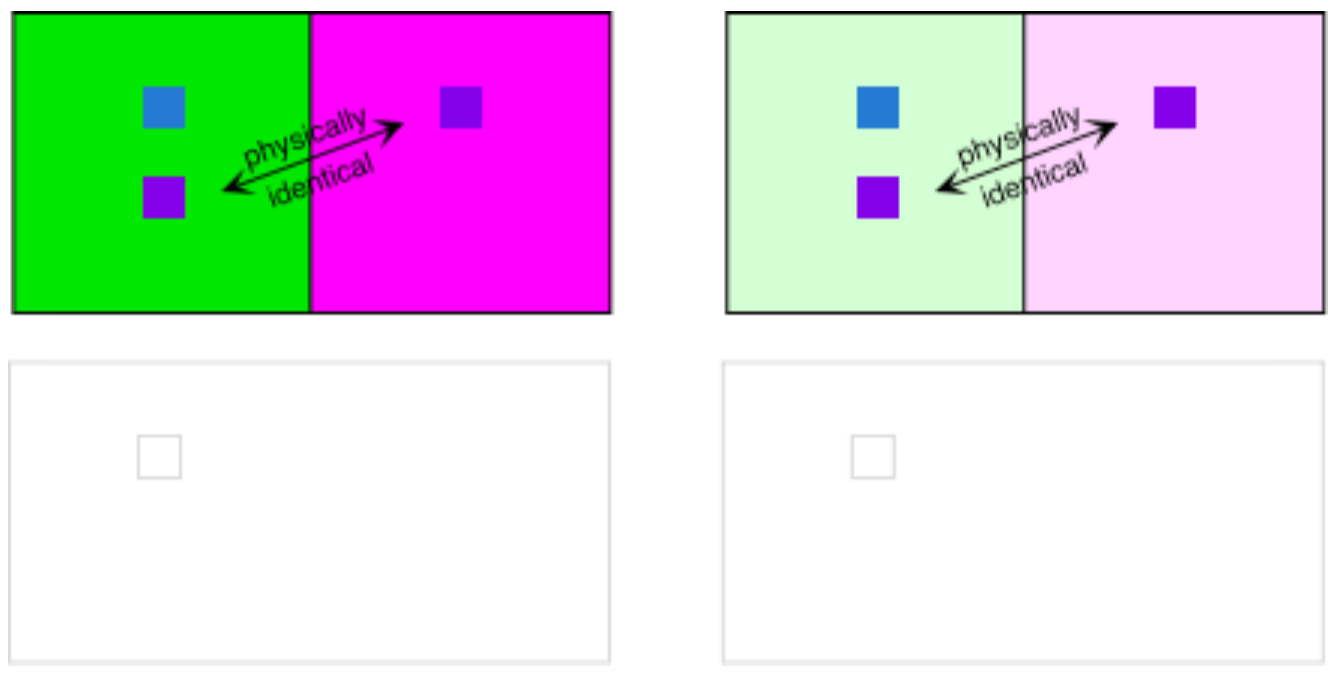
No painel superior, eles têm aproximadamente a mesma aparência.

No painel inferior esquerdo, no entanto, o par que é fisicamente idêntico tem aparências de cores diferentes devido aos efeitos de contraste simultâneos dos planos de fundo verde e magenta.



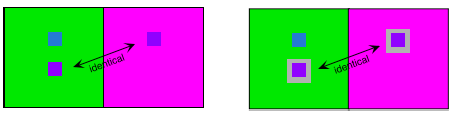
#### Contrastes sucessivos

O contraste sucessivo é o efeito dos campos de cores visualizados anteriormente ("campos indutores") na aparência do campo de teste visualizado no momento.



#### Proximidade espacial

Os efeitos dos campos indutores são maiores quando estão imediatamente adjacentes aos campos de teste e diminuem rapidamente com o aumento da separação espacial dos campos.



#### Proximidade temporal

Os efeitos dos campos indutores são maiores quando são visualizados imediatamente antes da visualização do campo de teste. A maior parte do efeito posterior geralmente desaparece em alguns segundos, mas na maioria dos aplicativos gráficos isso é longo o suficiente para ser um problema sério.

## Legibilidade

**Contraste de luminância:** diferença de brilho entre os símbolos e seus planos de fundo.

Um símbolo com a mesma luminância do fundo geralmente ficará ilegível

Grandes diferenças de matiz entre o texto e o plano de fundo não são suficientes para superar o contraste de luminância insuficiente.

