

3 Trabalhos Relacionados

Existe um grande número de processos e técnicas propostos para o projeto da interação humano-computador, todos com o objetivo de fazer com que os softwares produzidos atendam de fato às necessidades dos usuários para os quais foram construídos. Este capítulo apresenta uma visão geral de alguns processos e de uma técnica de avaliação que serviram de inspiração para a definição do eXCeed.

3.1. Modelo de ciclo de vida simples

O modelo de ciclo de vida simples proposto por Sharp e co-autoras (2007) é um modelo genérico para o processo de design de IHC (Figura 3). Composto por quatro atividades básicas, este modelo favorece a revisão iterativa e incentiva o design centrado no usuário (Gould e Lewis, 1985).

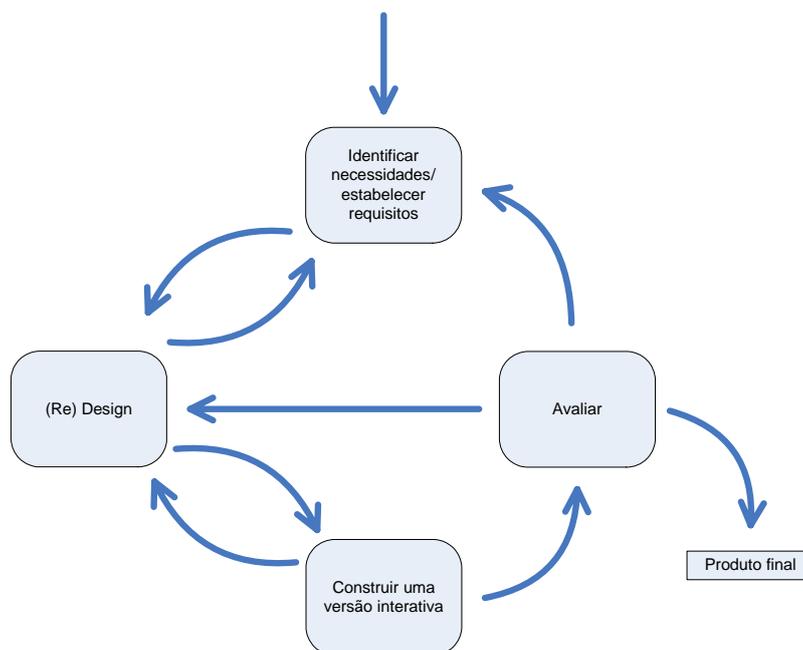


Figura 3 – Modelo de ciclo de vida simples (reproduzida de (Sharp et al., 2007)).

A primeira atividade proposta neste modelo é a identificação das necessidades do usuário, ou seja, quais atividades do usuário serão apoiadas pelo sistema. Em seguida, com base nas necessidades identificadas, são estabelecidos os requisitos que vão apoiar tais atividades.

Logo após a identificação das necessidades e requisitos, vem o design propriamente dito, onde serão propostas algumas soluções alternativas para o que foi identificado anteriormente. Em seguida, há a prototipação destas soluções para que elas possam ser avaliadas pelos usuários. Tal avaliação tem por objetivo verificar se as soluções propostas apóiam adequadamente as atividades dos usuários. Com base nos resultados obtidos desta avaliação, é possível que os designers identifiquem novas necessidades e requisitos para o sistema sendo projetado, melhorando desta maneira a qualidade da interface do sistema através da revisão iterativa do artefato, antes que a sua versão final seja produzida.

O modelo de ciclo de vida simples incorpora características fundamentais para qualquer processo de projeto da interação e o eXCeeD será definido em conformidade com este modelo.

3.2. Design dirigido por metas

O design dirigido por metas (*goal-directed design*) é formado pela combinação de diferentes técnicas e métodos amplamente utilizados para o design de IHC, focando principalmente os usuários típicos e suas necessidades (Cooper, 1996; Cooper, 1999; Cooper e Reimann, 2003). Em linhas gerais, este processo é composto de seis fases: pesquisa, modelagem, definição dos requisitos, definição do *framework*, refinamento e apoio. A primeira fase, de pesquisa, envolve uma extensa pesquisa qualitativa sobre os usuários do produto, incluindo estudos etnográficos, observações e entrevistas, e uma análise dos competidores existentes. Esta pesquisa resulta em um conjunto de padrões de comportamento e modos de execução de atividades, que guiarão as fases subseqüentes.

