

Engenharia de Software III

Aula 13

Projeto de Interface de Usuário Análise de Tarefas, Prototipação e Avaliação de Interfaces

l.bertholdo@ifsp.edu.br

Conteúdo

- Análise de Tarefas
- Prototipação
- Avaliação de Interfaces
- Métodos de Avaliação

Análise de Tarefas



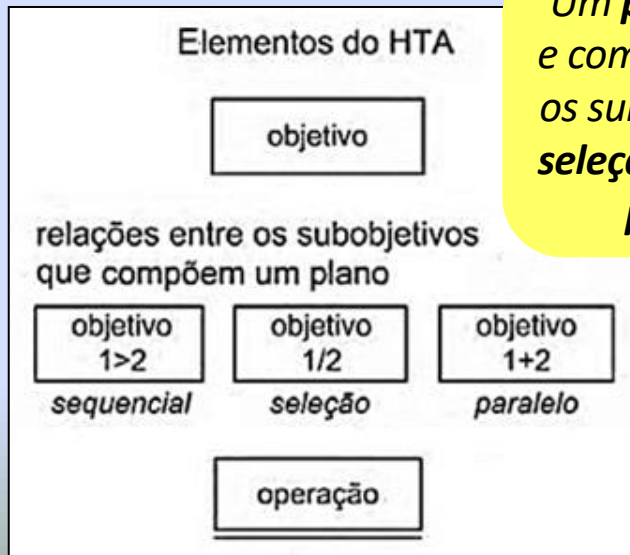
- A **análise de tarefas** visa obter ter um entendimento sobre qual é o trabalho dos usuários, como e por que eles o realizam.
- Um dos primeiros passos numa análise de tarefas é coletarmos um conjunto de **objetivos** das pessoas. Para cada objetivo, elaboramos uma **lista das ações** a serem realizadas para alcançar esse objetivo.

*Vale ressaltar que só é possível **simular** as tarefas de interesse, pois: (1) existem inúmeras tarefas e apenas algumas delas podem ser selecionadas para a análise; (2) apenas uma porção do trabalho de cada tarefa pode ser observada; (3) a coleta dos dados de uma tarefa pode não refletir fielmente sua realidade.*

- Dentre os métodos de análise de tarefas mais comuns estão a Análise Hierárquica de Tarefas (**HTA** – *Hierarchical Task Analysis*), o **GOMS** (*Goals, Operators, Methods and Selection Rules*) e a Árvore de Tarefas Concorrentes (**CTT** – *Concur Task Trees*).

Análise Hierárquica de Tarefas (HTA)

- A **HTA** busca entender as competências e habilidades empregadas em tarefas complexas e não repetitivas, bem como auxiliar na identificação de problemas de desempenho.
- Nesse método, primeiramente, são levantados os **objetivos maiores** (por exemplo, marcar uma reunião), decompondo-os em **subobjetivos** (por exemplo, decidir a data e local, convidar os participantes etc.).

