

Interação Humano-Computador

André Roberto Guerra

Aula 2

CONVERSA INICIAL

Olá, caro aluno!

Esta será nossa segunda aula de **Interação Humano-Computador!**

Nesta aula pretendemos proporcionar a você uma visão geral do processo de *Design* de Sistemas Interativos e introduzir uma abordagem conhecida como **design baseado em cenário**, que se baseia e formaliza a percepção obtida a partir da análise PACT.

Está preparado?

Acesse o material *on-line* para conhecer mais detalhadamente os tópicos a serem discutidos neste encontro!

CONTEXTUALIZANDO

Há muitas maneiras de caracterizar o processo de *Design*. O *Design* é um processo criativo que se preocupa em produzir algo novo. Ele consiste em uma atividade social com consequências sociais, trata da mudança consciente e da comunicação entre os *designers* e as pessoas que usarão o sistema.

As disciplinas de *Design* têm métodos e técnicas distintas para ajudar nesse processo. Abordagens e filosofias do *Design* mudam com o tempo. Em disciplinas maduras, são construídos exemplos de bom *design* que as pessoas podem estudar e, a partir daí, refletir sobre o que torna determinado *design* ótimo, bom ou péssimo. Diferentes disciplinas do *Design* têm restrições diferentes, como se o objeto projetado é “independente” ou se ele tem de se encaixar e conviver com sistemas ou conformar-se a padrões preexistentes.

Vamos ver o que mais o professor André Guerra tem a dizer sobre o processo de *design*? Acesse o material *on-line*!

PESQUISE

Processos implicados no *Design*

A abordagem que o *Design* utiliza consiste em quatro atividades:

- Entendimento
- *Design*
- Antecipação
- Avaliação

O *design* pode começar com qualquer uma das atividades, mas a avaliação é (crucialmente) fundamental, não pode ser ignorada!

Avaliação

Há muitas maneiras diferentes de caracterizar as atividades que participam do processo de *design*. A **avaliação** é fundamental para o *Design* de Sistemas Interativos e tudo é avaliado em cada etapa do processo. O processo pode começar em qualquer ponto – às vezes existe um *design* conceitual, às vezes começamos com um protótipo, às vezes começamos com os requisitos e as atividades podem acontecer em qualquer ordem.

Por exemplo, os requisitos podem ser avaliados e um protótipo talvez seja construído e avaliado, o que pode resultar na identificação de um *design* físico.

Os *designers* devem pesquisar a gama de pessoas, atividades e contextos relevantes para o domínio que estão investigando, de modo que possam entender os requisitos do sistema que estão desenvolvendo. É necessário, ainda, que os *designers* entendam as oportunidades fornecidas pelas tecnologias e suas restrições!

Entendimento

O entendimento trata do que o sistema deve fazer, como deve ser e como se “encaixar” com outras coisas; com os requisitos do produto, sistema ou serviço.

Uma análise PACT (**P**essoas, **A**tividades, **C**ontextos e **T**ecnologias) que será descrita em tópico específico nas aulas seguintes, complementa o processo. Observe que as tecnologias impõem restrições às soluções, elas limitam o que é possível, diferenciando os requisitos funcionais e não funcionais e desenvolvendo as ideias dos *stakeholders*.

Stakeholder, em inglês *stake* significa interesse, participação, risco. *Holder* significa aquele que possui. Assim, *stakeholder* significa **parte interessada**.

Design

As atividades de *Design* tratam tanto do **design conceitual** quanto do **design físico**.

O **design conceitual** trata da concepção de um sistema em termos abstratos e considera quais informações e funções são necessárias para que o sistema atinja seu propósito. Ele trata da decisão sobre quem terá de saber como usar o sistema e se ocupa em chegar a uma conceptualização clara de uma solução de *design* e de como ela será comunicada às pessoas, a fim de desenvolvam um modelo mental claro.

A característica-chave do *design* conceitual é manter as coisas abstratas – concentrar-se no “o quê” em vez de no “como” – e evitar fazer suposições a respeito de como as funções e as informações serão distribuídas. O *design* conceitual de um *site*, por exemplo, incluirá um mapa do *site* e uma estrutura de navegação. Muitos modelos conceituais diferentes são usados no método de investigação contextual.

Quer saber mais sobre *design* conceitual? Que tal fazer a leitura do texto a seguir?

