

1 Introdução

O processo de desenvolvimento de software interativo envolve diferentes tipos de abordagens, que vão desde as voltadas a especificação àquelas voltadas a prototipação (Boehm et al., 1984). As abordagens voltadas a especificação iniciam o processo com as fases de análise e projeto do sistema, incluindo uma extensa elaboração de documentos e modelagem da aplicação, antes de prosseguir com as fases de implementação e testes do sistema (Pressman, 2002). A elaboração de uma documentação inicial promove uma maior reflexão sobre as possíveis soluções para o software sendo construído. Neste caso, a equipe de desenvolvimento se vê obrigada a endereçar, por exemplo, a maneira como as diferentes partes do software serão integradas e os erros que possivelmente venham a ocorrer, antes que qualquer linha de código seja escrita, o que provavelmente acarretará uma melhor manutenibilidade deste software (Boehm et al., 1984).

No entanto, este tipo de abordagem traz alguns problemas. Apesar de motivar uma maior reflexão, a necessidade de elaboração de uma documentação extensa adiciona um esforço extra ao projeto. Essa documentação pode não ser adequadamente utilizada ao longo do desenvolvimento do software, ou até mesmo ser descartada. Além disso, por se tratar de um artefato técnico, constituído principalmente por descrições textuais (e.g., descrição de casos de uso) e modelos (e.g., diagramas UML¹), esta documentação pode ser inadequada para comunicar aos usuários as soluções adotadas para o sistema sendo projetado e obter um *feedback* destes usuários, verificando se as soluções propostas de fato correspondem às suas necessidades (ibid.).

¹ *Unified Modeling Language*. Disponível em <http://www.uml.org>. Acesso em 25 nov. 2007.

Como forma de solucionar as dificuldades apresentadas, no outro extremo estão as abordagens voltadas a prototipação. Nestas abordagens, o desenvolvimento do sistema é iniciado com um projeto rápido, englobando a elaboração de pouca ou nenhuma documentação do software. Em seguida, criam-se protótipos, para avaliação do software junto aos usuários, com o objetivo de verificar se o que foi projetado de fato é adequado às suas necessidades (Pressman, 2002). Um estudo apresentado por Boehm e co-autores (1984) aponta a necessidade da inclusão de prototipação no processo de desenvolvimento de softwares altamente interativos, enfatizando que eles resultam em softwares que são mais fáceis de aprender e usar. Além disso, os autores afirmam que a utilização de protótipos facilita a comunicação das soluções propostas pela equipe de desenvolvimento de software aos usuários, uma vez que os usuários não sabem com exatidão o software que almejam até que visualizem algo funcionando. Assim, através da exploração de um protótipo, é possível que os usuários informem de maneira mais consistente se a solução proposta realmente corresponde às suas necessidades.

Não somente a exploração de um protótipo é tida como importante, mas também a construção de diferentes protótipos e a sua conseqüente exploração por parte dos usuários. Esta atividade, isto é, a construção de projetos alternativos para um mesmo sistema computacional é destacada por Sharp e co-autoras (2007) como central ao processo de design da interação. Tohidi e co-autores (2006) realizaram um estudo onde foi comprovado que, quando mais de uma alternativa de design é apresentada aos usuários, o *feedback* obtido tem uma qualidade superior ao caso em que apenas uma alternativa é apresentada. Isso se deve ao fato de que os usuários se sentem mais confortáveis em expressar as suas opiniões sobre o software que está sendo construído, pois fica claro para eles que as soluções apresentadas ainda não são definitivas. O baixo custo de produção de protótipos em papel (Rettig, 1994; Snyder, 2003) possibilita a exploração de diversas alternativas de design e facilita a revisão iterativa. Ademais, através de ciclos iterativos de testes com o usuário e revisão dos protótipos, a equipe de desenvolvimento do sistema adquire um melhor entendimento a respeito do problema e procura adequar o sistema em construção de maneira a incorporar mais fielmente as necessidades dos usuários (Pressman, 2002).

Apesar das vantagens apresentadas, em decorrência do fato de possuírem pouca documentação, os processos de desenvolvimento voltados a prototipação podem dificultar a manutenibilidade do software. Além disso, possuir um projeto pouco documentado pode significar que se refletiu menos sobre a solução antes da implementação. Esta falta de reflexão pode acarretar um número elevado de testes e reparos, além de uma maior dificuldade para a integração entre as diversas partes e para o tratamento de erros do software sendo construído (Boehm et al., 1984).

Apesar de serem inadequados à realidade de diversos processos de desenvolvimento de software e apresentarem resultados pouco satisfatórios, os métodos tradicionais vêm sendo utilizados desde os primórdios do desenvolvimento de software até os dias atuais. A crescente dificuldade encontrada com esse tipo de processo tem motivado o surgimento de outras alternativas para o desenvolvimento de software. Neste âmbito, os métodos ágeis se destacam como uma proposta inovadora, que busca se adequar à natureza imprevisível e mutável inerente ao processo de desenvolvimento de software iterativo, a fim de incorporar mais eficiência a este processo.

À medida que os softwares se tornam mais interativos e acessíveis aos mais diversos usuários, preocupações com a interação humano-computador (IHC) vêm ganhando destaque no processo de desenvolvimento de software. No entanto, restrições de prazo e orçamento acabam por limitar a aplicação de técnicas e teorias de IHC já bem estabelecidas. Em consequência, semelhante ao que ocorreu com os processos de desenvolvimento de software, uma tendência natural tem sido a adoção de práticas de usabilidade simplificadas e de base empírica, tais como listas de verificação e diretrizes (Carroll, 2003).