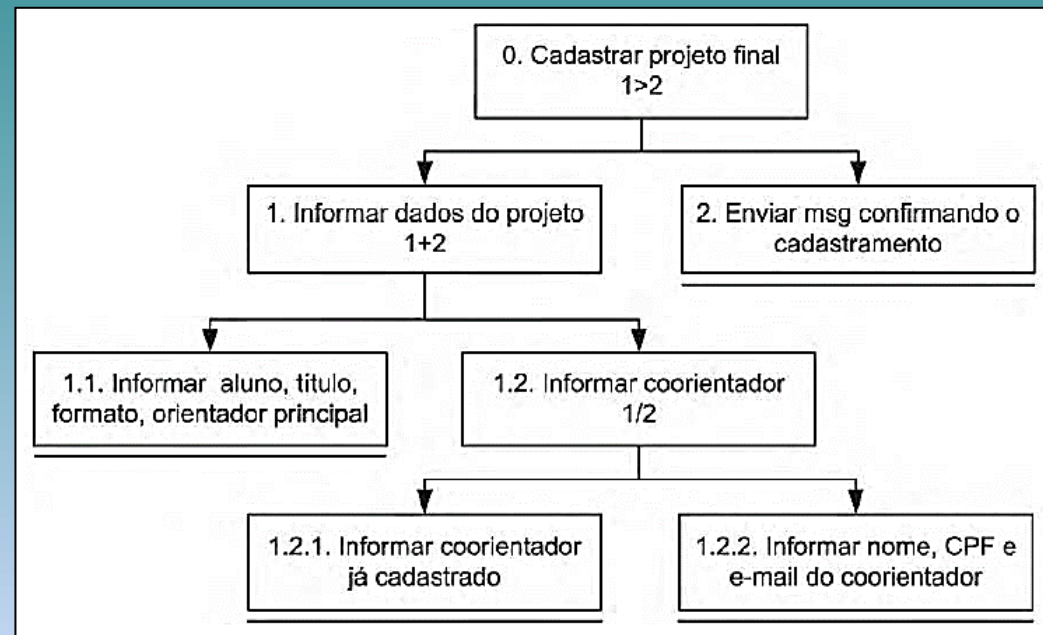


Um **plano** define os **subobjetivos** necessários para alcançar um **objetivo maior**, e como esses subobjetivos devem ser atingidos. Também define as relações entre os subobjetivos: **sequencial** (um subobjetivo deve ser atingido antes de outro); **de seleção** (os subobjetivos a serem atingidos dependem das circunstâncias); ou **em paralelo** (mais de um subobjetivo deve ser atingido ao mesmo tempo).

No nível mais baixo da hierarquia de objetivos, cada subobjetivo é atingido por meio de uma **operação**, a qual é especificada pelo conjunto **<input, ação, feedback>**. Por exemplo, em um sistema de agenda: **<[data, local, participantes], [convidar os participantes], [presença dos participantes confirmada]>**.

Análise Hierárquica de Tarefas (HTA)

- Diagrama HTA para o objetivo de **cadastrar um projeto final** em um sistema acadêmico:



Passos da HTA:

- 1. Decidir os objetivos da análise** que, em geral, envolve projetar um sistema novo ou modificar o sistema atual.
- 2. Obter consenso entre os envolvidos sobre os objetivos da tarefa e medidas de sucesso**, por meio da definição dos resultados de desempenho desejados e as formas como eles serão aferidos. As questões-chave são: “qual evidência indicará que o objetivo foi atingido?” e “quais as consequências da falha em atingir o objetivo?”.
- 3. Identificar as fontes de informações sobre as tarefas**, não apenas por meio da observação direta, mas também através de entrevistas com especialistas, registros de desempenho e relatos de incidentes nas tarefas, que costumam trazer informações preciosas para a HTA.
- 4. Coletar dados sobre as tarefas e esboçar um diagrama ou tabela de decomposição**, no qual os subobjetivos representem completamente o objetivo a que estão subordinados.
- 5. Verificar a validade da decomposição com as partes interessadas**, para assegurar a confiabilidade da análise.
- 6. Identificar operações significativas dos subobjetivos**, levando em conta o objetivo da tarefa.
- 7. Gerar e, se possível, testar hipóteses** relacionadas aos fatores que afetam o aprendizado e o desempenho.

GOMS (Goals, Operators, Methods and Selection Rules)

- GOMS é um conjunto de métodos para analisar o desempenho de usuários, realizando tarefas **dentro da sua competência**. Para isso, descreve uma tarefa e o conhecimento do usuário sobre como realizá-la em termos de objetivos, operações, métodos e regras de seleção.

*Os **objetivos** representam o que o usuário quer realizar usando o software (ex.: editar um texto). Os **operações** são as ações que o software permite que os usuários façam (ex.: clicar em um botão). Os **métodos** são sequências de subobjetivos e operações que permitem atingir um objetivo maior. Quando há mais de um método para atingir um mesmo objetivo, são usadas **regras de seleção**, que representam tomadas de decisão dos usuários sobre qual método utilizar numa determinada situação.*

- A análise GOMS se aplica principalmente a tarefas que os usuários já dominam, ou seja, em que eles já sabem o que deve ser feito. Dentre os modelos da família GOMS, destacam-se: o CMN (*Card, Moran e Newell*), o CPM (*Critical Path Method*) e o KLM (*Keystroke Level Model*).