<http://engenhariadesoftware.blogspot.com.br/2007/07/design-conceitual.html>

Também será interessante que você assista ao vídeo a seguir para conhecer a profissão do *Designer* de interação. Vamos lá?

<https://www.youtube.com/watch?v=Tf58sdsnPS8&feature=youtu.be>

O ***design físico***, por sua vez, preocupa-se com o modo como as coisas funcionarão e com o detalhamento da aparência e da percepção do produto. Ele trata de estruturar as interações em sequências lógicas e de esclarecer e apresentar a alocação de funções e conhecimento entre pessoas e dispositivos.

Enquanto o *design* conceitual refere-se ao propósito geral do sistema interativo como um todo, o *design* físico preocupa-se em tomar e traduzir essa representação abstrata em designs concretos. Por um lado, isso significa requisitos de *hardware* e *software* e, por outro, define o conhecimento, as tarefas e atividades que as pessoas terão de realizar.

Existem 3 componentes no *design* físico:

- ***Design operacional***

O *design* operacional preocupa-se em especificar como tudo funciona e como o conteúdo é estruturado e armazenado. Adotar uma visão funcional de uma atividade significa focar os processos e o movimento (ou fluxo) em

um sistema. Por exemplo, uma atividade pode ser desencadeada em um determinado dia ou em um determinado horário; outra pode ser desencadeada pela chegada de uma pessoa ou de um documento.

- ***Design* representacional**

O *design* representacional preocupa-se com a escolha de cores, formas, tamanhos e leiaute da informação. Ele se preocupa com estilo e estética e é particularmente importante em questões como atitudes e sentimentos das pessoas, mas também para a recuperação eficiente de informação. O estilo está relacionado à apresentação geral do sistema. Ele parece antiquado e pesado ou é ágil, harmonioso e moderno? Que humores e sentimentos o *design* provoca? Muitos sistemas procuram fazer a interação ser envolvente, mas há os que querem que ela seja instigante e outros que buscam o entretenimento. Em aplicações de multimídia e jogos isso é particularmente importante.

- ***Design* de interação**

O *design* de interação, nesse contexto, preocupa-se com a alocação de funções para os agentes humanos ou para a tecnologia e também com a estruturação e sequência das interações. A alocação de funções tem um impacto significativo em quão fácil e agradável de usar é um sistema. *Designers* criam tarefas para pessoas pela maneira como alocam funções. Por exemplo, considere a atividade de fazer um telefonema. Conceitualmente falando, certas funções são necessárias: indicar o desejo de fazer um telefonema, ligar-se à rede, informar o número de telefone, fazer a conexão.

Antecipação

Um *design* tem de ser visualizado, tanto para ajudar o *designer* a clarear suas próprias ideias quanto para que outras pessoas possam avaliá-las. A antecipação preocupa-se em encontrar a mídia apropriada

para demonstrar as ideias de *design*. O meio deve ser apropriado para o estágio do processo, para a plateia, para os recursos disponíveis e para as perguntas que o *designer* está tentando responder.

Há muitas técnicas de antecipação, mas elas incluem qualquer maneira pela qual ideias abstratas possam ser concretizadas. Esboços, protótipos totalmente funcionais e maquetes de papelão são apenas alguns dos métodos usados. Cenários, às vezes apresentados de forma pictórica, como *storyboards*, são parte essencial da prototipação e da antecipação, pois fornecem um meio de lidar com os problemas de uma ideia de *design* de forma que as questões-chave se destaquem.

Vamos retomar as características de cada uma dessas quatro atividades assistindo à videoaula com o professor André, disponível no material *on-line*!

Avaliação

A avaliação está intimamente ligada à antecipação porque a natureza da representação usada afetará o que pode ser avaliado. O critério de avaliação também dependerá de quem poderá usar a representação. Todas as outras atividades de *design* são seguidas por uma avaliação.

Às vezes, ela pode ser simplesmente uma verificação do *designer* para ter certeza de que algo está completo e correto.

As técnicas de avaliação são muitas e variadas e dependem (vale repetir) das circunstâncias. O importante é lembrar que a técnica usada deve ser adequada à natureza da representação, às perguntas que estão sendo feitas e às pessoas que participam da avaliação.