Em seguida, na fase de modelagem, são definidos os modelos de domínio e de usuário. Os modelos de domínio agregam principalmente informações do fluxo de trabalho, ou seja, informações referentes às atividades que os usuários desempenham no ambiente onde o sistema será implantado. Os modelos de usuários, por sua vez, são conhecidos como *personas* e representam usuários típicos do produto. Assim como qualquer pessoa, uma *persona* possui nome e outras informações pessoais, tais como personalidade, habilidades e ocupação profissional, definidas precisa e rigorosamente, de maneira que se torne o mais real possível para os designers que estão trabalhando com elas (Cooper, 1999).

Cada *persona* possui um conjunto de metas e motivações a ela associado, que devem conduzir o projeto do software. Assim, as *personas* constituem-se na base do design dirigido por metas, pois através de sua definição é possível estabelecer claramente quais as metas dos usuários, conforme a interpretação dos designers, e o produto que os designers devem projetar. Utilizadas durante todo o projeto do produto, as *personas* passam a fazer parte do cotidiano dos designers, auxiliando-os na definição e priorização das necessidades dos usuários que deverão ser atendidas com o produto (Cooper e Reimann, 2003). Por este motivo, *personas* e metas estão fortemente relacionadas, sendo usadas conjuntamente durante todo o processo (Cooper, 1999).

Cada projeto contém um conjunto determinado de *personas*, porém um software não deve atender às necessidades de todas elas. Cooper defende que atender a diferentes pontos de vista pode trazer problemas para o produto final. Por esta razão, o projeto de um software deve focar poucas *personas*, as chamadas *personas* principais.

Como as metas das *personas* principais guiarão todo o projeto do software, uma das tarefas mais importantes do design dirigido por metas é identificar quais são estas *personas* principais. Mesmo que estas *personas* representem usuários de apenas uma pequena fatia do mercado, torná-los extremamente satisfeitos é a maneira mais efetiva de tornar o produto um sucesso. Assim, se existirem produtos que atendam adequadamente cada pequena parcela de usuários, todos eles deverão ficar satisfeitos de alguma maneira.

Na fase da definição de requisitos, a partir das *personas* e metas identificadas, são empregados cenários, de maneira a validar as diferentes propostas de projeto sugeridas pelos designers (Cooper e Reimann, 2003). No design dirigido por metas, um “*cenário é uma descrição concisa de uma pessoa utilizando um software para alcançar uma meta*” (Cooper, 1999, p. 179). Para tanto, os usuários de um software devem executar tarefas, ou seja, eles devem executar passos intermediários que os ajudam a alcançar uma meta (Cooper e Reimann, 2003). Durante esta fase do processo, então, os cenários são usados para entender quais as tarefas que os usuários deverão executar com o produto. Os designers incorporam as *personas* como se fossem personagens, pensando e agindo de acordo com as características definidas para cada uma delas, numa espécie de *role-playing* (ibid.).

Uma das vantagens do design dirigido por metas reside no fato que, ao contrário das tarefas, as metas são independentes da tecnologia empregada. Por esta razão, quando os designers da interação analisam metas ao invés de tarefas, eles podem pensar em soluções diferentes e que tornem os usuários mais felizes e produtivos (Cooper, 1996). Segundo Cooper e Reimann (2003), o design dirigido por metas elimina tarefas desnecessárias dos produtos, em razão de o foco do projeto do sistema estar nas metas.

Logo após definir os requisitos, os designers definem a estrutura lógica geral do sistema na fase denominada definição do *framework* (Cooper et al., 2007). Durante esta fase, os designers transformam os requisitos definidos anteriormente em um projeto palpável do sistema, criando os *frameworks* básicos para o comportamento do produto, a sua representação visual e, no caso em que um novo hardware seja necessário, a sua forma física. Para definição desta estrutura, a equipe de designers explora diferentes idéias a partir de princípios de design da interação e de padrões de design. Estas idéias resultam na descrição textual e visual da interface do sistema, formada por esboços de interface, além da definição da seqüência de navegação entre eles.

O design dirigido por metas prossegue com a fase de refinamento, cujo intuito é refinar o projeto produzido na fase anterior, verificando-o de forma mais detalhada e preocupando-se com aspectos de implementação. Os designers se preocupam com a coerência entre as tarefas que serão desempenhadas pelos usuários e com os elementos visuais da interface, tais como fontes e ícones que serão utilizados. O resultado final da fase de refinamento é uma documentação detalhada do projeto com a especificação do comportamento e forma da interface.