GOMS-CMN (Card, Moran e Newell)

- Proposta original do método GOMS, elaborada por Card, Moran e Newell. Nesse método, há uma **hierarquia de objetivos**, em que as operações são executadas em **ordem sequencial**, e os métodos são representados como um **pseudocódigo** que inclui **submétodos** e **condições**.

GOAL 0: descobrir direção de tráfego de uma rua

GOAL 1: encontrar a rua

METHOD 1.A: zoom até o nível de ruas

(SEL. RULE: a região em que se situa a rua está visível no mapa e o usuário conhece o local)

METHOD 1.B: fazer busca pelo nome da rua

(SEL. RULE: o usuário não conhece o local ou o mapa visível está longe de lá)

GOAL 2: identificar a direção do tráfego na rua

*Modelo CMN simplificado representando as operações que envolvem descobrir a direção de tráfego de uma rua no Google Maps. Os **objetivos** e **métodos** são numerados para facilitar sua identificação. Algarismos indicam **sequência**, e letras indicam **alternativas**.*

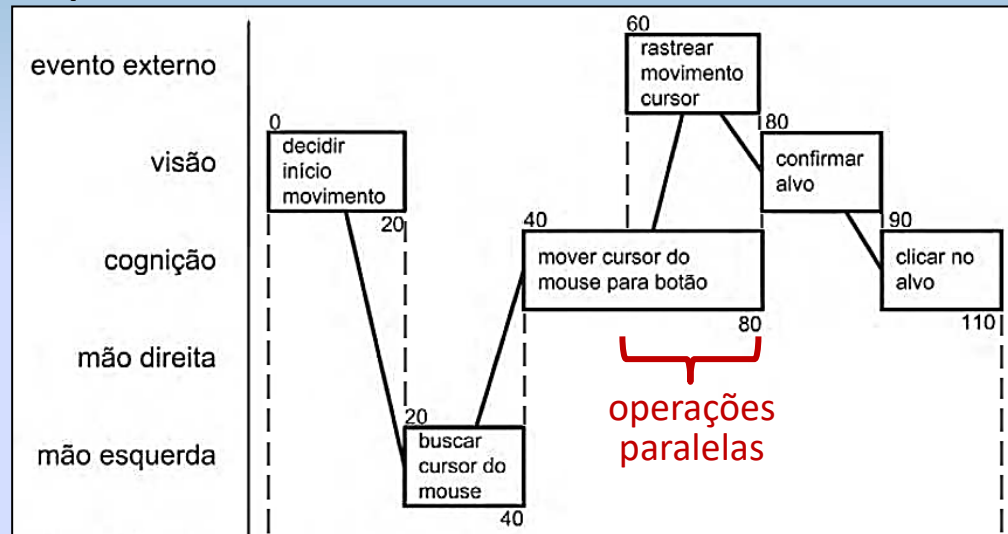


*Quantitativamente, o modelo CMN permite prever a **sequência de operações** e o **tempo de execução**. Qualitativamente, esse modelo permite uma **visualização clara dos métodos** para alcançar os **objetivos**: métodos semelhantes são facilmente identificados, métodos muito curtos ou longos se destacam e podem originar ideias de design, como, por exemplo, a inclusão de teclas de atalho para comandos curtos e frequentes.*

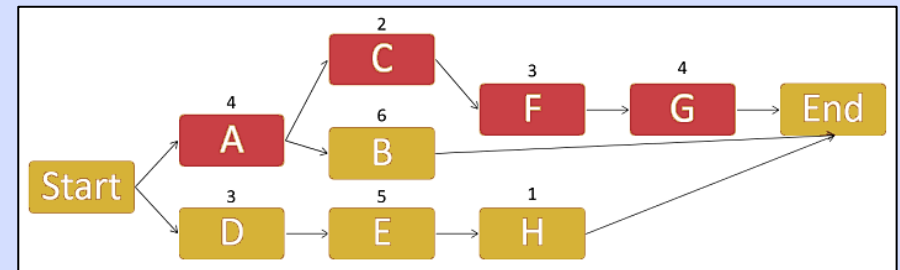
GOMS-CPM (Critical Path Method)

- Abordagem GOMS de análise do caminho crítico, a qual pressupõe que as operações cognitivas, perceptivas e motoras não são executadas sequencialmente, podendo se tornar paralelas dependendo da tarefa.

