

Técnicas de usabilidade em projetos ágeis aplicadas no desenvolvimento de software livre

Renan Costa Filgueiras, Paulo Meirelles

¹Faculdade Gama – Universidade de Brasília (UnB) – Brasília – DF – Brazil

renan2727@outlook.com, paulormm@unb.br

Abstract. *Many free software projects have no practices to improve usability in their development process, which may influence on the adoption of them. The free software production can be considered an agile methodology from the software engineering point of view. The aim of this study was to investigate what agile usability techniques are addressed by communities agile and software free. For this, we used the technique of systematic review where identified techniques aimed at writing stories, the design of the user, or even the use of beta testing for payment.*

Resumo. *Muitos dos projetos de software livre não possuem práticas para melhoria da usabilidade em seu processo de desenvolvimento, o que pode influenciar na adoção. O desenvolvimento de software livre pode ser considerado um método ágil do ponto de vista da engenharia de software. O objetivo deste trabalho foi investigar quais técnicas de usabilidades ágeis são abordadas pelas comunidades de software livre. Para isso, foi empregada a técnica de revisão sistemática e foram identificadas técnicas voltadas à escrita de histórias, ao design do usuário, ou mesmo o emprego de teste beta por pagamento.*

1. Introdução

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 prioridade do mesmo é a implementação das funcionalidades. Essa cultura leva o desenvolvedor a iniciar o projeto pelo código, deixando o design de interfaces em segundo plano []

Métodos de usabilidade são utilizados por empresas especialistas em projeto de interação e, 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 limita a expansão de uso e aceitação de sistemas livres, que com baixa usabilidade, perde-se também em confiança dos usuários. O uso desse tipo de sistema limita-se a usuários mais experientes em computação, o que resulta em perdas para usuários inexperientes. Muitos sistemas de software livre possuem código de qualidade, o que lhes confere algoritmos eficientes e de bom desempenho, pois foram produzidos, na maioria das vezes, por indivíduos motivados a resolver desafios ou problemas no sistema. Santos (2012) ressalta que a perda de usuários potenciais devido à baixa usabilidade configura, portanto, um desperdício de recursos para a sociedade.

Já Nichols e Twidale (2006) afirmam que 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 a linha de comando, e em contrapartida, empresas que produzem software comercial muitas vezes possuem funcionários especialistas em usabilidade, 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].

2. Usabilidade

O termo usabilidade de modo geral pode ser escrito como a facilidade com a qual um equipamento ou programa pode ser usado. O termo a ser utilizado é descrito na ISO 25010 que define como uma medida pela qual um produto pode ser usado por usuários específicos para alcançar metas específicas com eficácia, eficiência e satisfação em um contexto específico de uso [].

A usabilidade não é uma qualidade intrínseca de um sistema, ela é dependente de um acordo entre as características de sua interface e as características de seus usuários na busca de determinados objetivos e situação de uso [Cybis, 2010]. Uma interface que pode ser considerada satisfatória para determinado grupo de usuários pode ser inviabilizada por outros, e estes podem ser usuários experientes ou novatos. E a percepção pode ser diferente dependendo do ambiente de hardware, com desempenho alto ou baixo. Pode-se dizer então que a usabilidade é um acordo entre interface, usuário e ambiente.

3. Revisão Sistemática

O objetivo desta revisão sistemática é analisar relatos de experiência da abordagem de usabilidade nas comunidades ágeis que possam ser aplicadas na comunidade de software livre, com o propósito de identificar e analisar técnicas de usabilidade, com relação à forma que a usabilidade é abordada em projetos ágeis, do ponto de vista de organizações que implementam técnicas de usabilidade no processo de desenvolvimento envolvidas nas iniciativas e no contexto de projetos e estudos de caso reais.

Questão de pesquisa definida para alcançar o objetivo descrito:

1. Quais técnicas de usabilidade são abordadas nas comunidades ágeis e software livre?

Em relação ao escopo da pesquisa, os critérios adotados para selecionar as fontes de busca foram: bases com renome e difundidas na área de TI; que grande quantidade de material publicado e engenhos de busca intuitivos para filtrar os resultados; e possuir relação com o tema a ser pesquisado. Outro ponto levantado foi selecionar bases que pudessem evidenciar o cenário das universidades brasileiras em relação ao tema.

Foram selecionadas as bibliotecas digitais Google Academic e Springer Link que possuem máquinas de busca com bom funcionamento e abrangência, além das bases de teses e dissertações da UnB e USP.