Após finalizar o projeto do sistema, a equipe de designers repassa-o aos responsáveis por sua implementação. Como provavelmente, durante este período, o projeto sofrerá algumas modificações, os designers estarão disponíveis para realizar as alterações que sejam necessárias, fornecendo apoio ao desenvolvimento do sistema. Esta fase é denominada de apoio, a última fase definida no design dirigido por metas.

Concordamos com os proponentes do design dirigido a metas quando eles dizem que focar as metas do usuário é fundamental para o sucesso do produto, e que é importante elaborar uma estrutura lógica geral do projeto. Por esta razão, assim como este processo, o eXCeed manterá o foco na identificação das metas do usuário e na estrutura lógica do projeto, na forma de um diagrama de interação representado em MoLIC (Paula, 2003; Silva, 2005), complementando e contextualizando esboços de interface. Isso será realizado com um envolvimento intenso dos usuários ao longo destas atividades do processo.

3.3. Design participativo

Diferentemente do design dirigido por metas, o design participativo não possui um passo-a-passo bem definido. Ele inclui um conjunto de teorias e práticas diversas, todas com o mesmo objetivo de cooperação entre os designers e usuários no projeto de artefatos computacionais (Muller e Kuhn, 1993). Desta maneira, além de fornecer as informações necessárias, os usuários também participam intensamente durante o projeto do sistema, colaborando com informações e idéias.

Um estudo apresentado por Clement e Van den Besselaar (1993) sobre projetos que utilizaram o design participativo como parte do seu processo de desenvolvimento apontou cinco elementos importantes que ressaltam o seu forte cunho democrático. São eles: (i) o acesso dos trabalhadores a informações relevantes, (ii) posição independente destes trabalhadores nos problemas e (iii) nas tomadas de decisão, (iv) disponibilidade de métodos apropriados de design participativo e (v) flexibilidade organizacional e técnica para que haja espaço para planos alternativos. Em sua maioria, estes projetos envolviam um pequeno grupo de participantes para desenvolver um produto que seria usado apenas pela própria instituição no qual foi construído. Neles, além da produção conjunta de softwares propriamente dita, os participantes criaram novos critérios e diretrizes para avaliação da tecnologia definida e o desenvolvimento de novas técnicas de design participativo.

Carmel e co-autores (1993) apontam mais dois princípios básicos do design participativo. O primeiro princípio é o do aprendizado mútuo recíproco, no qual designers e usuários ensinam uns aos outros sobre as práticas do trabalho através de experiências conjuntas. O outro é o do “projetar fazendo”, no qual a modelagem, as avaliações e os testes são feitos conjuntamente por usuários e designers.

Para que estes princípios sejam realizados, Muller (2003), assim como Carmel e co-autores (1993), comenta sobre algumas técnicas, métodos e práticas utilizados no design participativo. Entre eles estão:

- **imersão:** da equipe de designers, que passa a trabalhar junto com os usuários em seu ambiente de trabalho de forma que possam aprender sobre o domínio e as atividades que o sistema deverá apoiar, vivenciando de fato o cotidiano do usuário;
- **jogos:** são utilizados para promover atividades que colaboram também para o aprendizado do domínio e servem como uma maneira efetiva de trazer os usuários para o processo participativo;
- **workshops:** utilizados como um momento de discussão entre os designers e usuários. É principalmente nestes workshops que eles realizam um trabalho em comum, compartilhando e negociando as decisões tomadas pelos designers (e.g., conceitos do domínio, metas e tarefas que o software deverá apoiar) de maneira que cheguem a um consenso; e
- **prototipação cooperativa:** na qual os usuários estão envolvidos também na construção dos protótipos, ao invés de somente avaliá-los.

Para realizar estas atividades, todos os envolvidos normalmente utilizam materiais familiares, tais como quadros-negros, cartões ou notas adesivas como parte da documentação durante o projeto do software (Carmel et al., 1993).