Clique em um botão:



O CPM utiliza um **diagrama tipo PERT** para representar as operações e as dependências entre elas. Nessa análise, o **caminho crítico** (maior caminho em termos de tempo) fornece uma previsão simples do tempo total da tarefa.



A construção de um modelo CPM inicia com a construção do modelo CMN. A partir do modelo CMN, atribuímos uma duração estimada a cada operação e calculamos o tempo de execução previsto para a tarefa. É possível ainda efetuar uma **análise qualitativa** da relação entre **aspectos do design** e o **tempo de execução**. Com base nisso, é possível **simular designs alternativos** e identificar por que um apresenta desempenho melhor do que o outro.

GOMS-KLM (Keystroke Level Model)

- Essa técnica é a mais simples dos métodos GOMS e contempla um conjunto predefinido de operações:

Algumas operações do KLM-GOMS e suas durações médias

operação	duração média
K: pressionar e soltar uma tecla do teclado	
exímio digitador (135 ppm)	0,08 s
bom digitador (90 ppm)	0,12 s
digitador mediano (55 ppm)	0,20 s
digitador inexperiente (40 ppm)	0,28 s
P: apontar o cursor do mouse num objeto da tela	1,10 s
B: pressionar ou soltar o botão do mouse	0,10 s
H: levar a mão do teclado ao mouse ou vice-versa	0,40 s
M: preparação mental	1,20 s

*Essa análise evidencia que usar as **teclas de atalho** é quase duas vezes mais eficiente do que usar o **botão correspondente na barra de ferramentas**, e quase três vezes mais eficiente do que usar o **item de menu correspondente**.*

Análise da Tarefa: Salvar arquivo

método	operação	descrição	tempo (em s)
menu Arquivo > Salvar	M	preparação	1,20
	H	levar a mão do teclado ao mouse	0,40
	P	levar cursor até menu Arquivo	1,10
	B	pressionar o botão do mouse	0,20
	B	soltar o botão do mouse	0,20
	P	levar cursor até menu Salvar	1,10
	B	pressionar o botão do mouse	0,20
	B	soltar o botão do mouse	0,20
		TOTAL	4,60
botão Salvar na barra de ferramentas	M	preparação	1,20
	H	levar a mão do teclado ao mouse	0,40
	P	levar cursor até botão Salvar	1,10
	B	pressionar o botão do mouse	0,20
	B	soltar o botão do mouse	0,20
		TOTAL	3,10
teclas de atalho (Ctrl+S), considerando um digitador mediano	M	preparação	1,20
	K	teclar Ctrl	0,20
	K	teclar S	0,20
		TOTAL	1,60

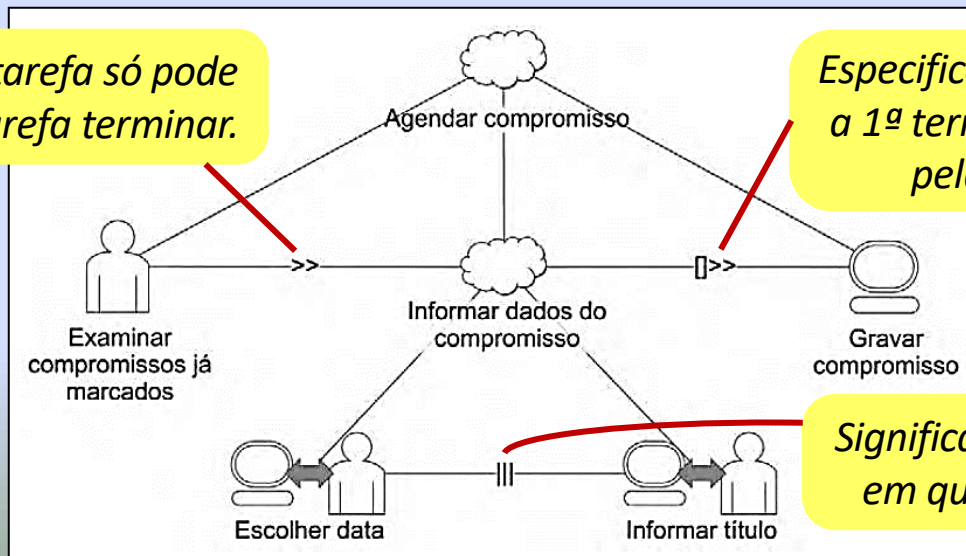
Árvores de Tarefas Concorrentes (CTT)