Implementação

Em última instância, todos os itens têm de ser concebidos, o *software* tem de ser escrito e testado, os bancos de dados têm de ser projetados e preenchidos e os programas têm de ser validados. O sistema todo tem de ser verificado, para garantir que ele atenda aos requisitos, até que finalmente possa ser aprovado e lançado como concluído. Os clientes, muitas vezes, pedem características extras quando veem que um sistema está próximo de ser concluído, mas essas características adicionais têm de ser orçadas e pagas.

Por outro lado, os programadores têm de garantir que seu sistema realmente atende às especificações e não contém “*bugs*” ou erros. No *Design* de Sistemas Interativos há uma variedade de métodos **formais**, **semiformais** e **informais** de especificação.

Desenvolvimento Ágil

Nos últimos anos, as grandes abordagens de engenharia com relação ao desenvolvimento de sistemas interativos vêm sendo deixadas de lado em favor de métodos “ágeis” de desenvolvimento. Esses métodos foram criados para produzir sistemas de alta qualidade adequados ao propósito a que se destinam, mas sem o enorme custo fixo associado ao planejamento e à documentação de um grande projeto de TI.

Quer conhecer um exemplo de método ágil? O vídeo a seguir nos apresentará o SCRUM!

<https://www.youtube.com/watch?v=uE3gBr4tOY0>

Há uma série de métodos concorrentes, mas provavelmente o mais conhecido vem do DSDM, um consórcio sem fins lucrativos de empresas de *software*. Seu sistema, chamado Atern, é totalmente documentado e mostra como o *software* pode ser desenvolvido em pequenas equipes. Ainda há muito debate sobre a compatibilidade desses métodos, como o *Extreme*

Programming com a abordagem centrada no humano, mas muitos dos métodos promovem a participação de programadores e *stakeholders*.

Agora, façamos a leitura do artigo a seguir para que possamos compreender a relação entre avaliação de usabilidade e os métodos ágeis sobre os quais estamos discutindo.

http://www.pucrs.br/edipucrs/Vmostra/V_MOSTRA_PDF/Ciencia_da_Computacao/83359-TIAGO_SILVA_DA_SILVA.pdf

O que mais o professor André teria a agregar à nossa discussão? Vejamos, acessando o material *on-line*!

Personas e Cenários

Para orientar o processo de *design*, os *designers* têm de considerar os elementos PACT (**P**essoas, **A**tividades, **C**ontextos e **T**ecnologias) que será discutida mais detalhadamente nas próximas aulas. As pessoas que usarão o sistema são representadas por personas (Perfis de diferentes tipos, ou arquétipos de pessoas para as quais o *design* está sendo feito).

As atividades e os contextos nos quais elas acontecerão são antecipadas por meio de **cenários de uso**. Diferentes cenários concretos podem ser usados para antecipar como diferentes tecnologias poderiam funcionar para atingir o propósito central do sistema.

Personas e cenários são desenvolvidos por meio do processo de entendimento e realizando uma análise PACT. Quase que inevitavelmente, personas e cenários evoluem juntos, já que pensar nas pessoas implica pensar no que elas fazem e pensar nas atividades implica pensar em quem irá realizá-las!

Personas

Personas são representações concretas dos diferentes tipos de pessoas para as quais o sistema ou serviço está sendo projetado. Personas devem ter um nome, alguns antecedentes e – o que é muito importante – algumas metas e aspirações.

Alan Cooper introduziu a ideia de “personas” no final da década de 1990, ideia que obteve rápida aceitação como meio de captar conhecimento sobre as pessoas às quais o sistema ou o serviço se destina. Na última edição de seu livro, ele estabelece uma ligação muito próxima entre personas e suas ideias de *design* voltado a um objetivo.

Personas querem poder fazer coisas usando seu sistema! Elas querem atingir suas metas e querem realizar atividades significativas usando o sistema que o *designer* produzirá!

Usando Personas

Os *designers* precisam reconhecer que não estão criando para si mesmos. Eles criam personas para que possam colocar a si próprios no lugar dos outros.

Como qualquer novo sistema que provavelmente será usado por diferentes tipos de pessoas, é importante desenvolver várias personas diferentes. Por exemplo, ao projetar um site para pessoas interessadas no escritor Robert Louis Stevenson, desenvolvemos personas para um professor na Alemanha, um professor universitário da Grã-Bretanha, uma criança na África e um entusiasta de Robert Louis Stevenson nos Estados Unidos.