Foram utilizados os termos em inglês e uma tradução em português. A expressão de busca utilizada compreendida como a String de busca foi: ((usabilidade OR usability) AND (“software Livre” OR “free software” OR “open source”) AND (“agile model” OR “método ágil”)). A String formulada por mais que abordasse aspectos da hipótese continha termos muito genéricos como palavras chave. Obteve assim resultados demasiados, chegando a encontrar mais de mil artigos em apenas uma única base, dificultando a filtragem e busca de artigos adequados e relevantes ao tema.

A fim de obter um resultado mais expressivo em relação a questão levantada, em um domínio mais específico criou-se uma nova String: ((“technical usability” or “usability practices” or “interaction design” and usability) and (“free software” or “open source”) and “agile software development”) com uma variação de sintaxe para se adequar a todas as bases selecionadas: ((“technical usability” OR “usability practices” OR “interaction design” AND usability) AND (“free software” OR “open source”) AND “agile software development”). Os termos da expressão relacionados a usabilidade estão voltados para implementação, aplicação e parte prática. Os outros termos restringem os resultados para as abordagens das comunidades e projetos ágeis e de software livre.

A seleção das publicações foi realizada em 5 etapas:

1. Seleção e catalogação preliminar das publicações coletadas nas fontes a partir da expressão de busca;
2. Filtro de seleção das publicações relevantes - 1, verificar data de publicação aplicando o critério de seleção “CS1 – ter sido publicado entre 2007 até 2013”;
3. Filtro de seleção das publicações relevantes - 2, por meio da ferramenta de busca filtrar resultados por relevância (critérios definidos pela base de busca) e selecionar os 20 primeiros da lista;
4. Filtro de seleção das publicações relevantes - 3, por meio de análise do resumo (abstract) e aplicando o critério de seleção “CS2 - possuir informações sobre técnicas de usabilidade na comunidade ágil ou software livre”;
5. Filtro de seleção das publicações relevantes - 4, por meio da leitura completa dos artigos e aplicando os critérios de seleção “CS3- possuir evidência de técnicas que foram praticadas, utilizadas, discutidas dentro de um projeto ágil ou de software livre”.

Para as publicações consideradas relevantes, os seguintes dados foram extraídos: dados da publicação (título, autor(es), data da publicação, fonte de publicação), resumo da publicação, listagem de como as técnicas de usabilidade foram adotadas e quais técnicas de usabilidade foram abordadas em projetos ágeis ou de software livre.

Os dados coletados nas publicações selecionadas foram analisados quantitativa e qualitativamente. A análise qualitativa se deu através de um mapeamento via Mapa Conceitual dos termos e técnicas encontradas e como elas interagiram. A análise quantitativa resultou em: uma lista das técnicas de usabilidade abordadas nas publicações; uma lista das técnicas utilizadas pelas comunidades ágeis; uma lista das técnicas que podem ser aplicadas pelas comunidades de software livre.

4. Resultado da Revisão sistemática

4.1. Execução da Pesquisa

Com o estabelecimento do protocolo da revisão sistemática, a pesquisa foi executada em Outubro de 2013. Na primeira etapa de seleção das publicações, a expressão de busca foi executada nas máquinas de buscas Google Academic e Springer Link e verificada nos arquivos das bibliotecas digitais das universidades USP e UnB. No Google Academic, 165 publicações foram obtidas e no Springer Link foram retornadas 55, sendo 19 comuns com o Google Academic. Nas universidades nacionais, a USP retornou 6 resultados e na UnB nenhuma publicação que atendesse à expressão de busca foi encontrada.

Na primeira etapa de seleção das publicações, o critério de seleção CS1, foi aplicado filtrando as publicações de entre o período de 2007 até 2013. No Google Academic, 118 resultados foram obtidos, porém 12 destes eram livros ficando 106 publicações e no Springer Link foram retornadas 52, sendo 16 comuns com o Google Academic e na USP 3 dos resultados eram apenas links sem referência a publicações, restando assim 3 publicações.

Na segunda etapa de seleção das publicações, as publicações foram filtradas de acordo com a relevância e foram selecionados os 20 primeiros resultando em um total de 43 publicações selecionadas conforme apresentado na Figura 1.

Na terceira etapa de seleção das publicações, o resumo (abstract) de cada publicação foi lido. Seguindo o critério de seleção CS2, as publicações que não apresentavam o resumo (abstract) via engenho de busca foram descartadas automaticamente, assim selecionando dez publicações conforme apresentado na Figura 2.