- Modelo que permite registrar, por meio de suas notações, as diversas relações entre as tarefas. Para isso, define quatro tipos de tarefas:



- Tarefas do usuário**, realizadas fora do sistema;
- Tarefas do sistema**, em que o sistema realiza um processamento sem interagir com o usuário;
- Tarefas interativas**, em que ocorrem os diálogos usuário-sistema;
- Tarefas abstratas**, que não são tarefas em si, mas sim uma representação de uma composição de tarefas.

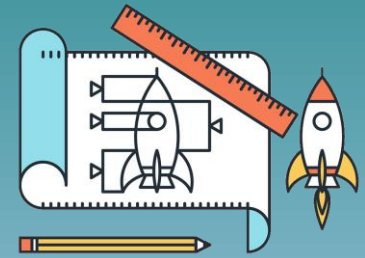
Significa que a 2ª tarefa só pode iniciar após a 1ª tarefa terminar.



Especifica que a 2ª tarefa só pode iniciar após a 1ª terminar, e que a informação produzida pela 1ª tarefa é passada para a 2ª.

Significa que as tarefas podem ser realizadas em qualquer ordem ou ao mesmo tempo.

Prototipação



- Uma interface pode ser representada através de simples esboços, de modelos estruturados ou até mesmo através de protótipos funcionais. E seu design pode ser realizado em diferentes **níveis de abstração**:

*Na elaboração da **interface abstrata**, definimos os agrupamentos e as características dos elementos de interface, como, por exemplo, um grupo com três campos de texto editável e um campo de seleção simples com dez opções. Na elaboração de **interface concreta**, definimos o posicionamento e escolhemos os componentes de interface interativos (widgets), por exemplo, textboxes, dropdowns, listboxes, radiobuttons etc.*

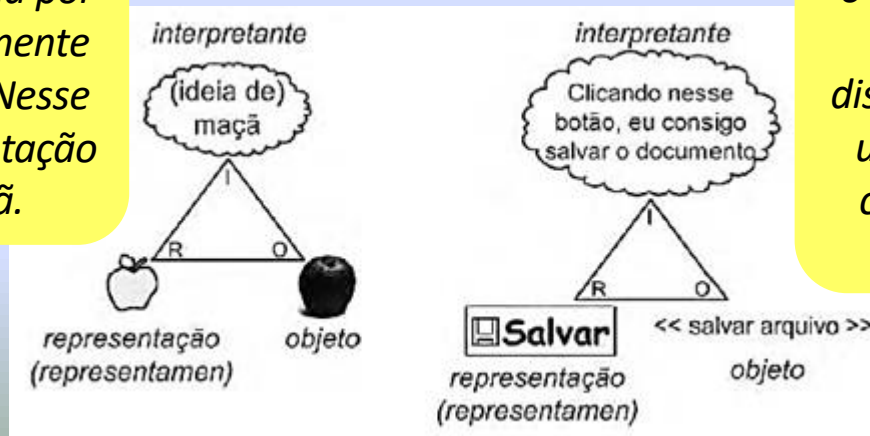
- Embora existam linguagens de descrição de interfaces (**UIDL – User Interface Description Languages**), em uma abordagem informal, o design da interface costuma ser representado em *wireframes* e protótipos, que são refinados sucessivamente ao longo do processo.

Prototipação

- Uma classificação comum diz respeito ao **grau de funcionalidade** embutida nos protótipos: desde *wireframes*, que apresentam apenas os **signos estáticos e metalinguísticos** da interface, até protótipos funcionais, que incluem também os **signos dinâmicos**.

No contexto da **Semiótica**, um **signo** é uma representação que comunica uma ideia, chamada “interpretante”, a qual remete a algum objeto real. Desse modo, para ser um signo, uma **representação** deve possuir uma relação triádica com seu **objeto** e com o seu **interpretante**.

