

Adaptação de metodologias de usabilidade para o contexto de desenvolvimento de software livre

Ana Paula Oliveira dos Santos, Prof. Dr. Fabio Kon

Departamento de Ciência da Computação – IME/USP

{ana, kon}@ime.usp.br

Resumo. *A maioria dos projetos de software livre não possuem em seu processo de desenvolvimento, práticas para obtenção de boa usabilidade. Isso se deve pela mentalidade de parte dos desenvolvedores deste tipo de software, voltada mais para a funcionalidade do que para os usuários do sistema. Iremos propor uma metodologia para se obter melhor usabilidade em software livre, com ênfase em sistemas Web, através da adaptação de métodos eficazes para o modelo de desenvolvimento de comunidades de código aberto. Tais métodos são normalmente utilizados por empresas especialistas em projeto de interação e, em menor ou maior grau, são inseridos nos processos de desenvolvimento de projetos de software fechado, o que garante em geral, uma melhor usabilidade quando comparados à maioria dos sistemas de software livre. Esse cenário, é um dos fatores que ainda limita a expansão de uso e aceitação de sistemas livres, pois perdendo-se em usabilidade perde-se também em confiança dos usuários. Inserindo-se práticas para obtenção de boa usabilidade no ciclo de desenvolvimento de software livre, não só consegue-se aumentar a aceitação desse tipo de software na sociedade como também esse processo favorece o surgimento de novas ideias para interfaces. Afinal, o modelo colaborativo e aberto do software livre possibilita uma maior exploração de um dos conceitos-chave de métodos de usabilidade: a aceitação de várias ideias de design e interação, promovida pela possibilidade de abertura à contribuição para qualquer pessoa interessada em participar. Dessa forma, o processo não é limitado a especialistas da área de Computação, pois a dinâmica das comunidades livres facilita a melhor exploração desta abertura.*

Palavras-Chave: Usabilidade, DCU (Design Centrado no Usuário), Design Participativo, Software Livre

1. Caracterização do Problema

Ao lado da própria dificuldade em se mensurar objetivamente a usabilidade de um sistema, é comum em programas de software livre que haja pouco incentivo (ou interesse) nesse aspecto, dado que a implementação das funcionalidades geralmente é a prioridade dos desenvolvedores. Essa cultura leva o desenvolvedor a iniciar o projeto pelo código, deixando o design de interfaces em segundo plano [Thomas 2008].

Em aplicações de código aberto, também existe uma tendência para o crescimento em complexidade das funcionalidades e interfaces, muitas vezes em detrimento da usabilidade. Embora aplicações não-livres também cresçam em complexidade, a competitividade do mercado faz com que existam alguns fatores moderadores desse crescimento, incluindo o custo direto do desenvolvimento de características extras e a própria preocupação com o possível impacto negativo sobre a usabilidade.

Desenvolvedores de software livre não costumam notar a baixa usabilidade de diversos sistemas, por serem geralmente usuários experientes e acostumados a interfaces de baixo nível como, por exemplo, a linha de comando. Em contrapartida, empresas que produzem software comercial muitas vezes possuem funcionários especialistas em usabilidade, o que é raro em equipes de desenvolvimento de software livre [Nichols and Twidale 2003].

Apesar desse cenário, o desenvolvimento de sistemas de código aberto vem mudando nos últimos anos com a crescente preocupação com a usabilidade, ainda que essa preocupação esteja normalmente limitada a projetos de grande visibilidade, geralmente patrocinados por grandes empresas. Porém, no âmbito das comunidades, a usabilidade raramente é considerada no processo [Nichols and Twidale 2006].

Esse tema é importante na medida em que, ao perder em usabilidade, programas de código aberto perdem usuários. Isso ocorre porque a escolha entre dois produtos com funcionalidades similares leva fortemente em conta a usabilidade, já que o software melhor nesse aspecto propicia maior credibilidade ao sistema. Uma evidência dessa importância pode ser observada na existência de diversos sistemas que, a despeito de possuírem todas as funcionalidades necessárias, não obtiveram sucesso porque os usuários não conseguiam entender como utilizá-los.