Apesar de a proposta democrática do design participativo pregar que todos os envolvidos devem ter voz ativa na construção da solução proposta (Muller, 2003), nem sempre é isso que acontece. No estudo apresentado por Clement e Van den Besselaar (1993), um dos usuários participantes dos projetos citados afirma que a participação dos usuários nem sempre significa que eles, designers e usuários, estão chegando a todas as soluções juntos. Em determinados momentos, é necessário que os designers assumam seu papel de produtores de tecnologia na defesa de uma solução que eles julguem ser a mais adequada para o problema em

questão. Essa observação está de acordo com a EngSem, teoria que embasa o eXCeed. Na visão dessa teoria, apesar de os usuários contribuírem com o projeto de software de diversas formas, cabe aos designers tomar decisões sobre a solução interativa. Segundo a EngSem, essa solução não deve apenas ser adequada ao problema, mas deve também ser comunicada apropriadamente aos usuários através da interface do software.

A proposta de cooperação entre designers e usuários do design participativo parece dar um primeiro passo na colaboração valorizada pelos métodos ágeis mais recentemente, como veremos no Capítulo 2. Da mesma maneira que o design participativo, o eXCeed também pretende envolver os usuários em grande parte do projeto de IHC, permitindo que eles forneçam informações e cooperem com os designers, como veremos mais adiante.

3.4. Communication-centered design

Um primeiro passo na direção de se definir um processo de design fundamentado na EngSem foi uma abordagem informal denominada *communication-centered design* (CCD - Barbosa et al., 2004). Os proponentes desta abordagem acreditam que, para projetar a metacomunicação designer-usuário⁷, é necessário antes apoiar a comunicação entre designers com o objetivo de criar um entendimento compartilhado sobre o que deve ser comunicado ao usuário. Para que isso possa ocorrer, em sua versão mais recente apresentada na Figura 4 (Silva et al., 2006), o CCD propõe o uso de cenários como forma de identificação das necessidades dos usuários. Já como representações de design, utiliza-se um modelo de interação conjuntamente com a prototipação através de esboços de interface. Além disso, essa abordagem propõe que as questões do sistema de ajuda online propostas por Silveira (2002) sejam utilizadas durante

⁷ Como será visto na seção 2.2, a metacomunicação designer-usuário (comunicação sobre a comunicação usuário-sistema durante a interação) é objeto de estudo da EngSem.

todo o design do software, desde a fase de elicitação de requisitos até a definição da interface do usuário (Barbosa et al., 2004).

Segundo os autores, o CCD tem duas vantagens importantes. Em primeiro lugar, o uso das expressões de comunicabilidade definidas em Silveira (2002) em tempo de design promove a criação de um conhecimento compartilhado do domínio pela equipe de designers. As respostas a estas questões auxiliam os designers na identificação de como o software deve apoiar as necessidades dos usuários no domínio identificado. Além disso, o CCD fornece recursos para projetar a interação do sistema e, também a construção e/ou escolha dos elementos de interface. Assim, o CCD surge como uma tentativa de assegurar que os conceitos do domínio a serem comunicados aos usuários através da interface do sistema sejam bem representados e entendidos por todos os envolvidos no projeto ainda durante as suas fases iniciais.

A utilização destas questões em tempo de design pretende auxiliar a manutenção da consistência entre as diferentes partes da metamsagem que estão sendo construídas. A falta de consistência na metacomunicação resulta em problemas enfrentados pelos usuários durante a interação com o sistema. Ao considerar estas questões ainda em tempo de design, os designers constroem a metacomunicação buscando evitar que estes problemas venham a surgir durante a interação do usuário com a interface do sistema. Então, através do CCD, Barbosa e co-autoras esperam contribuir com uma abordagem para a construção da metacomunicação de forma coerente e consistente.