A maçã pode ser representada por uma ilustração e evocar na mente de alguém a ideia de maçã. Nesse caso, dizemos que a representação é um **signo** da fruta maçã.

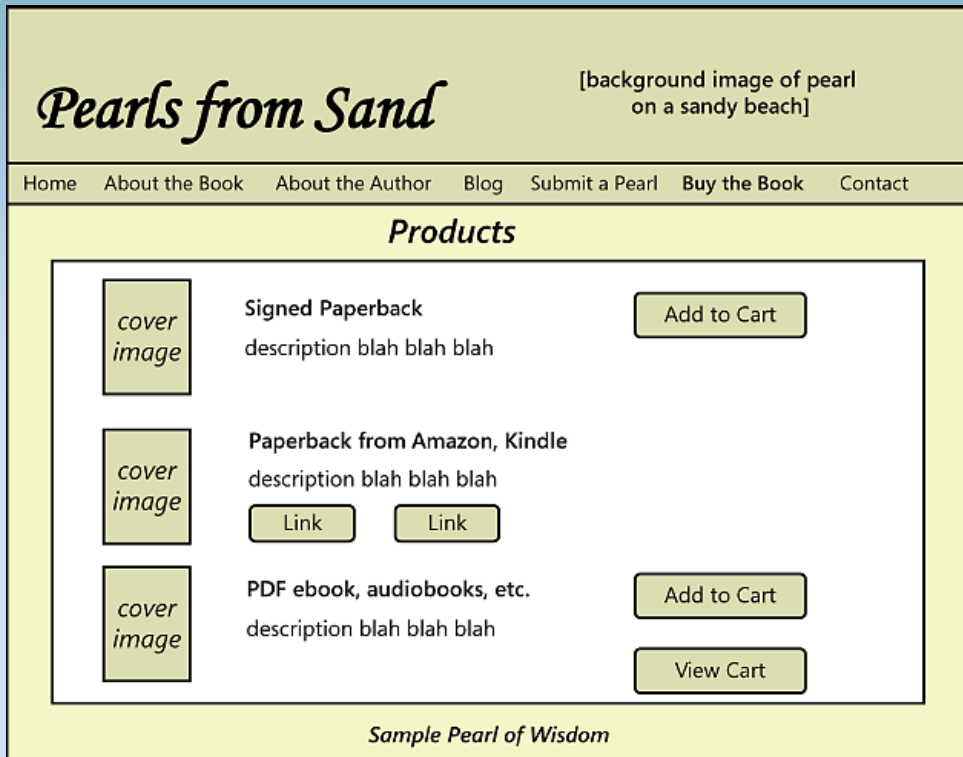


Ao representar a operação “salvar o documento” por um botão com o rótulo Salvar e um ícone de disquete, o designer espera que os usuários interpretem esse signo como “Clique nesse botão para salvar o documento”.

Prototipação

- Os protótipos também são classificados conforme seu **grau de fidelidade**:

Protótipo de Baixa Fidelidade (*wireframe*)



Nos protótipos de **baixa fidelidade**, não existe muita preocupação com detalhes gráficos. Os protótipos podem ser feitos por meio de um editor de imagens simples, ou até mesmo manualmente.

Protótipo de Alta Fidelidade



Nos protótipos de **alta fidelidade**, feitos em ferramentas apropriadas, o desenho da interface é completo e já incorpora decisões sobre posições, cores, fontes e outros detalhes visuais dos elementos.

Prototipação

- Vale ressaltar que a identificação dos principais **objetivos** e **cenários de uso**, realizada ao longo de todo o processo de design, fornece insumos para as decisões sobre o design da interface. Por exemplo:

- *Quais elementos de interface devem ser destacados (ex.: atalhos na barra de ferramentas, itens de menu, posição dos elementos de destaque na tela).*
- *Quais elementos de interface podem ser deslocados para posições secundárias (ex: link de retorno) ou mesmo para telas secundárias (ex.: opções de configuração).*



- Além disso, atividades de análise, pesquisas com usuários e outras fontes de informação auxiliam na definição do **vocabulário** a ser utilizado na interface.