Em um grupo tão diversificado, as pessoas têm diferentes metas e aspirações e diferem de todas as formas: física, psicologicamente e em termos do uso que fariam do *site*.

Cenários

Cenário é o conhecimento adquirido sobre o usuário e suas necessidades. Eles aparecem de várias formas no decorrer do *Design* de Sistemas Interativos e são um componente-chave de muitas abordagens do *design*. Mais recentemente, o *design* baseado em cenários emergiu como uma abordagem importante para o *Design* de Sistemas Interativos no século XXI.

Um dos principais defensores do *design* baseado em cenário é John Carroll, e em seu livro “*Making Use*”, ele ilustra como os cenários são usados para lidar com a dificuldade inerente de fazer *design*.

Ambiente

Suponhamos que, em um certo cenário, estamos analisando a influência do ambiente sobre a interação.

Dispositivos pequenos têm uma capacidade para toque mais limitada do que dispositivos de exibição maiores. Usar um dispositivo que simplesmente está longe demais da pessoa para ser tocado reflete bem os atuais ambientes de sala de estar. Em uma situação dessas, o gesto físico torna-se uma opção adequada, tanto usando as mãos como brandindo um objeto, como é o caso de um console de jogos Nintendo Wii U, por exemplo. Isso permite parâmetros como velocidade, direção e forma do movimento.

Usando cenários ao longo do *design*

Cenários (e as personas a eles associadas) são uma técnica fundamental para o *Design* de Sistemas Interativos. São úteis para entendimento, antecipação, avaliação em ambos os *designs* (conceitual e físico): os quatro estágios-chave do *Design* de Sistemas Interativos.

São quatro tipos de cenários: **histórias**, **cenários conceituais**, **cenários concretos** e **casos de uso**.

As **histórias** são experiências reais de pessoas.

Os **cenários conceituais** são descrições mais abstratas das quais alguns detalhes foram retirados.

Os **cenários concretos** são gerados a partir dos abstratos, acrescentando decisões de *design* e tecnologias específicas e, uma vez completos, eles podem ser usados como **casos de uso**. São descrições formais passadas aos programadores.

Cenários em diferentes estágios

Em diferentes estágios do processo de *design*, cenários ajudam a entender a prática corrente e quaisquer problemas ou dificuldades que as pessoas possam ter; gerar e testar ideias; documentar e comunicar essas ideias a outros e avaliar os *designs*.

Muitas histórias serão representadas por uns poucos cenários conceituais. No entanto, cada cenário conceitual pode gerar muitos cenários concretos!

Por outro lado, vários cenários concretos serão representados por um único caso de uso.

Os *designers* abstraem dos detalhes das histórias para chegar aos cenários conceituais. Eles especificam restrições de *design* nos cenários conceituais para chegar aos cenários concretos. Por fim, eles formalizam as ideias de *design* como casos de uso.

Histórias

As histórias são experiências reais, ideias, fatos curiosos e conhecimentos das pessoas. Elas podem ser captadas de qualquer forma e podem compreender fragmentos de atividades e contextos nos quais ocorrem.

As histórias das pessoas são ricas em termos de contexto. Elas também captam muitos detalhes aparentemente triviais que geralmente são deixados de lado se as pessoas forem solicitadas a fornecer relatos mais formais do que fazem.

Cenários conceituais

Os Cenários conceituais são particularmente úteis para gerar ideias de *design* e para o entendimento dos requisitos do sistema.

Cenários conceituais são mais abstratos do que as histórias. Boa parte do contexto é retirada durante o processo de abstração e histórias semelhantes são combinadas.

Uma vez que o *designer* tenha acumulado uma série de histórias, elementos comuns começarão a surgir. Nesse caso, uma série de histórias resulta no cenário conceitual, que descreve alguns requisitos de um sistema computadorizado.

Abstração

O processo de abstração envolve **classificação e agregação**: passar dos detalhes de pessoas específicas realizando atividades específicas em um contexto específico usando determinada tecnologia para uma descrição mais geral que consegue, mesmo assim, captar a essência da atividade.

Agregação é o processo de tratar um todo como entidade única em vez de olhar para seus componentes. Classificação é o processo de

reconhecer que as coisas podem ser reunidas, pois lidar com uma categoria de coisas é mais simples (mais abstrato) do que lidar com as coisas individualmente. Entre si, agregação e classificação produzem abstrações.