* O azul puro não deve ser usado para detalhes finos ou fundo
* O amarelo difere do branco apenas na ausência do azul primário, então eles também fazem combinações pobres de símbolo/fundo.
* O azul pode ser usado na maioria dos contextos se for tomado cuidado para obter um contraste de luminância adequado
* Use um azul claro (adicionando luz amarela, aumentando a luminância, mas diminuindo a pureza) no preto ou contorne o texto

#### Discriminação e identificação de cores

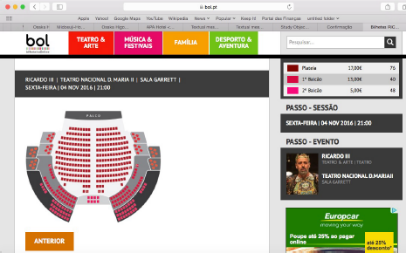
Pequenas diferenças de cor podem ser distinguidas quando as áreas a serem discriminadas são:

* amplas
* imediatamente adjacentes (compartilham uma aresta perto do ponto visualizado)
* exibidas ao mesmo tempo

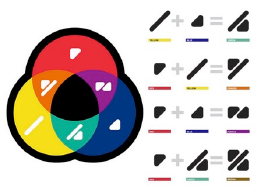
Ex: visualização de dados contínuos quantizados como em mapas de clima ou terreno.

* A discriminação de cores é melhor quando uma borda nítida separa as cores a serem discriminadas.
* Quando um gradiente suave separa duas cores, a diferença na aparência das cores é reduzida.

#### Guidelines

* Use no máximo seis cores para rotular elementos gráficos
* Use cores em conformidade com as convenções culturais
* Use o código de cores consistentemente nos monitores e
* Páginas
* Use o código de cores redundantemente com outros
* dimensões gráficas
* Não use codificação de cores em gráficos pequenos
* elementos
* Use contornos cinza neutros onde a cor
* julgamentos são críticos
* Evite cores saturadas

#### ColorADD

## Escolhendo a Cor

Use esquemas de cores

Escolha:

* 1 cor e vários tons de cinza
* 2 cores no máximo
* extraia cores de uma fotografia de cena natural

### Design gráfico

**Trade-offs** devem ser: identificados, avaliados e decididos com base nas melhores informações disponíveis.

#### Tipos de design

**Funcional**

Existem critérios especificos (gama de possiblidades) que podem ser explorados

**Estético**

Possibilidades não compatíveis com o padrão mínimo de usabilidade devem ser descartadas.

#### Guidelines

* Simplicidade (menos é mais)
* Contraste
* Espaço em branco
* Equilíbrio
* Alinhamento

# Simplicidade

**Redução**

* remover elementos não essenciais

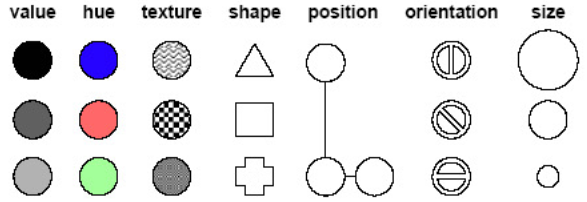
**Regularidade**

* Usar padrões regulares e os elementos que se devem de destacar não devem de se regularizar

**Combinando elementos para vantagem máxima**

* Os elementos devem de desempenhar várias funções, contudo não se pode exagerar

# Contraste



## Visual variables

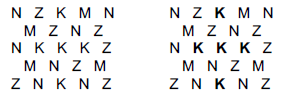
**Scale: tipos de comparações**

* + - Nominal (comparação de igualdade) 🡪 Todas as variáveis
    - Ordenado (> < comparação) 🡪 Posição, tamanho, valor, granularidade da textura
    - Quantitativo (compare a quantidade de diferença) 🡪 Posição e tamanho;

**Length: valor da dimensão**

* + - Formato: longo (variedade infinita)
    - Posição: longa (limitada pelo tamanho e resolução da tela) e refinada
    - Orientação: muito curta (+- 4 níveis)
    - Outras variáveis ​​(+- 10 níveis)

#### Interação entre variáveis ​​visuais

* **Tamanho** e **Valor** dominam a percepção. (Ex.: é difícil determinar a tonalidade de um ponto muito pequeno ou linha fina)
* **Forma** não é seletiva

# Técnicas de contraste

* Escolha variáveis ​​visuais apropriadas.
  + - Use o máximo de comprimento possível (mas minimize o número de valores distintos)
    - Focar nas distinções para uma percepção mais fácil
    - Teste de estrabismo