Durante a obtenção das publicações 3 foram eliminados devido a falta de acesso e um por não conter seu desenvolvimento apenas breve descrição e resumo. Na quarta etapa de seleção das publicações com os 7 artigos remanescentes, foi realizada a leitura completa dos artigos aplicando o critério de seleção CS3 para verificar a adequação junto a questão levantada.

4.2. Análise dos resultados da pesquisa

As publicações selecionadas levaram a seleção de 8 técnicas de usabilidade utilizadas na comunidade ágil que podem ser aplicadas na comunidade de software livre. Essas técnicas foram propostas visando os princípios e práticas ágeis e algumas validadas em projetos ágeis.

Santos e Kon (2012) propõe a aplicação do próprio processo de DCU (Design Centrado no Usuário) 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. []

Já Lee, Judge e McCrickard (2011) defendem a aplicação de definição de histórias de usuário de usabilidade dirigidas pela decisão do responsável por design realizando a

prototipação dessas histórias e depois sendo validadas através de testes de usabilidade conforme Figura 3 [].

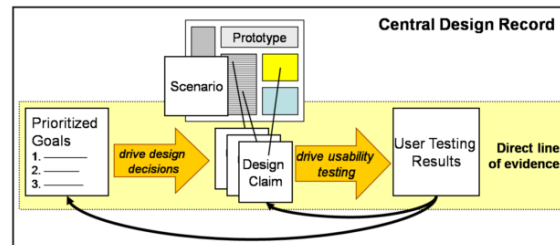


Figure 3. Ciclo de atividades de usabilidade

As atividades de usabilidade seguem o modelo de ciclo de vida das outras atividades dentro do desenvolvimento ágil com a diferença que o ciclo de usabilidade tem sua iteração primeiro que as outras atividades de desenvolvimento conforme Figura 4:

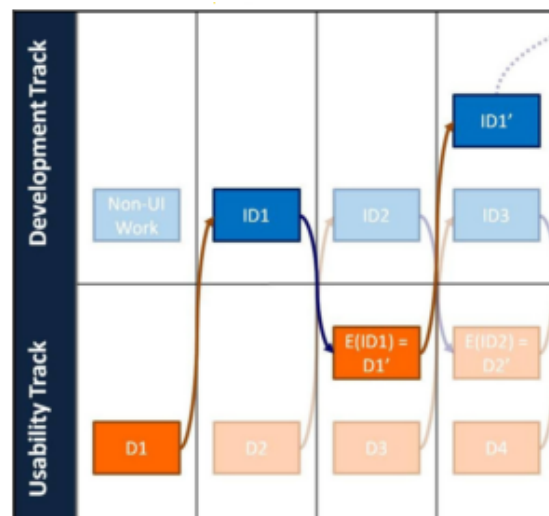


Figure 4. Ciclo das tarefas de usabilidade x desenvolvimento

Santos (2012) em uma nova pesquisa aplicada propôs a aplicação das técnicas de usabilidade de acordo com as fases do DCU, conforme a descrição das técnicas de usabilidade das comunidades de métodos ágeis e de software livre. Para a fase do DCU Criar soluções de design, não foram identificadas propostas de adaptações, sendo portanto, descritas propostas para as fases: Identificar necessidades para design centrado em humano, Especificar contexto de uso, Especificar requisitos e Avaliar Designs conforme Figura 5.

Práticas Propostas	
Fases do DCU	Práticas de Usabilidade
Identificar necessidades para design centrado em humano	Equipe-Núcleo como donos do produto Caminhos completos Especialista-generalista
Especificar contexto de uso	Pouco design antecipado e distribuído
Especificar requisitos	Definir metas de usabilidade automáticas
Avaliar designs	RITE para desenvolvedores de software livre

Figure 5. Técnicas de usabilidade baseado no DCU

Moreno e Yagüe (2012) trás em seu trabalho a inserção da usabilidade por meio da criação de User Stories de usabilidade que denominam de *Usability Stories*. Tendo que especificar os recursos funcionais de usabilidade que serão utilizados no projeto, o artigo trás um tabela que é necessária para definir as características dessa funcionalidade.