É importante manter um vocabulário consistente e familiar ao usuário a fim de tornar a comunicação com ele eficiente e clara.

Avaliação de Interfaces



- A avaliação de IHC permite fazer um julgamento sobre a **qualidade de uso** da solução e a identificar **problemas na interface e na interação** que prejudiquem a experiência do usuário durante o uso.
- Sempre que possível, a avaliação de IHC deve ser conduzida por pessoas que não participaram da concepção da solução, pois elas possuem melhores condições de analisá-la sob um ponto de vista que prioriza os **usuários**, e não o **design** concebido.

*Avaliar a **qualidade de uso** não representa apenas um aumento do custo de desenvolvimento, como alguns gerentes de projeto costumam pensar. A **curto prazo**, essa atividade contribui para aumentar a **produtividade** dos usuários, diminuir o **número e a gravidade dos erros** cometidos durante o uso, e aumentar a **satisfação** dos usuários. A **médio e longo prazo**, contribui para diminuir os **custos de treinamento e suporte** e para **planejar versões futuras** do sistema, a partir do conhecimento das partes do sistema que podem melhorar.*

Avaliação de Interfaces

- É possível avaliar diversos **aspectos** relacionados ao uso, de acordo com os interesses dos *stakeholders*. Os principais aspectos avaliados são:

Apropriação de tecnologia pelos usuários:

Permite avaliar, em diferentes momentos do processo de design, o contexto em que o sistema se insere, quais os objetivos dos usuários, como eles costumam alcançá-los, em que grau as tecnologias satisfazem suas necessidades e como elas afetam sua vida pessoal e profissional.

Ideias e alternativas de design:

Permite comparar diferentes alternativas de solução de acordo com critérios de uso e de construção da interface. Por exemplo, com relação ao uso, podemos avaliar o apoio à recuperação de erros e, com relação à construção, o custo e o tempo para o desenvolvimento de cada alternativa.

Conformidade com um padrão: *Avalia a solução em relação a padrões estabelecidos. Por exemplo, podemos avaliar se os padrões determinados pelos sistemas operacionais Microsoft Windows e MacOS foram seguidos, de modo que os usuários acostumados com esses ambientes tenham menos dificuldades para realizar suas operações.*

Problemas na interação e na interface: *São os aspectos mais avaliados na área de IHC. Os problemas identificados costumam ser classificados de acordo com sua gravidade, com sua frequência e com os critérios de qualidade de uso prejudicados (usabilidade, experiência do usuário, acessibilidade ou comunicabilidade).*

Avaliação de Interfaces



- Os objetivos de uma avaliação de IHC precisam ser detalhados em questões mais específicas para torná-los atingíveis. Alguns exemplos de perguntas associadas aos aspectos de avaliação de IHC:

Aspectos	Exemplos de perguntas a serem respondidas
Analisar a apropriação da tecnologia	<ul style="list-style-type: none">De que maneira os usuários utilizam o sistema? Em que difere do planejado?Como o sistema interativo afeta o modo de as pessoas se comunicarem e relacionarem?Que variação houve no número de erros cometidos pelos usuários ao utilizarem o novo sistema?O quanto os usuários consideram o apoio computacional adequado para auxiliá-los em suas atividades?O quanto eles são motivados a explorar novas funcionalidades?A tecnologia disponível pode oferecer formas melhores de os usuários atingirem seus objetivos?
Comparar ideias e alternativas de design	<ul style="list-style-type: none">Qual das alternativas é a mais eficiente? Mais fácil de aprender?Qual delas pode ser construída em menos tempo?De qual delas se espera que tenha um impacto negativo menor ao ser adotada?Qual delas torna mais evidente os diferenciais da solução projetada?Qual delas os usuários preferem? Por quê?
Verificar a conformidade com um padrão	<ul style="list-style-type: none">O sistema está de acordo com os padrões de acessibilidade do W3C (<i>World Wide Web Consortium</i>)?A interface segue o padrão do sistema operacional? E da empresa?Os termos e expressões exibidos na interface seguem as convenções estabelecidas no domínio?
Identificar problemas na interação e interface	<p>Considerando cada perfil de usuário:</p> <ul style="list-style-type: none">O usuário consegue atingir seu objetivo? Em quanto tempo? Após cometer quantos erros?Que parte da interface e da interação o deixa insatisfeito?Ele entende o que significa e para que serve cada elemento de interface?Quais barreiras o usuário encontra para atingir seus objetivos?Ele tem as informações necessárias para usar o sistema?