## Espaço em branco

* Use espaço em branco para agrupar elementos de interface.
* Use margens ao redor dos elementos da interface.
* Os objetos devem ser dimensionados proporcionalmente ao seu plano de fundo.
* Não aglomere os controles, a aglomeração cria tensão espacial e inibe a varredura

**Usado para:** separar; estrutura; realçar

## Equilíbrio e Simetria

* Escolha um eixo (geralmente vertical)
* Distribua os elementos igualmente em torno do eixo

#### Tipografia

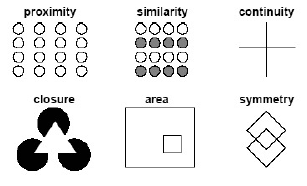
* fonte e espaçamento do texto na tela
  + - legiblidade (Métricas: velocidade, compreensão, taxa de erro, preferência subjetiva)
    - proporcional (largura entre caractere diferente) VS. monoespaçado (mesma largura entre caracteres)

#### Espaçamento

Sempre deixe margens ao redor do texto

## Alinhamento

#### Os princípios da Gestalt explicam o reconhecimento do grupo



### Técnicas de avaliação

Objetivos:

* Testar funcionalidades do sistema
* Ver impacto da interface no utilizador
* Identificar problemas específicos do sistema

# GOMS

Técnicas de modelação para análise da complexidade de sistemas interativos.

**O comportamento do usuário é modelado em termos de:**

* Objetivos: declarar o que o user quer alcançar
* Ações perceptivas, motoras ou cognitivas que devem ser executadas para atingir o objetivo
* Procedimentos que descrevem como atingir objetivos
* Regras de seleção são usadas para determinar o método que deve de ser usado quando há vários métodos possíveis para realizar algo

O modelo é usado para:

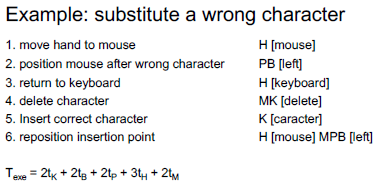
* Verificar as funcionalidades do sistema
* Visualizar tempos de execução
* Visualizações qualitativas sobre as tarefas em que os usuários cometem poucos erros

## Keystroke level model

Fornece visualizações numéricas sobre o desempenho do usuário. Compara o tempo necessário para concluir uma tarefa usando os diferentes métodos possíveis.

* + - K - pressionamento de tecla
    - P - apontando
    - H - homing
    - B - botão pressionando
    - D – linha de desenho (mouse)
    - M - preparação mental
    - R - resposta do sistema
    - Os tempos são determinados empiricamente (T=Tarefa).

O tempo de execução de uma tarefa é estimado através da sequência de operadores que compõem o método, somando os tempos associados a cada operador. Uma boa visualização – erro 20%



## Vantagens do GOMS

* Ajuda a encontrar problemas de usabilidade.
* Economiza tempo e recursos.
* Fácil de construir um modelo simples.

**Preditivo**: prever o tempo que um user levará para realizar as tarefas em análise

**Descritivo**: os métodos, sub-objetivos e regras de seleção fornecem ao projetista uma descrição do processo.

**Prescritivo**: serve de guia para o desenvolvimento de programas de treinamento e sistemas de ajuda.

## Limitações

* Apenas para expert users
* Não considera:
* diferenças entre users (média estatística do tempo de execução do operador)
* impacto social
* satisfação do usuário

### Heuristic Evaluation

Executado por um expert e baseado em análise e julgamento, e pode ser aplicado a Sketches, protótipos de papel e protótipos instáveis.

**Processo:**

1. O avaliador inspeciona a interface
2. Compara a interface com a heurística
3. Elaborar lista de problemas de usabilidade (explica e justifica cada problema de acordo com heurísticas)
4. Usa vários avaliadores (avaliadores **≠** encontram problemas **≠** 🡪 Cada novo avaliador encontra poucos novos problemas 🡪 3-5 avaliadores para melhor custo/benefício)

# Heurística de Usabilidade – J. Nielsen

**Visibilidade do status do Sistema:** Informação sobre o estado do sitema ao utilizador

**Corresponde ao mundo real:** Se o sistema usa o idioma correto

**Controle do usuário e liberdade:** “Saídas claramente marcadas”

**Consistência e Padrões:** Coisas semelhantes devem parecer e agir de maneira semelhante

**Prevenção de erros:** Não dar ao user a possiblidade de cometer erros

**Reconhecimento em vez de recordação:** informações necessárias devem estar visíveis

**Flexibilidade e eficiência de uso:** Atalhos para operações frequentes

**Design estético e minimalista:** Simplicidade e linguagem concisa

**Relatório de erros, diagnóstico e recuperação**: msg´s precisas|construtivas|educadas|pouco detalhe técnico

**Ajuda & Documentação:** users normalmente não leem guias exceto quando não têm outra escolha

## Processo formal

#### Treinamento

* + Reunião da equipe de design e avaliadores
  + Preparação de aplicação, usuários-alvo, cenários, ...