Status do Sistema	Para informar os usuários sobre a situação interna do sistema
Aviso	Para informar os usuários de qualquer ação com conseqüências importantes
Longa ação Feedback	Para informar aos usuários que o sistema está processando uma ação que vai demorar algum tempo para completar
Desfazer global	Para desfazer as ações do sistema em vários níveis
Abortar Operação	Para cancelar a execução de uma ação ou toda a aplicação
Abortar comando	Para cancelar a execução de uma tarefa em andamento
Voltar	Para voltar a um estado particular em uma seqüência de execução de comando
Entrada de texto estruturada	Para ajudar a prevenir o usuário de cometer erros de entrada de dados
Execução Passo-a-Passo	Para ajudar os usuários a realizar tarefas que requerem diferentes passos com a entrada do usuário e tal entrada correta
Preferências	Para gravar as opções de cada usuário para usar as funções do sistema
Favoritos	Para gravar determinados locais de interesse para o usuário
Ajuda Multinível	Para fornecer diferentes níveis de ajuda para usuários diferentes

Table 1. Mecanismos de usabilidade

Associado a essas características é definido três maneiras pelas quais a incorporação de usabilidade influencia as histórias de usuário.

1. A adição de novas histórias para representar requisitos diretamente derivados usabilidade.
2. Adição ou modificação de tarefas em histórias de usuários existentes. Isto significa que algumas ações decorrentes de limitações de usabilidade devem ser realizadas por um usuário existente na história. Esta tarefa pode ser tão simples ou detalhados, conforme necessário.
3. Adição ou modificação de critérios de aceitação. Estes critérios de aceitação aparecem porque a funcionalidade da história de usuário precisa incluir algumas

ações específicas para modificar o ambiente operacional.

Com isso os autores realizam um mapeamento entre os mecanismos de usabilidade e as ações a serem tomadas finalizando a proposta de como a usabilidade deve ser abordada por equipes ágeis.

	Nova Tarefa	Modificar Tarefa	Novo Critério de Aceitação	Modificar Critério de Aceitação	Nova História de Usabilidade	Nova História de Usuário
Status do Sistema		X	X	X	X	
Aviso	X		X	X	X	
Longa ação		X	X			X
Abortar Operação		X	X			
Abortar comando	X	X	X			
Voltar	X	X	X			
Entrada de texto estruturada		X	X			
Execução Passo-a-Passo	X		X	X		
Preferências						X
Favoritos		X	X		X	
Ajuda Multinível		X	X		X	

Table 2. Mapeamento entre os mecanismos de usabilidade e ações

Hussain, Slany e Holzinger (2009) não evidenciam proposta de técnicas de usabilidade, eles realizam uma busca sobre o que está sendo utilizado nas comunidades ágeis e trazem como resultados:

- Técnicas de IHC (Iteração Humano Computador). As técnicas mais utilizadas são de baixa fidelidade de protótipos, seguido de projetos conceituais, observa-se estudos adicionais dos usuários, avaliações de especialistas de usabilidade, estudos de campo, testes rápidos iterativo e testes de usabilidade em laboratório, respectivamente.
- Técnicas de IHC utilizadas com menor frequência. Contém investigação contextual, os testes não-formais de usabilidade (em pessoa), design participativo, especificações de interface do usuário completas, de alta fidelidade de protótipos, e inquérito model-driven.

Como conclusão observam que o uso de protótipos de baixa fidelidade, avaliações de especialistas de usabilidade, e teste rápido iterativo é facilmente alocado dentro das iterações rápidas de métodos ágeis [].

Ferreira e Biddle (2007) defendem o foco em design de iteração ágil antes do início do desenvolvimento de software dentro da iteração. Para eles a inserção de técnicas de design no início do ciclo de desenvolvimento podem trazer diversos benefícios sem atrapalhar ou prejudicar a equipe. Para validar essa teoria utilizaram de entrevistas com designers de interação e desenvolvimento de software em várias equipes ágeis e acompanharam a aplicação desta dentro do projeto, em um contexto de 3 projetos de desenvolvimento ágil [].

Algumas técnicas não muito aceitas pela comunidade de software livre devido a motivação partir de razões financeiras, mas exemplificadas:

Oferecimento de 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 das técnicas abordadas, 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. []

5. Considerações finais

O levantamento de técnicas de usabilidade da comunidade de métodos ágeis, técnicas de usabilidade ágil, e das técnicas de usabilidade da comunidade de software livre, foi realizado com o objetivo de refletir como técnicas de usabilidade poderiam ser aplicadas em comunidades de software livre.