Em boa parte dos casos, pode-se dizer que para o usuário, a interface é o próprio sistema. Sendo assim, a interface torna-se parte fundamental para o sucesso de qualquer projeto, porque o produto final não será utilizado por seus desenvolvedores e projetistas – ao menos na grande maioria das vezes – e sim por usuários que não participaram do seu processo de desenvolvimento e que provavelmente não tiveram contato anterior com tal tecnologia.

Evidentemente, a preocupação com a eficiência do código também é fundamental, pois o melhor desempenho de sistemas que a consideram é inegável. Destaca-se apenas que essa não deve ser a única preocupação pois, caso contrário, sistemas de software livre não atingirão novos usuários e tampouco terão os níveis de aceitação esperados na sociedade e na indústria.

2. Fundamentação Teórica

A preocupação com usabilidade vem aumentando progressivamente no desenvolvimento de software. Isso se deve ao fato de que, a comunidade que tem acesso aos meios digitais também mudou. Segundo David Liddle [Moggridge 2007], que conduziu a equipe que projetou a interface de usuário gráfica Star, o estágio atual no desenvolvimento de tecnologias de computadores e telecomunicações é o estágio consumidor, no qual as pessoas estão menos interessadas na tecnologia em si do que em o que tal tecnologia pode fazer por elas. As pessoas não querem gastar muito tempo aprendendo como usar; portanto, se é difícil de usar, elas não utilizam a tecnologia. Desse modo, várias metodologias foram desenvolvidas com o intuito de melhorar a usabilidade de interfaces entre o ser humano e um produto computacional.

Com o foco em sistemas de computação, surgiu IHC (Interação Humano-Computador) que é um campo multidisciplinar no qual Psicologia e outras Ciências Sociais unem-se com a Ciência da Computação e campos técnicos relacionados, com o objetivo de construir sistemas que são tanto úteis quanto usáveis [Olson and Olson 2003].

Mais recentemente, uma nova grande área surgiu, Design de Interação (*Interaction Design*), que, como Rogers, Sharp e Preece definem [Sharp et al. 2007], busca projetar produtos interativos para apoiar a forma com que as pessoas se comunicam e interagem em suas vidas diárias e profissionais. Por outro lado, ela trata também de criar experiências para usuários que reforcem e melhorem a maneira que as pessoas trabalham, comunicam-se e interagem.

A fim de colocar em prática a filosofia de projetar sistemas com a devida preocupação com o usuário, algumas abordagens foram definidas como o Design Centrado em Usuário (DCU) [Vredenburg et al. 2002] que tem como base três grandes princípios: (1) Foco desde o início em usuários e tarefas; (2) Medições empíricas; (3) Design Iterativo.

A abordagem de DCU ágil [SY 2007] associa os métodos ágeis [Cockburn 2002] ao processo de DCU através da utilização da filosofia de desenvolvimento de métodos ágeis, na qual se valoriza a diminuição da distância entre design e desenvolvimento, integrando pesquisa e desenvolvimento durante todo o ciclo.

Dentre as técnicas de projeto de interface com o usuário, a que mais se encaixa no contexto de abertura do processo a todos que desejam participar é a técnica de Design Participativo [Kensing and Blomberg 1998], que possibilita a participação ativa de usuários no processo de entendimento do usuário e suas tarefas, no desenvolvimento de projetos de interface alternativos do sistema e na avaliação das alternativas. Nessa técnica, o usuário participa de todas as fases de desenvolvimento do sistema, como parte da equipe.

Projetos de software livre muitas vezes não possuem em suas equipes, pessoas com a preocupação necessária com a usabilidade. Dessa forma, esses processos são considerados menos importantes no ciclo de desenvolvimento. Porém, normalmente os sistemas possuem a usabilidade proporcional à preocupação que foi dada a ela durante o desenvolvimento.

3. Caracterização da Contribuição

A principal contribuição deste projeto será uma nova metodologia para desenvolvimento de sistemas Web de software livre, levando em conta a usabilidade em todas as etapas do processo de desenvolvimento e considerando o caráter dinâmico, colaborativo, voluntário e distribuído do software livre. Para tanto, serão necessárias adaptações aos métodos de usabilidade atuais, adequando-os às especificidades do modelo de desenvolvimento das comunidades de software livre.