#### Avaliação

* + Os avaliadores trabalham separadamente
  + Produzir um **relatório** escrito ou comentários orais registrados pelo observador

#### Classificação de gravidade

* + - * Problemas identificados são compilados numa lista
      * Os avaliadores classificam cada um dos problemas
      * Calcule a média das notas do avaliador.

**Fatores:** Frequência (comum ou rara); Impacto (fácil ou difícil de superar); Persistência (com que frequência superar?)

**Escala:** 1.Cosmética: correção não obrigatória 🡪 2.Menor: correto, baixa prioridade 🡪 3.Maior: correto, alta prioridade 🡪 4.Catastrófico: a correção é essencial

#### Discussão dos resultados

* + - * Equipe de design e avaliadores
      * Brainstorm de soluções

## No relatório

* Seja educado
* Seja específico
* Inclua também comentários positivos

## Prós e contras da avaliação heuristica

**+** Mais barato

**+** Rápido

**+** Identifica muitos problemas: menor e maior

**-** Difícil identificar elementos ausentes em um esboço

**-** Difícil identificar problemas relacionados ao domínio do problema

# Princípios de Tog

1. **Estética:** estéticanunca deve superar a usabilidade.
   * 1. **Antecipação:** antecipar as necessidades do user
     2. **Autonomia:** dar o controle ao user.
     3. **Cor:** a cor como elemento vital da interface.
2. **Consistência:** consistência com as expectativas do user.
3. **Padrões:** fáceis de mudar, selecionado
4. **Descoberta:** se o user não encontra, não existe
5. **Eficiência do usuário:** concentre-se na produtividade do user, não no computador
6. **Interfaces exploráveis:** ações reversíveis
7. **Lei de Fitts:** botões grandes são mais rápidos
8. **Objetos de interface humana:** diretórios, arquivos, lixeira; consistente, estável, auto-significativo
9. **Redução de latência:** 
   * + Feedback visual e de áudio dos botões
     + Barra de progresso, sinal de finalização das operações (bipe)
     + Mensagens que indicam as ações do sistema
     + Trap de vários cliques do mesmo botão ou objeto
10. **Aprendível:** sem curva de aprendizgem
11. **Metáforas:** cria imagens na mente do usuário.
12. **Proteger o trabalho do usuário:** usuários nunca devem de perder o seu trabalho
13. **Legibilidade:** contraste; tamanho da fonte (deficiências visuais, idosos)
14. **Simplicidade:** evite a “ilusão de simplicidade”
15. **Estado da Faixa:** cookies
16. **Navegação visível**
    * + Tornar a navegação visível
      + Reduza as necessidades de navegação
      + Navegação clara e natural

**Cognitive Walkthrough**

**Foco na capacidade de aprendizagem:** os users preferem aprender enquanto exploram.

Os avaliadores executam uma sequência de ações para atingir um objetivo, buscando possíveis problemas de usabilidade:

1. Descrição do protótipo (não precisa ser completa, mas precisa ser detalhada).
2. Descrição da tarefa (tarefa frequente).
3. Lista completa das ações necessárias para concluir a tarefa com o protótipo fornecido.
4. Indicações sobre os usuários e sua experiência.

# Teste ao user

1. **Selecione os participantes:**
   * + - users representativos em termos de conhecimento do domínio
2. **Selecionar tarefas**
   * + - realistas
       - não fragmentadas
       - evitar tarefas longas

#### Estudo de campo: Observações qualitativas (usuários no ambiente real)

#### Experiências controladas: Observações quantitativas (tempo, nº de erros).

## Trate o usuário com respeito

**Os users podem apresentar:**

* + - * Ansiedade de desempenho
      * Parecer um teste de inteligência
      * Medo de falhar
      * Sentirem-se observados

**Eles precisam:**

* + - * Tempo
      * Conforto
      * Consentimento
      * Privacidade
      * Ao controle