No sentido de lidar com tais preocupações ainda em tempo de projeto, este trabalho propõe um processo para o **projeto** da interação humano-computador que busca unir o apoio à reflexão oferecido por uma teoria de IHC com a característica de agilidade dos métodos ágeis e de técnicas de prototipação de interfaces.

1.1. Motivação e objetivo

Segundo Sharp e co-autoras (2007), o design da interação objetiva “*projetar produtos interativos para apoiar a maneira como as pessoas se comunicam e interagem em suas vidas cotidianas, seja no lar ou no trabalho*” (p. 8). A área de interação humano-computador (IHC) visa “*entender como as pessoas utilizam dispositivos e sistemas computacionais e em como torná-los mais úteis e usáveis*” (Carroll, 2003 p. 1). Os designers² de IHC, por sua vez, são aqueles profissionais preocupados em “*analisar e projetar interfaces e novas tecnologias de interface do usuário*” (ibid., p. 1).

Em sua teoria de IHC denominada Engenharia Semiótica (EngSem), de Souza caracteriza a interação como uma conversa entre os designers e os usuários de um artefato computacional (de Souza, 2005). Segundo a EngSem, o projeto da interação envolve a elaboração da metacomunicação designer-usuário sobre como este poderá ou deverá interagir com o artefato para alcançar seus objetivos, e por quê. Como forma de auxiliar os designers no processo de construção de IHC para um sistema computacional de acordo com os conceitos envolvidos na EngSem, foram propostas algumas ferramentas epistêmicas, tais como a Manas (Barbosa, 2006), a MoLIC (Paula, 2003; Silva, 2005) e o método para a construção do sistema de ajuda (Silveira, 2002)³. A Manas foi definida para a construção de sistemas colaborativos, auxiliando os designers a refletir sobre os efeitos sociais que a interação com estes sistemas pode trazer à vida dos seus usuários. A MoLIC, por sua vez, foi proposta para a modelagem e especificação da interação, antes da definição e implementação da interface do sistema. Finalmente, o método para a construção do sistema de ajuda, como o próprio nome sugere, foca a

² O termo designer será utilizado durante toda a dissertação para designar o profissional “designer de IHC”, conforme definido neste capítulo.

³ A EngSem apresenta também ferramentas epistêmicas que apóiam atividades de avaliação da IHC. São elas: o método de avaliação de comunicabilidade (MAC – de Souza, 2005) e o método de inspeção semiótica (MIS – de Souza et al., 2006). Esses métodos foram elaborados primariamente para lidar com protótipos já funcionais. Sua adaptação para lidar com protótipos em papel, como será o caso desta proposta, está fora do escopo deste trabalho.

construção do sistema de ajuda para um determinado software. Em nenhum destes trabalhos, no entanto, um processo para o projeto de IHC fundamentado na EngSem foi definido.

Uma primeira tentativa nessa direção foi a elaboração de uma abordagem que não foi definida formalmente como um processo para o projeto de IHC. Como veremos mais adiante, essa abordagem, denominada *communication-centered design* (CCD – Barbosa et al., 2004), foi utilizada como ponto de partida para a definição do processo apresentado nesta dissertação. Uma característica do CCD é a extensa documentação de análise e design gerada durante todo o processo.

Neste trabalho, propomos um processo de design para a EngSem inspirado nos métodos ágeis de desenvolvimento de software, buscando nos apoiar em lições aprendidas e aproveitar alguns aspectos positivos de diferentes abordagens. Das abordagens voltadas a especificação, propomos como documentação a utilização de descrições de metas e um modelo de interação que represente a metacomunicação sendo construída. O uso dessa documentação tem como objetivos importantes guiar o processo e promover a reflexão durante todo o projeto, no entanto sem torná-lo por demais oneroso como em documentações mais extensas. Das abordagens voltadas a prototipação, propomos a prototipação rápida através do uso de esboços de interface para facilitar a comunicação das soluções propostas pela equipe de design aos usuários e o uso da técnica de prototipação em papel para avaliar os protótipos. Dos métodos ágeis de desenvolvimento de software, utilizaremos valores e princípios do eXtreme Programming (XP) (Beck, 1999) para nortear o processo de design.

Denominamos *eXtreme Communication-Centered Design* (eXCeeD) o processo proposto nesta dissertação. No contínuo entre especificação e prototipação, o eXCeeD encontra-se mais próximo das abordagens voltadas a prototipação, sem no entanto deixar de representar alguma documentação como forma de especificação da interação, conforme ilustrado na Figura 1.



Figura 1 – Num contínuo entre a especificação e a prototipação, o eXCeed encontra-se mais próximo da prototipação.

1.2. Organização da dissertação

Este trabalho está organizado em seis capítulos. No Capítulo 2, descrevemos os principais conceitos envolvidos na fundamentação deste trabalho, os métodos ágeis e a EngSem. No Capítulo 3, é feita uma revisão da literatura com a apresentação de alguns processos de design da interação que podem ser relacionados ao eXCeed, bem como a prototipação em papel, técnica utilizada para a prototipação rápida no projeto de IHC. Em seguida, no Capítulo 4, apresentamos o eXCeed, o processo ágil proposto para o design de IHC. No Capítulo 5, apresentamos um estudo de caso realizado para avaliar a proposta do eXCeed e os resultados obtidos. Finalmente, no Capítulo 6, descrevemos as contribuições deste trabalho e sugerimos alguns desdobramentos interessantes para esta dissertação.