Para melhor definição e entendimento da proposta, aplicou-se o próprio processo de DCU para levantamento dos usuários e do contexto de uso da metodologia. Os usuários do processo são membros de equipes de software livre, que desejam inserir práticas de usabilidade no processo de desenvolvimento. O contexto de uso são sistemas distribuídos geograficamente, com pessoas trabalhando colaborativamente. Os métodos podem ser aplicados tanto para o início de novos projetos, que tenham como usuários, pessoas não acostumadas a sistemas livres ou mesmo para a melhoria de sistemas existentes, com respeito à usabilidade. A fim de limitar o escopo do trabalho, a metodologia foi pensada, a priori, para ser aplicada no processo de desenvolvimento de aplicações Web com interfaces baseadas em XHTML e HTML para computadores pessoais (como *desktops* ou *notebooks*). Inicialmente, não estariam incluídas aplicações para sistemas móveis (como celulares) e ambientes virtuais como *Second Life*.

Um modelo que atenda efetivamente o contexto de desenvolvimento de software livre deve possibilitar que não apenas desenvolvedores participem do projeto de interface com o usuário de um sistema, mas também qualquer pessoa que tenha interesse e motivação para contribuir. Dessa maneira, possibilita-se o surgimento de novas ideias que não estarão viciadas com a mentalidade de desenvolvedores e pessoas diretamente envolvidas com a área de Computação.

Para que essa adaptação das metodologias existentes ocorra com uma integração maior com a equipe principal de desenvolvimento, que existe na maioria dos projetos livres, também se faz necessária a presença de pelo menos um membro da equipe que ficará responsável pelos assuntos de usabilidade e realizará o filtro das ideias de design e interação recebidas colaborativamente dos usuários.

Dessa forma, além de métodos de usabilidade integrarem o processo de desenvolvimento de sistemas de software livre, também torna-se possível maior inovação através da participação de vários tipos de usuários.

4. Estado Atual do Trabalho

Através de pesquisa bibliográfica e conversas com especialistas levantou-se os problemas de usabilidade decorrentes da cultura de desenvolvimento de sistemas livres e relacionou-se tais problemas com a não aplicação de métodos de desenvolvimento preocupados com o usuário final, não especialista. Decidiu-se então pela adaptação das metodologias estudadas para o contexto de desenvolvimento de software livre, para ajudar a solucionar os problemas de usabilidade desses sistemas.

No momento, os métodos de usabilidade existentes estão sendo analisados para elaboração de adaptações que concordem com o contexto de desenvolvimento de sistemas de código aberto e para que estas adaptações possam ser integradas no ciclo de

desenvolvimento. O próximo passo será estudar ferramentas que auxiliem a comunicação e a colaboração entre usuários, projetistas e desenvolvedores.

5. Trabalhos Relacionados (Visão Comparativa)

Entre os trabalhos relacionados da literatura, encontra-se o sistema UsabilityBOK (*Usability Body of Knowledge*) [UPA 2005], patrocinado pela UPA (*Usability Professional Association*), que estuda as metodologias atuais de desenvolvimento no que diz respeito a usabilidade e compartilha a necessidade de obtenção deste conhecimento com profissionais, estudantes e interessados nesse tipo de informação, pedindo a eles a colaboração com temas ainda não especificados. Esse trabalho difere da abordagem proposta por não estar diretamente ligado ao contexto de desenvolvimento de software livre e também porque os participantes contribuem com a escrita de definições e métodos para alcançar boa usabilidade, diferentemente da proposta deste mestrado, na qual a participação é realizada durante o processo de desenvolvimento de uma aplicação Web.

O sistema *Web Fast Usability* [Clark Software 2008] apresenta uma estratégia de participação de usuários na avaliação de sistemas Web, através do oferecimento do pagamento de US\$15,00 para pessoas que se interessem em testar a usabilidade de um sistema, o qual é fornecido por empresas que desejam obter um *feedback* sobre a usabilidade de seus projetos. Essa estratégia também difere da abordagem proposta pois os usuários envolvidos são motivados por razões financeiras e não filosóficas como no caso de sistemas livres e também por ser mais focada em avaliações de usabilidade e não em desenvolvimento com usabilidade.