É claro que existem diferentes graus de abstração e uma das habilidades de um *designer* é determinar um nível adequado. O nível mais abstrato é tratar tudo simplesmente como uma “coisa” e cada atividade como “fazer alguma coisa”, mas uma representação tão abstrata geralmente não é muito útil.

Graus de Abstração

Como acabamos de ver, há vários graus de abstração.

Exemplo: Marcando um horário no médico

Pessoas com conhecimentos básicos de informática poderão entrar em contato com a clínica a qualquer momento, pela *Internet*, para ver que horários estão livres para cada médico. Elas podem marcar o horário e receber uma confirmação da consulta.

Como visto, nesse estágio há pouca ou nenhuma especificação de tecnologias precisas ou de como as funções serão proporcionadas.

O cenário pode se tornar mais abstrato, se não for especificado que a *Internet* deve ser usada, ou mais concreto (menos abstrato), especificando que a consulta deve ser marcada pelo computador e não pelo telefone celular. Encontrar o nível adequado de abstração para descrever as coisas com um determinado objetivo é uma das habilidades-chave do *designer*.

Cenários Concretos

Cada cenário conceitual pode gerar muitos cenários concretos, isso já sabemos. Quando os *designers* estão trabalhando em um problema ou questão particular, frequentemente identificam alguma característica que se

aplica somente sob determinadas circunstâncias. Nesse caso, podem desenvolver uma versão específica mais elaborada do cenário e ligá-la ao original.

Cenários concretos também começam a ditar o *design* de uma interface em particular e de determinada alocação de funções entre as pessoas e os dispositivos. Eles são particularmente úteis na prototipação e antecipação de ideias de *design* e também para a avaliação, já que são mais explícitos quanto a alguns aspectos da tecnologia. No entanto, não existe uma divisão clara entre cenários conceitual e concreto. Quanto mais o cenário for específico quanto a alguns aspectos, mais concreto ele será.

Casos de uso

Um caso de uso descreve a interação entre pessoas (ou outros “atores”) e dispositivos. É um caso de como o sistema é usado e, portanto, precisa descrever o que as pessoas fazem e o que o sistema faz. Cada caso de uso cobre muitas pequenas variações nas circunstâncias – muitos cenários concretos. Antes que os casos de uso possam ser especificados, tarefas e funções têm de ser alocadas para os seres humanos ou para o dispositivo.

A especificação dos casos de uso informa e é informada pelo processo de alocação de tarefa/função. Um conjunto de casos de uso pode ser produzido especificando a funcionalidade total do sistema e as interações que ocorrerão. Existem várias maneiras diferentes de representar os casos de uso – de diagramas muito abstratos a “pseudocódigos” detalhados.

Um método de *design* baseado no cenário

O uso de diferentes tipos de cenários ao longo do *design* pode ser formalizado como método de *design* baseado em cenários. Além dos quatro

diferentes tipos de cenário, outros quatro artefatos são produzidos durante o processo:

- Requisitos/problemas
- Corpus do cenário
- Modelo de objeto
- Linguagem de *design*

A especificação de um sistema é a combinação de todos os diferentes produtos que surgem durante o processo de desenvolvimento.

Cenários no decorrer do *design*

Algo importante para observarmos é a relação entre a especificação das restrições de *design* e o uso de cenários.

Para antecipação e a maior parte da avaliação, os cenários devem ser mais concretos. Isso significa impor restrições ao *design*. No entanto, não significa que o *designer* precisa criar um novo cenário físico e concreto cada vez que quiser antecipar um possível *design*. Pode ser que os *designers* imaginem um cenário com determinadas restrições e que isso os ajude a avaliar o *design*.

Esse tipo de geração e avaliação “e se...?” de cenários concretos é um aspecto-chave e comum do *Design*!

Requisitos e problemas

Ao reunir as histórias das pessoas e durante o processo de análise e abstração, várias questões e dificuldades surgirão. Elas ajudam o analista e

o *designer* a estabelecer uma lista de requisitos – qualidades ou funções – que qualquer novo produto ou sistema deve ter.

Por exemplo, no caso do HIC, o dispositivo tem de ser usado por pessoas idosas e míopes. Outro requisito era de que o dispositivo ficasse bonito em uma sala de estar e não se parecesse com um computador executando o sistema operacional da Microsoft®.

O formato final de requisitos e problemas pode ser uma lista priorizada de questões ou pode possuir um formato mais padronizado.