**Após o teste:**

* + - * Conforto
      * Informação e consentimento
      * Privacidade

## Avaliação formativa

Observações **qualitativas** 🡪 usabilidade

1. Seleciona alguns users apropriados
2. Dê a cada usuário algumas tarefas
3. Observe os usuários fazerem as tarefas

#### Funções

* **Do utilizador**
* **Do Facilitador**
  + - * + faz o briefing
        + dá as tarefas
        + incentiva o usuário a pensar em voz alta (fazendo perguntas)
        + controla a sessão
* **Dos Observadores**
  + - * + fica quieto
        + fazer anotações

#### Observações de gravação

* + - * Notas de caneta e papel
      * Gravação de áudio
      * Gravação de vídeo
      * Captura de tela e registro

Observação: usa 5 usuários para encontrar 85% de problemas com cada uma das três iterações de design.

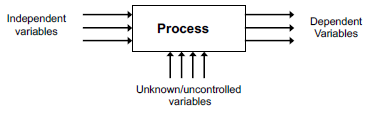
**Controlled experiments**

Os users devem ser representativos da população de users-alvo

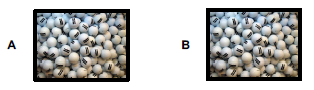
## Variáveis

**Independente** - manipulado para produzir condições diferentes para permitir a comparação de resultados

**Dependente** - afetado pelas variáveis ​​independentes. Seus valores resultantes são medidos



**Exemplo**



#### Validade interna

* + - * A contagem manual é confiável para alguns números de bolas
      * A contagem repetida melhora a confiabilidade, mas é lenta...

**Efeitos de ordenação (**a ordem das tarefas deve ser dada aleatoriamente ao usuário**)**

**Efeitos de seleção (**Atribua aleatoriamente usuários a grupos**)**

**Viés do experimentador (**experimentador pode preferir uma interface sobre a outra**)**

#### Validade externa

* + - * Pese as caixas em vez de contar as bolas
      * A bola A pode ter um peso diferente da bola B (variável dependente - peso total)
      * A caixa A pode ter um peso diferente da caixa B

**População (**amostra aleatória de usuários**)**

**Ecológico (**condições de laboratório o mais realista possível**)**

**Treinamento (**imitar como a interface real seria encontrada**)**

**Tarefa (**Basear tarefas na análise de tarefas**)**

Confiabilidade

* + - * Esse resultado se aplica a todas as caixas do mundo?

**Experiência anterior (**Novatos e especialistas: separados**)**

**Diferenças do usuário (**Inteligência, acuidade visual, memória**)**

**Desenho de tarefas;**

**Erros de medição (**O tempo na tarefa inclui treinamento, distrações**)**

**Soluções (**Dê a todos os usuários o mesmo treinamento**)**

Observação: Elimine a variação descontrolada e, portanto, aumente a confiabilidade

# Métodos

“Between-subjects” design

* Cada grupo de users testa 1 interface
* Resultados comparados entre grupos diferentes (elimina o efeito de ordenação)

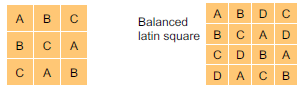
“Within-subjects” design

* + - Cada user testa todas as interfaces (em ordem aleatória)
    - Resultados comparados dentro de cada usuário (Elimina a variação devido às diferenças do usuário; Efeito de pedido; Efeito de fadiga)

**Counterbalancing**

Reduza os efeitos de ordenação variando sistematicamente a ordem das condições

**Desenho quadrado latino**



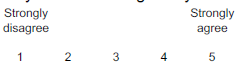
# Questionários

* Perguntas pré-definidas – menos flexíveis e mais rápidas do que entrevistas.
* Pode ser usado em várias fases do processo de design.
* Simples e barato de executar, fornece informações subjetivas.

#### Tipos de perguntas:

* Em geral
* aberto
* Escalar
* Múltipla escolha
* classificado

**Likert scales**



#### Evaluation

#### Estudos laboratoriais

* permitem experimentação controlada e observação
* perde a naturalidade do ambiente do usuário

#### Campo de estudos

* não permite o controle sobre a atividade do usuário
* Ambos os estudos devem ser feitos:
  + Estudos de laboratório dominando os estágios iniciais
  + Estudos de campo para novas implementações

## Measurements

**Quantitativas**

* numérico
* podem ser facilmente analisados ​​usando técnicas estatísticas

**Qualitativas**

* não numérico
* difícil de analisar
* fornecem detalhes importantes que não podem ser determinados a partir de números.

Escalas numéricas podem ser usadas para coletar dados subjetivos – escalas Likert.