Projetos de grande porte que geralmente possuem o patrocínio de grandes empresas, trás é melhor observado a aplicação das técnicas de usabilidade, pois estas podem contar com uma equipe formada por engenheiros de usabilidade e/ou especialistas em usabilidade. Já em projetos menores, raramente alguma prática de usabilidade é utilizada. Contudo, a maioria dos projetos de software livre e de projetos utilizando a metodologia ágil são desenvolvidos por equipes pequenas, compostas geralmente, apenas por desenvolvedores e que não possuem especialistas em usabilidade ou áreas relacionadas como participantes do projeto. Ainda assim, a usabilidade é importante neste contexto.

Algumas técnicas de usabilidade identificadas podem ser aplicadas sem nenhuma modificação em ambientes de desenvolvimento de software utilizando metodologia ágil, onde seja possível a realização de práticas presencialmente. Porém mesmo em um desenvolvimento distribuído é possível a aplicação dessas técnicas sem grandes modificações com apoio de ferramentas de comunicação via internet.

As técnicas foram retiradas de publicações que traziam a aplicação de técnicas de usabilidade dentro do ambiente de desenvolvimento ágil ocasionando o menor impacto possível nesse ambiente, o que facilita a compreensão e utilização dessas técnicas por parte de quase qualquer equipe de desenvolvimento ágil.

Springer	Tailoring Usability into Agile Software Development Projects
	A Green Paper on Usability Maturation
	Up-Front Interaction Design in Agile Development
	POLVO - Software for Prototyping of Low-Fidelity Interfaces in Agile Development
	What Drives Software Development: Bridging the Gap Between Software and Usability Engineering
	Second XP Workshop about Dealing with Usability in an Agile Domain
	Designing the Minimal
	User Centered Interaction Design for Mobile Applications Focused on Visually Impaired and Blind People
	iUCP – Estimating Interaction Design Projects with Enhanced Use Case Points
	Institutionalizing HCI in Asia
	Supporting Universal Usability of Mobile Software: Touchscreen Usability Meta-test
	Agile Project Building for iOS Devices
	Values and Assumptions Shaping Agile Development and User Experience Design in Practice
	Designing Interaction Concepts, Managing Customer Expectation and Mastering Agile Development in Rich Application Product Development
	A Model Based Approach for GUI Development in Groupware Systems
	Terminology Entries (Alphabetical Presentation)
	The impact of agile practices on communication in software development
	Do Teams Achieve Usability Goals? Evaluating Goal Achievement with Usability Goals Setting Tool
	A Survey of User-Experience Development at Enterprise Software Companies
	Customized Usability Engineering for a Solar Control Unit: Adapting Traditional Methods to Domain and Project Constraints
Google Academic	Facts, trends and challenges in modern software development
	Agile User Stories Enriched with Usability
	Some impacts of big data on usability practice
	Current state of agile user-centered design: A survey
	An educational testbed for the computational analysis of collaboration in early stages of software development processes
	Evaluating eXtreme scenario-based design in a distributed agile team
	Embedding e- Research Applications: Designing for Usability
	The Craft of Programming Interaction
	An Interdisciplinary Approach to Emergency Responder Mobile Technology Design
	Best practice for efficient development of inclusive ICT
	User research in a scientific software development project
	Making It Everyone's Finna-Cross-Sector Collaboration and User Experience Design in a Digital Library
	Investigating the design process: participatory design in agile software development
	Agile user centered design: enter the design studio-a case study
	Joint implicit alignment work of interaction designers and software developers
	Dissolving boundaries: social technologies and participation in design
	Adaptação de metodologias de usabilidade para o contexto de desenvolvimento de software livre
	Cloud as Context: Virtual World Learning with Open Wonderland
	User-centred design and agile development: Rebuilding the swedish national union catalogue
	Orchestrating Mobile Applications: A Software Engineering View
USP	Aplicação de práticas de usabilidade ágil em software livre
	SEEALL: Sistema Interativo de apoio ao planejamento e acompanhamento de aprendizagem baseado em competências
	Padrões de testes automatizados

Figure 1. Resultado segunda etapa de seleção das publicações

Springer	Tailoring Usability into Agile Software Development Projects
	A Green Paper on Usability Maturation
	Up-Front Interaction Design in Agile Development
	Second XP Workshop about Dealing with Usability in an Agile Domain
	Values and Assumptions Shaping Agile Development and User Experience Design in Practice
Google Academic	Agile User Stories Enriched with Usability
	Current state of agile user-centered design: A survey
	Evaluating eXtreme scenario-based design in a distributed agile team
	Adaptação de metodologias de usabilidade para o contexto de desenvolvimento de software livre
USP	Aplicação de práticas de usabilidade ágil em software livre

Figure 2. Terceira etapa de seleção das publicações