Corpus de cenários

Após realizar algumas atividades de análise, os *designers* terão reunido uma ampla gama de histórias de usuários. Algumas delas serão muito genéricas e outras bastante específicas; algumas serão tarefas bastante simples e diretas e outras serão mais vagas e assim por diante. Esses cenários conceituais frequentemente ainda estarão ancorados em um exemplo real. O difícil é encontrar um exemplo que compartilhe características com uma série de outras atividades.

É importante que, em algum momento, o *designer* reúna essas experiências aparentemente díspares para obter uma visão abstrata e de alto nível das principais atividades que o produto deverá suportar.

Características do domínio

O argumento para desenvolver um *corpus* de cenários é descobrir as **dimensões** da situação de *design* e demonstrar os diferentes aspectos dessas dimensões. Essas dimensões incluem características dos vários domínios dentro dos quais o produto operará, as várias mídias e tipos de dados que precisam ser acomodados e as características das pessoas que estarão usando o sistema.

O *corpus* dos cenários deve cobrir todas as principais funções do sistema e os eventos que desencadeiam essas funções. Diferentes tipos de interação devem estar presentes com quaisquer questões-chave referentes à usabilidade. As dimensões incluem diferentes tipos de conteúdo, diferentes maneiras de ele ser estruturado, além de questões de estilo e estética.

O objetivo é especificar os cenários a um nível de abstração que capte um grau apropriado de generalidade que seja útil ao longo de toda a gama de características demonstrada dentro de um domínio.

Modelo conceitual

O modelo conceitual é um modelo de objeto ou de dados que resulta do processo de modelagem conceitual, incluindo o desenvolvimento dos cenários e a análise de objetos e ações do *corpus* de cenários. A modelagem conceitual mostra os principais objetos do sistema, seus atributos e as relações que existem entre eles. É, então, uma parte muito importante do *Design* de Sistemas Interativos que é frequentemente negligenciada.

Um modelo conceitual claro e bem feito tornará o *design* mais fácil, de modo que as pessoas poderão desenvolver um modelo mental bom e preciso do sistema. O modelo conceitual também formará a base da arquitetura de informação de um sistema e de qualquer outra metáfora que seja usada no *design*.

Dois modelos conceituais famosos são o conceito da planilha de cálculo e os vários objetos como impressoras, pastas, documentos etc. que compõem a metáfora “Desktop” dos sistemas operacionais Windows e Mac OS.

Linguagem de *design*

A linguagem de *design* produzida consiste em um conjunto de padrões normais de interação e em atributos físicos (cores, formas, ícones etc.). Eles são todos reunidos com ações e objetos conceituais e, assim, a apresentação do *design* fica completa.

Uma “linguagem de *design*” define os elementos-chave do *design* (como o uso da cor, estilo e tipos de botões, controles deslizantes e outros *widgets* etc.), bem como alguns princípios e regras para juntá-los.

Uma linguagem de *design* consistente significa que as pessoas precisam aprender apenas um número limitado de elementos para poder lidar com uma grande variedade de situações.

Documentando cenários

Às vezes, os cenários podem se tornar confusos, e para controlá-los, é necessária uma estrutura. Podemos usar o *framework* PACT (o qual já mencionamos anteriormente), para criticar cenários e estimular *designers* a obter melhor descrição do cenário. Para cada cenário, o *designer* lista as diferentes pessoas envolvidas, as atividades que elas estão realizando, os contextos dessas atividades e as tecnologias que estão sendo usadas.

Também devemos estruturar descrições de cenários. A cada cenário deve ser dada uma introdução. A história e a autoria podem ser registradas com uma descrição de como o cenário se generaliza (em que domínios) e qual a argumentação para o cenário. Cada parágrafo de cada cenário deve ser numerado para facilitar as referências e as notas de fim incluídas onde forem levantadas questões específicas de *design*.

As notas de fim são particularmente úteis na documentação das reclamações feitas sobre os cenários (ROSSON e CARROLL, 2002). Exemplos de dados e mídia relevantes também devem ser colhidos!

Análise de reclamações

A análise de reclamações é uma parte importante do *design* baseado em cenários e é usada na identificação de problemas ou na reflexão de possíveis *designs* futuros (ROSSON e CARROLL, 2002).