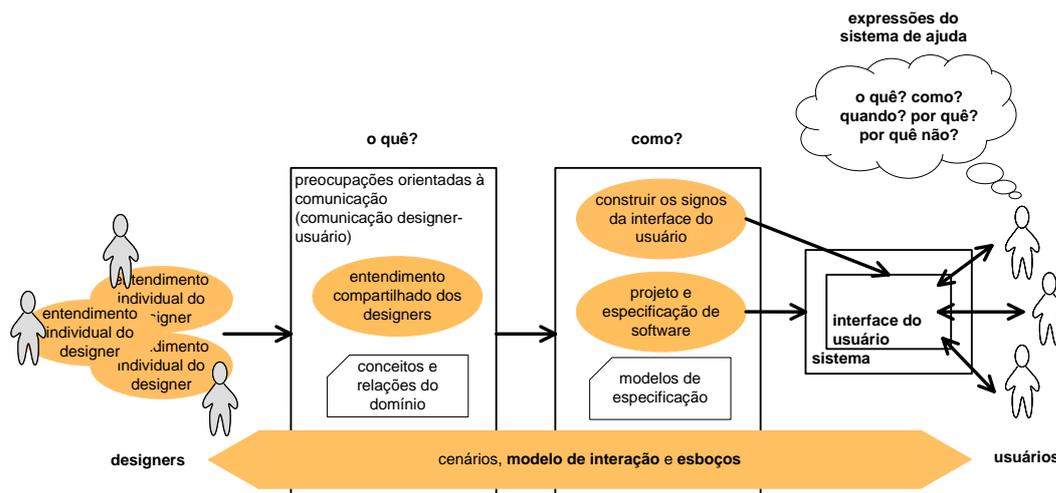


Figura 4 – Design centrado na comunicação (reproduzida de (Silva et al., 2006)).

Semelhante ao CCD, o eXCeed propõe a utilização de algumas representações que, combinadas com questões derivadas do método de ajuda online, visam contribuir para o entendimento compartilhado dos designers, bem como para a construção da metacomunicação de maneira coerente e consistente. Porém, isso ocorrerá de modo diferente. No CCD, as atividades de análise e design ocorrem sequencialmente, ou seja, elaboram-se primeiro os cenários de análise, e em seguida o modelo de interação completo durante o design. No eXCeed, as atividades de análise e design são intercaladas, e a construção da metacomunicação se dá de modo incremental. Além disso, como será visto no Capítulo 4, boa parte do que no CCD é representado explicitamente em documentação, no eXCeed será mantido na memória dos envolvidos no processo de design e em registros de áudio para eventuais consultas. Essa característica impõe uma limitação ao processo: o eXCeed se propõe a apoiar projetos de pequeno e médio porte, cuja duração seja curta o suficiente para que a equipe de design não precise recorrer com frequência aos registros de áudio para se lembrar do que foi discutido anteriormente.

3.5. Prototipação em papel

Amplamente utilizada para projeto, avaliação e refinamento de interfaces do usuário, a técnica da prototipação em papel proposta por Snyder (2003) é um “*tipo de teste de usabilidade onde usuários representativos realizam testes reais interagindo com uma versão em papel da interface do sistema, manipulada por uma pessoa simulando as ações de um computador*” (p. 4). Resumidamente, para conduzir um projeto ou uma avaliação de interface com a técnica de prototipação em papel é necessário:

- (i) determinar quais são os usuários representativos do sistema que está se querendo projetar ou avaliar;
- (ii) especificar quais as tarefas que deverão ser realizadas pelos usuários; e
- (iii) construir os esboços de interface, incluindo todo o material necessário para a realização destas tarefas, tais como as telas e os menus do sistema.

Segundo Snyder, para realizar o teste de usabilidade com a prototipação em papel, é essencial a participação de 4 personagens:

- **usuário:** pessoa que interage diretamente com o protótipo construído, simulando a seleção de opções do menu ou o clique de botões clicando sobre a sua porção correspondente no esboço;
- **computador:** pessoa da equipe de designers que simula as ações do computador, manejando os esboços de interface de maneira a traçar a saída coerente à entrada do usuário;
- **facilitador:** pessoa da equipe de designers que conduz o teste; e
- **observador:** pessoa da equipe de designers que além de observar os testes, faz anotações a respeito do que observou.

Esta técnica é utilizada principalmente para a detecção de problemas de interação da interface sendo projetada ou avaliada. Uma vez que um problema é encontrado, a facilidade de reconstruir os esboços em papel permite que uma correção seja realizada e que o teste seja refeito rapidamente. Desta maneira, é possível encontrar problemas avaliando os protótipos junto aos usuários e refiná-los de maneira a corrigir tais problemas, sem que o sistema seja (re)implementado.

Rettig (1994) afirma que a prototipação em papel é a melhor maneira de apoiar a elaboração de soluções interativas alternativas, tal como proposto pelo modelo de ciclo de vida simples apresentado na seção 3.1, uma vez que é possível revisar iterativamente os protótipos, o que contribuirá positivamente para a qualidade do software sendo projetado. Vale observar que, diferente de algumas formas de avaliação de usabilidade, a prototipação em papel não se limita a avaliar “fatores” de usabilidade, mas sim realiza uma avaliação qualitativa da experiência do usuário com o protótipo. No eXCeeD, nos inspiramos na prototipação em papel proposta por Snyder para avaliação dos protótipos construídos durante a modelagem do software para apoiar o refinamento dos protótipos e dos trechos de metacomunicação correspondentes.