Apesar dos trabalhos descritos tratarem problemas similares ao mencionado na Seção 1, eles não reúnem alguns pontos importantes para o problema de usabilidade de sistemas de código aberto, pois não analisam as práticas de usabilidade sob a ótica de desenvolvimento de sistemas de software livre.

6. Avaliação dos Resultados

Para avaliar a metodologia proposta pretende-se realizar estudos de caso envolvendo sistemas reais. Dessa forma, foram escolhidos dois projetos do Centro de Competência em Software Livre (CCSL) do IME-USP (<http://ccsl.ime.usp.br>), cujos aspectos de usabilidade serão analisados para verificar se houve melhoria com a aplicação dos novos métodos.

Os projetos selecionados são o SaguSaúde, desenvolvido em Ruby on Rails e o Balcão de Dúvidas, desenvolvido em Java, ambos sistemas Web. O sistema SaguSaúde possui como objetivo prover para os Centros de Saúde do SUS, ferramentas inovadoras para gerenciamento de informações dos seus usuários. O sistema Balcão de Dúvidas foi projetado para ajudar usuários a esclarecerem suas dúvidas em relação a Software Livre através do apoio de especialistas do CCSL.

Os dados obtidos com os testes e avaliações serão analisados para acompanhamento do cumprimento de métricas de usabilidade [Nielsen 2001] no decorrer do projeto. Dessa maneira, pode-se relacionar o cumprimento de tais métricas à efetiva melhora da usabilidade do sistema. Além disso, possibilita-se a compreensão de quais métodos mostraram um ganho significativo de boa usabilidade no sistema, através da análise de determinados momentos do processo, como depois da aplicação de determinada técnica.

A validação da metodologia será realizada através da utilização de critérios de comparação e medidas de desempenho de métodos de usabilidade, conforme a avaliação de métodos similares [Hartson et al. 2003].

Para encaixar a metodologia da forma mais adequada a processos usuais na comunidade de software livre, serão elaborados pesquisas e um questionário *online* com o objetivo de entender a dinâmica de desenvolvimento de sistemas abertos. Dessa forma, visa-se integrar a metodologia de provimento de maior usabilidade em sistemas de software livre, sem grandes mudanças estruturais no processo atual da comunidade de código aberto, mas com grande impacto na qualidade da interação com os usuários e na cultura de desenvolvimento.

Referências

- Clark Software, T. (2008). Fast usability. Available on-line at: <http://www.fastusability.com>.
- Cockburn, A. (2002). *Agile Software Development*. Addison-Wesley Longman Publishing Co., Inc.
- Hartson, H. R., Andre, T. S., and Williges, R. C. (2003). Criteria for evaluating usability evaluation methods. *INTERNATIONAL JOURNAL OF HUMAN-COMPUTER INTERACTION*, 15(1):145–181.
- Kensing, F. and Blomberg, J. (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work (CSCW)*, 7(3).
- Moggridge, B. (2007). *Designing Interactions*. The MIT Press.
- Nichols, D. M. and Twidale, M. B. (2003). The usability of open source software. *First Monday*, 8(1).
- Nichols, D. M. and Twidale, M. B. (2006). Usability processes in open source projects. *Software Process Improvement and Practice*, 11:149–162.
- Nielsen, J. (2001). Usability metrics. Available on-line at: <http://www.useit.com/alertbox/20010121.html>.
- Olson, G. M. and Olson, J. S. (2003). Human-computer interaction: Psychological aspects of the human use of computing. *Annual Review of Psychology*, 54:491–516.
- Sharp, H., Yvonne, R., and Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction*. Paperback.
- SY, D. (2007). Adapting usability investigations for agile user-centered design. *Journal of Usability Studies*, 2(3):112–132.
- Thomas, M. P. (2008). Why free software has poor usability, and how to improve it. *Computing & Internet, Usability*.
- UPA (2005). Usability body of knowledge. Available on-line at: <http://www.usabilitybok.org/>.
- Vredenburg, K., Mao, J.-Y., Smith, P. W., and Carey, T. (2002). A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 471–478.