O processo trata simplesmente de identificar características-chave de um cenário e de listar os aspectos bons e ruins do design. ROSSON e CARROLL usam a técnica de colocar ‘+’ ao lado das características boas e ‘-’ ao lado das características ruins. A análise de reclamações torna explícita a argumentação por trás de um *design*. Um método semelhante é listar as **questões do design**, as **opções do design** e os critérios usados para fazer as escolhas, ou seja, o **método QOC**.

Em 2013, o pessoal da IDEO produziu um vídeo chamado *Learn From Failure*, mostrando como o processo de prototipação e testes nos ajuda a entender a necessidade das pessoas, e assim aperfeiçoar os produtos. A ideia é mostrar que muitas vezes um erro não deve ser visto como um fracasso, e sim como um experimento que te ajudará a aprender!

Sobre essa questão, leia a matéria:

<http://www.uxdesign.blog.br/design-de-interacao/prototipacao-e-o-aprendizado-com-os-erros/>

Coletando dados

Quando o trabalho é feito em uma grande equipe de *design*, é útil que os cenários sejam acompanhados por dados reais. Isso significa que diferentes membros da equipe podem compartilhar exemplos concretos e usá-los como tema de discussão.

Outra característica-chave para escrever cenários é raciocinar sobre as suposições que estão sendo feitas: fazer suposições explícitas ou deliberadamente evitar tornar as coisas explícitas a fim de provocar

debate. Particularmente, o uso de **personas** pode ajudar a focar determinadas questões. Os princípios que orientam a criação de cenários são **pessoas, atividades, contextos e tecnologias**.

Referência cruzada de tipos de cenário

Outro aspecto útil da documentação é fazer a referência cruzada entre as histórias e os cenários conceituais, por meio de exemplos concretos e, finalmente, com os casos de uso.

Um sistema simples baseado na *Web* também pode ser desenvolvido! Outros pesquisadores já sugeriram ideias semelhantes que captam as múltiplas visões necessárias para avaliar como os cenários e as reclamações trabalham juntos para proporcionar um entendimento amplo de como um *design* chegou à sua forma final.

Quantas coisas novas discutimos neste tema, não é?

Vamos assistir à videoaula, disponível no material *on-line*, para reforçarmos a fixação do conteúdo!

NA PRÁTICA

Agora, lançaremos um desafio para você: vamos elaborar um esboço de sistema focando em seus possíveis usuários? Confira, a seguir, as etapas que você deverá cumprir:

- **Elaboração de Perfil de Usuário**

Trace os perfis de alunos e professores que deverão utilizar um sistema de apoio ao planejamento das aulas, divulgação de material didático e agendamento de trabalhos, provas e outras atividades. Identifique quais

perguntas de uma entrevista ou de um questionário fornecem as informações necessárias para traçar esses perfis.

- **Elaboração de Personas**

Com base nos perfis de alunos e professores traçados anteriormente, crie o elenco de personas que representam os usuários do seu sistema.

- **Elaboração de Cenários**

Elabore cenários de problema para as personas atingirem seus objetivos. Considere os objetivos mais frequentes e os mais infrequentes de cada persona. Indique quais perguntas são respondidas ou endereçadas pelo cenário.

SÍNTESE

O *Design* de Sistemas Interativos preocupa-se com pessoas, com as atividades que elas estão realizando, com os contextos dessas atividades e com as tecnologias que estão sendo usadas.

Nesta aula pudemos conhecer os principais elementos do *Design* – **entendimento, antecipação, *design* e avaliação** – e como o *design* baseado em cenário e o desenvolvimento de personas pode ser usado para orientar o *designer*.

Investigamos cenários e seus diferentes usos nesses processos e aprendemos que cenários são histórias sobre as interações entre **pessoas, atividades, contextos e tecnologias**. Cenários oferecem uma maneira eficaz de explorar e representar atividades, permitindo que o *designer* gere ideias, considere soluções e comunique-se com outros.

Cenários são usados ao longo do processo de *design* e, junto de requisitos e problemas, *design* conceitual e linguagem de *design*, podem fazer parte das especificações do sistema.

Vamos recapitular o conteúdo desta aula? Acesse o material *on-line*!

Referências

BENYON, David. **Interação Humano-Computador**. 2. ed. São Paulo: Pearson Prentice Hall, 2011

PREECE; ROGERS; SHARP. ***Design de Interação***: além da interação Homem-Computador. Porto Alegre: Bookman, 2005.