Métodos de Avaliação

- Existem vários métodos para avaliar a qualidade de uso. Cada um atende melhor a certos objetivos de avaliação, orienta quando e onde os dados devem ser coletados, como eles devem ser analisados, e quais critérios de qualidade de uso sua análise privilegia.
- Os **métodos de investigação** usam questionários, entrevistas e grupos focais, para interpretar e analisar concepções, opiniões, expectativas e comportamentos do usuário relacionados com sistemas interativos.



*Esses métodos são utilizados em **etapas iniciais** do processo de design, para ratificar ou retificar o entendimento da situação atual e identificar necessidades e oportunidades de intervenção. Em geral, os métodos de investigação não exigem que os usuários utilizem um sistema durante a coleta de dados. Porém, o uso de um sistema ou de materiais de apoio, como imagens, cenários e outros tipos de artefatos, pode contribuir para a investigação.*

Métodos de Avaliação



- Os **métodos de inspeção** permitem ao avaliador antever problemas que os usuários podem vir a ter, comparar designs alternativos e avaliar a conformidade do design com um padrão. Alguns exemplos são: a avaliação heurística, o percurso cognitivo e a inspeção semiótica.
- Esses métodos geralmente não envolvem diretamente usuários, pois tratam de experiências de uso potenciais, e não reais. Ao inspecionar a interface, os avaliadores tentam se colocar no lugar de um usuário com um certo conhecimento e experiência em atividades específicas.

*Métodos de inspeção costumam ser mais **rápidos** e **baratos** do que métodos que envolvem usuários, porém existe um limite na **empatia** do avaliador. Ele pode não encontrar problemas que os usuários teriam, pode julgar como problemáticos pontos que não causariam dificuldades aos usuários, pode se concentrar mais em alguns aspectos de usabilidade do que em outros. Por isso, sempre que possível, devemos usar diferentes métodos para fazer uma avaliação mais precisa.*

Métodos de Avaliação

- Os **métodos de observação** fornecem dados sobre situações em que os usuários realizam suas atividades, com ou sem apoio de sistemas interativos. Alguns exemplos são: o teste de usabilidade, o método de avaliação de comunicabilidade e a simulação via prototipação em papel.
- Através do registro dos dados observados, esses métodos permitem identificar **problemas reais** que os usuários enfrentam, e não apenas **problemas potenciais** como em uma avaliação por inspeção.



*O avaliador pode observar os usuários em **contexto** ou em **laboratório**. A **observação em contexto** permite coletar uma gama mais ampla de dados mais ricos sobre a atuação dos usuários em seu ambiente real de atividade. Já a **observação em laboratório** costuma ser mais direcionada e simples, pois a análise é realizada em um ambiente controlado pelo avaliador.*

Métodos de Avaliação

- Os **objetivos (ou aspectos)** a serem considerados na avaliação servem de diretrizes para o avaliador escolher os **métodos de avaliação**:

- *Se o objetivo for **identificar problemas de IHC**, pode ser mais adequado empregar um método de inspeção para cobrir quase toda a interface, e selecionar um poucas partes a serem avaliadas pelos métodos de investigação ou observação (geralmente, as partes cuja inspeção não forneceu resultados suficientemente confiáveis).*
- *Já avaliar a **apropriação da tecnologia pelos usuários** requer o emprego de um método de investigação ou de observação, por contar com a participação dos usuários.*
- *Para avaliar a **conformidade com um padrão**, pode ser mais apropriado empregar um método de inspeção, pois a participação dos usuários é desnecessária.*

Além disso, o avaliador precisa escolher métodos de avaliação considerando prazo, orçamento, equipamentos, número de usuários disponíveis, número de avaliadores capacitados e experientes em cada método e demais recursos necessários.

Referências

- BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana da. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.
- WIEGERS, Karl; BEATTY Joy. **Software Requirements**: Third Edition. 3. ed. Redmond, United States: Microsoft Press, 2013.